ADAPTIVE OBSERVER-BASED SUPER TWISTING SLIDING MODE CONTROL OF QUADCOPTERS FOR TRAJECTORY TRACKING AND LOW ALTITUDE GRASPING

by

Sean Smith

Submitted in partial fulfillment of the requirements
for the degree of Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
August 2023

Table of Contents

List of Tables

List of Figures

Abstract

Aerial grasping is beginning to revolutionize industrial applications through robotics in Industry 4.0. However, disturbance handling, particularly in complex trajectory tracking and low altitude flight, remains a significant issue in flight performance and aerial grasping, where UAV stability is crucial for successful grasps.

This thesis addresses the problem of position tracking and low altitude stabilization by designing, simulating, and experimentally demonstrating a control strategy to compensate for real-world unmodelled system disturbances. These disturbances include varying payloads, wind gusts, and ground effects, which can affect position tracking and object grasping during stable flight. To handle these disturbances, a multi-component control solution was developed, namely a robust Super-Twisting Sliding Mode Controller (STSMC) based on an Adaptive Higher-Order Sliding Mode Observer (AHOSMO). The STSMC approach provides a robust control method that is insensitive to unmodelled dynamics, parametric uncertainties, and external disturbances while attenuating the chattering phenomena. The AHOSMO provides access to unmeasurable states and introduces robustness in disturbance rejection by estimating system disturbances using varying gains in the observer. The strategy was extended to perform in the presence of sensor noise, which practical systems suffer from. A four-bar rigid gripper was built and attached to the COEX Clover drone to experimentally demonstrate the effectiveness of the proposed methods. The Clover consists of a COEX Pix flight controller running the PX4 flight control stack. The attitude control gains were tuned using an indirect adaptive control module while the position control gains were tuned using manual iterations.

Software-in-the-Loop (SITL) simulations were performed with the Clover Gazebo simulator to ensure the STSMC-AHOSMO worked as expected in the PX4 environment. After successful simulations, a multi-phase flight experiment was conducted to demonstrate the effectiveness of the proposed controller design in a real time application. The STSMC-AHOSMO showed significant improvement upon standard linear control methods such as Proportional Integral Derivative (PID) control.

List of Abbreviations and Symbols Used

The next list describes several abbreviations and symbols that will be later used within the body of the document

Abbreviations

2D 2-Dimensional

3D 3-Dimensional

AHOSMO Adaptive Higher-Order Sliding Mode Observer

ARC Adaptive Robust Control

DOF Degree-of-Freedom

EKF Extended Kalman Filter

ENU East-North-Up

eVTOL electric Vertical Take-off and Land

FAA Federal Aviation Administration

GMVC Generalized Minimum Variance Control

GPS Global Positioning System

HITL Hardware-in-the-Loop

HOSMO Higher-Order Sliding Mode Observer

HPF High Pass Filter

IESO Integral Extended State Observer

IGE In Ground Effect

IMU Inertial Measurement Unit

LiDAR Light Detection and Ranging

LiPo Lithium Polymer

LPF Low Pass Filter

MAE Mean Absolute Error

MRAC Model Reference Adaptive Control

NDO Nonlinear Disturbance Observer

NED North-East-Down

NMPC Nonlinear Model Predictive Control

OGE Out of Ground Effect

PD Proportional Derivative

PID Proportional Integral Derivative

RLS Recursive Least Squares

RMS Root Mean Squared

ROS Robot Operating System

RTOS Real Time Operating System

SDF Simulation Description Format

SITL Software-in-the-Loop

SMC Sliding Mode Control

STO Super Twisting Observer

STSMC Super-Twisting Sliding Mode Controller

TCP Transmission Control Protocol

UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
XML	Extensible Markup Language

Symbols

$\forall$	For all
$\in$	Belongs to
∞	Infinity
$\mathbb{Q}$	Set of rational numbers
$\mathbb{Q}^n$	Set of n × 1 rational vectors
$\mathcal{X}$	Banach space
$\otimes$	Kronecker product
$\prod$	Product sequence
$\subset$	Is a proper subset of
$\subseteq$	Is a subset of
$\sum$	Summation
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^+$	Set of positive real numbers
$\mathbb{R}^{n \times n}$	Set of n × n real matrices
$\mathbb{R}^n$	Set of n × 1 real vectors
$A > 0$	Positive definite matrix A
A^{-1}	Inverse of matrix A
A^T	Transpose of matrix A

Acknowledgements

Conducting the research presented in this thesis has been an incredibly rewarding, exciting, and enjoyable. However, it presented many challenges that I would not have been able to overcome without the guidance, encouragement, and support of many.

First and foremost, my heartfelt gratitude goes to my supervisor, Dr. Ya-Jun Pan, for granting me the opportunity to embark on this captivating research journey. Throughout my studies, Dr. Pan has been an exceptional source of inspiration, and always there to lend a helping hand when needed. Under her guidance, I have been given the freedom to explore various areas of interest, allowing me to develop and refine my skills. Her mentorship has played a significant role in shaping me into the individual I am today. My appreciation also extends to Dr. Bauer and Dr. ElSankary for their assistance as my supervisory committee members.

I would like to express my sincere appreciation to my colleagues in the Advanced Control and Mechatronics Lab for their support and collaborative teamwork throughout this journey. Being a part of this exceptional team of ambitious leaders and remarkable role models has been truly inspiring. I am grateful for the camaraderie and the willingness of my colleagues to offer their assistance whenever needed. Many friendships were made that I am grateful for.

Furthermore, the entire COEX Clover community deserves my thanks for their technical support throughout my research journey with robotic hardware. Their constant guidance and assistance have been instrumental in expanding my knowledge in this field. A special appreciation goes to Oleg Kalachev, the software lead, whose dedicated involvement in late-night software debugging played a pivotal role in bringing this research and development to fruition. His expertise and willingness to provide guidance have been invaluable, and I am truly grateful for his assistance.

Ultimately, none of this would have been possible without the support from my family, to whom I owe so much. I am thankful for the encouragement through all of the highs and lows of this journey. My mother, who taught me patience and to take a step back to keep the big picture in mind. I appreciate the unwavering

support, both emotional and financial, and constant encouragement while helping in many components of my work. My father, who taught me persistence which has been a guiding light during challenging times, and to not give up when things become difficult. My sister and brother for their continued encouragement and support, along with many other family members.

Chapter 1

Introduction

1.1 Background

The research and development of the Unmanned Aerial Vehicle (UAV) has grown extensively over the past few decades. A popular configuration comes in the form of a multirotor (Fig. 1.1) which are equipped with multiple rotors to achieve vertical takeoff, hovering, and maneuverability in the air.

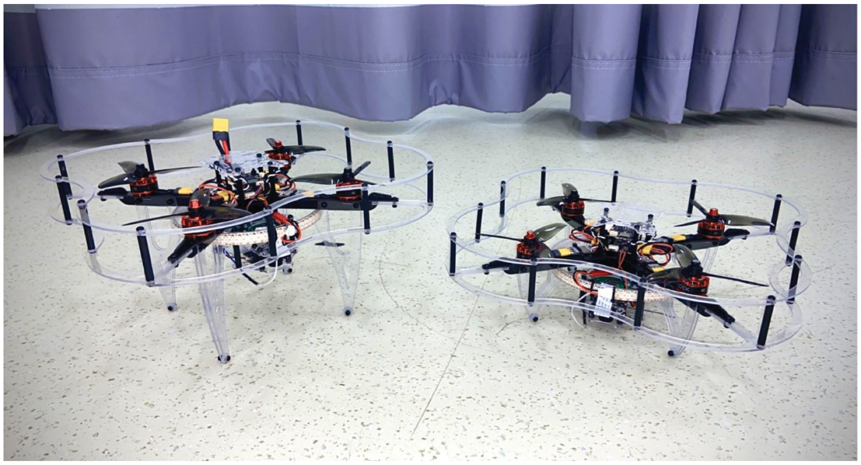


Figure 1.1: COEX Clover 4.2 drones at the Advanced Control and Mechatronics Lab, Dalhousie University.

Due to their versatile capabilities and advancements in technology, multirotors have gained immense popularity across a wide range of industry sectors. These sectors are beginning to realize the benefits through a variety of applications, including agriculture, autonomous construction, warehouse automation, transportation and delivery, firefighting, mining, security, and aerial photography, to name a few. This is by virtue of the multirotors ability to perform fixed point hovering and aggressive maneuvering with trajectory tracking. However, such tasks pose inherent challenges due to the multirotor's instability caused by nonlinear dynamics, under-actuated system behavior, modeling inaccuracies, parametric uncertainties, and the presence of bounded external disturbances. Consequently, the control system must provide sufficient tracking performance under these conditions, which constitutes the main focus of this thesis.

Significant investments are being made by companies and research groups worldwide to develop drone technology for various applications. However, policies and regulations have struggled to keep up with these advancements, which is a common challenge in the engineering field. Emerging technologies often outpace regulatory frameworks designed to ensure safety and manage risks effectively. Despite this challenge, the Federal Aviation Administration (FAA) has taken notable steps to update drone regulations. For instance, the implementation of a Remote Identity requirement for drones and the establishment of rules for commercial drone operations [12] indicate progress in adapting to evolving industry needs. As a result, we can expect a surge in the popularity of commercial drone applications in the near future, leading to an increased need for research and development.

The following sections provide examples of current companies and research groups operating within different industry sectors.

1.1.1 Consumer Deliveries

Drones have emerged as a promising solution for goods delivery, offering faster and more efficient transportation in certain scenarios. A notable company in this space is Amazon Prime Air. Amazon has been actively developing and testing delivery drones to revolutionize last-mile delivery with their first successful delivery coming in 2016 [13].

In addition to Amazon Prime Air, several other companies are contributing to the advancement of drone delivery systems. Wing, a subsidiary of Alphabet (Google's parent company), is actively involved in drone delivery initiatives. Wing's drones are capable of transporting a range of products, including food, medications, and other goods, to customers in a timely manner. Drone delivery is currently taking place in Australia with deliveries being made to customers through "DoorDash Air" [14]. Another company is Flytrex that is actively involved in the field of goods delivery. Flytrex has conducted pilot programs in collaboration with companies like EASE Drones and Walmart [2] and are currently surging in goods delivery within North

Carolina, Texas, and other locations.

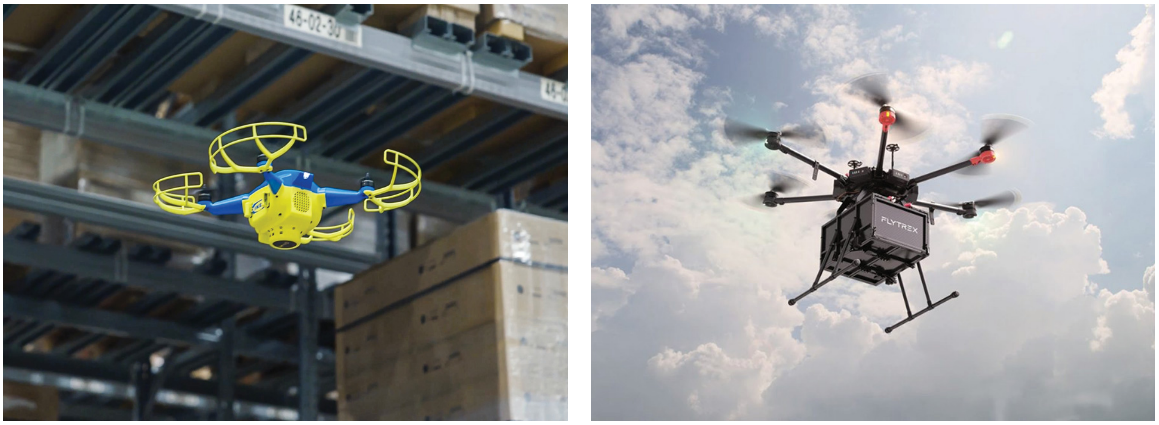
1.1.2 Agriculture

Drones are making significant strides in the field of agriculture, revolutionizing traditional farming practices and offering new solutions to enhance efficiency and crop yield. One prominent application of drones in agriculture is crop spraying. The company Guardian Agriculture employs drone technology for precise and targeted crop spraying, reducing chemical usage and promoting sustainable farming practices. Additionally, drones capture valuable data to provide growers with insights for refining crop management strategies and optimizing yield while minimizing resource waste. Guardian Agriculture recently received approval from the FAA to operate nation wide involving aerial crop spraying [15] providing an efficient substitute for standard spraying methods.

Apart from Guardian Agriculture, several other companies are actively contributing to the use of drones in agriculture. PrecisionHawk is another notable player that offers drone-based solutions for agriculture. Their drones, coupled with sophisticated software, provide detailed crop analytics, enabling farmers to identify variability across fields and optimize inputs accordingly. Additionally, companies like DJI, SenseFly, and AeroVironment have developed agricultural drones that are equipped with multispectral or hyperspectral cameras to capture specific bands of light, providing valuable data on plant health and stress levels.

1.1.3 Industry 4.0

Factories are beginning to take advantage of drone technology as automation continues to evolve with the objective to realize smart and efficient production [16]. Delivery drones have started to push forward Industry 4.0 by providing rapid and flexible logistic solutions for facility operations. Examples of these advancements include effective human-machine collaboration "cobot", where UAVs work alongside humans to produce an undisrupted flow of components, products, and workstations to their intended destinations within the facility. They extend the flow of materials by utilizing high-rise infrastructures, which provide sufficient airspace for the transportation of intralogistics.



(a)

(b)

Figure 1.2: (a) Verity drone in IKEA facility [1]; (b) Flytrex drone for aerial transportation [2].

One prominent company in the field of inventory management drones is Verity. Verity drones are specifically designed for industrial applications, including factory inventory management. They solve the problem of inventory management errors in large warehouses by automating most of the manual scanning process which can be greatly impacted by labour shortages. These drones have been integrated into 16 different IKEA store locations across Europe [1]. It is only a matter of time before we see them here in Canada.

1.1.4 Medical Needs

Drones are playing a crucial role in meeting medical needs, particularly in remote or underserved areas. Companies like Zipline, Matternet, and Swoop Aero are actively contributing to this field. Zipline has established a drone delivery network for medical supplies in several countries, delivering blood, vaccines, and medications to healthcare facilities in challenging terrains. An example being Rwanda, where Zipline became the national drone service provider [17]. Matternet focuses on transporting laboratory samples, medications, and supplies between hospitals and clinics using secure payload containers. DHL and Swoop Aero have also conducted pilot projects, utilizing drones to transport medical supplies, vaccines, and essential medications to remote areas, improving healthcare access and reducing response times.

(a) (b)

Figure 1.3: (a) Volocopters eVTOL urban taxi vehicle [3]; (b) A multirotor used for irrigation [4].

1.1.5 Transportation of People

Multirotors, also known as eVTOLs (electric Vertical Take-Off and Landing aircraft), are emerging as a potential mode of transportation for short-distance urban air mobility. They have the capability to transport people in a vertical take-off and landing manner, offering a new dimension to urban transportation.

Volocopter, known as the pioneer of urban air mobility, conducted its first manned flight in 2011 with its eVTOL [18] and since then has conducted many test flights, including public demonstrations with passengers aboard, to showcase the feasibility and potential of multirotor transportation for urban mobility. They plan on launching commercial flights in Paris by summer 2024 and being a mode of transportation for the Olympics during this time [3]. Apart from Volocopter, other companies such as EHang, Lilium, Joby Aviation, and Airbus are also actively involved in the development of multirotor aircraft for human transportation.

1.1.6 Summary

It is clear the rapid advancement of drones in various industrial sectors is driving the demand for improved UAV performance and development. As a result, there is a growing body of research from academia and an emergence of start-up companies seeking to bridge the gap between research and real-world application.

However, the aforementioned sectors still face limitations in both academic and industrial settings. With numerous sources of unknown disturbances, such as payload and aerodynamic, the flight dynamics of UAVs require robust controllers. While many academic sources rely on complex modeling or focus on isolated disturbances, practical applications call for a different approach.

In industrial UAVs, the payloads are typically restricted in size and mass, resulting in negligible effects on flight dynamics. Consequently, standard linear control methods prove inadequate, necessitating the adoption of robust-adaptive control techniques to meet critical performance constraints. Achieving trajectory tracking under these conditions requires UAV control systems that are both robust and capable of adapting to varying forces to optimize overall performance. The present research endeavors to address these challenges.

1.2 Research Problem and Contributions

This thesis aims to enhance the performance of quadrotor position tracking in the presence of various disturbances commonly encountered in real-world applications. These disturbances include aerodynamic effects such as wind disturbances and ground effect, as well as variations in payloads during pick-and-place missions. While a grasped payload affects the vehicle's dynamics by altering the mass moment of inertia, this work assumes it to be negligible and approximates it as a point mass. Consequently, the payload primarily influences the vertical axis due to the added weight on the multirotor.

The main challenge addressed in this work is dealing with the accumulation of multiple time-varying disturbances during position tracking missions. These disturbances are treated as unmodelled dynamics acting on the system to avoid the complexity of their explicit modeling. However, sophisticated adaptive-robust controller developments are required to sustain performance in such conditions. To accomplish this goal, the research objectives can be summarized as follows:

1. Investigate state-of-the-art control algorithms utilized for disturbance handling, specifically focusing on wind, ground effect, and varying payloads.

2. Derive a mathematical model for the quadrotor incorporating external disturbances for simulation and controller development.

3. Develop a highly robust and innovative control solution to address various position tracking challenges posed by the discussed disturbances.

4. Discuss the proposed software tools required for simulation verification and practical flight experiments.

5. Construct and evaluate a quadrotor rigid gripper mechanism for aerial pick-and-place missions.

6. Derive the control solution and mathematically verify its performance through rigorous Lyapunov stability analysis.

7. Validate the proposed algorithms through simulations, highlighting performance objectives.

8. Experimentally verify the control solution in a multi-phase aerial transportation mission.

These steps delineate the process for attaining the primary objectives of this thesis. The significance of this research is twofold: it encompasses both practical implementation and the advancement of control theory. While existing solutions have primarily concentrated on managing and modeling individual disturbances, this thesis tackles the more complex issue of effectively handling the accumulation of multiple disturbances commonly encountered in real-world scenarios. Furthermore, while existing literature confines the controller-observer pair presented in this thesis to improving trajectory tracking, this research seeks to expand the scope by extending the developed control solution to address manipulation tasks. Additionally, a novel adaptive mechanism is proposed to enable the control algorithm of this research in both theoretical and practical contexts. Further details are provided in Section 2, with a comprehensive understanding gained through subsequent chapters.

1.3 Thesis Outline

The structure of the thesis is presented.

- Chapter 1 provides an overview of the practical background and motivation for this research. It includes real-world examples of current and future drone technology problems, highlighting the need for improvements in position tracking performance under disturbances.

- Chapter 2 contains a literature study, examining state-of-the-art solutions to the problems found in literature. This exploration encompasses both hardware and control theory domains, facilitating a thorough comparison between the methods proposed in this study and those already established in literature.

- Chapter 3 derives the mathematical model of the quadrotor under time varying disturbances used for simulation and model based controller development.

- Chapter 4 provides a comprehensive system overview, covering a range of software tools including flight control firmware, the ground control station, and simulation environments. Additionally, the discussion extends to the hardware components, featuring details about the COEX Clover drone, sensors, and the custom-built rigid gripper mechanism.

- Chapter 5 derives, presents, and mathematically verifies the proposed controller-observer algorithm used in this work. The onboard control architecture is thoroughly explained, and the results of gain auto-tuning are presented.

- Chapter 6 tests the proposed control algorithms through extensive simulations using strategic frameworks to verify the solution in the discussed control problems. A final multi-phase experiment is conducted to demonstrate the effectiveness of this work.

- Chapter 7 draws conclusions based on the research findings and provides insights into potential future work that can further enhance the understanding and application of the proposed methods.

Chapter 2

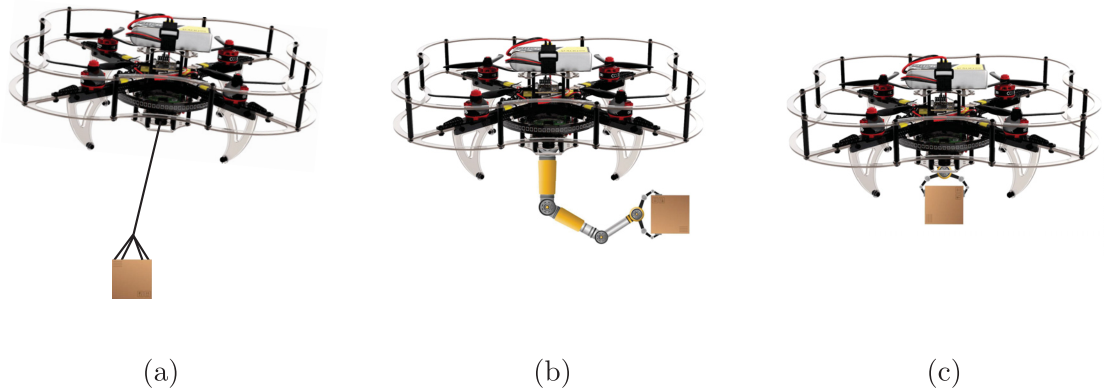
Literature Review

This chapter explores state-of-the-art methods for multirotor load transportation and control algorithms for handling disturbances in multirotor systems across various real-world applications. The primary focus is on aerodynamic wind disturbances in complex trajectory tracking, ground effect in low altitude flight, and varying payloads in transportation tasks. Additionally, the chapter identifies unexplored application limitations and presents a comparison between existing control solutions and the novel ideas proposed in this thesis.

2.1 Multirotor Load Transportation

The research on multirotor aerial load transportation can be categorized into three main segments. The first category involves the suspension of payloads, where the payload is attached to the multirotor using a rope, cable, or rigid rod as illustrated in Figure 2.1(a). These payloads are free to swing beneath the multirotor, creating an underactuated system that significantly affects the vehicle's flight characteristics. From a control perspective, two primary areas of application emerge: algorithms for generating minimum swing trajectories and anti-swing controllers.

Suspended payloads offer numerous advantages, including the capability to transport loads of various sizes and shapes, or heavy payloads with the option to employ multiple drones attached via cable connections. While this research primarily focuses on active and precise interactions with the UAV's environment, such as pick-and-drop and search-and-transport applications, the quest for a more active gripper mechanism becomes desirable. This requirement leads us to the second mode of transportation, which involves the use of manipulator arms, as depicted in Figure 2.1(b). These arms typically possess two or more Degrees of Freedom (DOF) and are generally fully actuated and compliant. However, achieving this level of control comes at the cost of



(a) (b) (c)

Figure 2.1: The COEX Clover [5]. (a) Suspended payload; (b) Robotic manipulator; (c) Rigid gripper.

increased complexity, regardless of the chosen control method. Moreover, full actuation contributes to a higher system weight due to the increased number of actuators [19]. The desire to minimize weight and reduce torsion has consequently driven the use of relatively small end effectors, which significantly curtails the manipulators' versatility and grasping capabilities [20].

The third and final method for aerial manipulation involves mounting a gripper onto the UAV to grasp objects, as depicted in Figure 2.1(c). The grasped payload is rigidly attached to the multirotor using a gripper, container or a fixed joint, effectively increasing the mass of the vehicle, changing the mass moment of inertia of the system, and possibly changing the center of gravity of the system [21]. This manipulation technique forms the central focus of this thesis, with various design considerations elaborated upon in the subsequent Section 2.1.1.

2.1.1 Payload Grasping

Aerial gripping poses a wide range of challenges including positional inaccuracies, physical and energy constraints, disturbance handling and the ability to grasp a variety of objects [22]. To compensate for positional inaccuracies and contact forces, passive mechanical compliance is introduced [23] where a large self-centering work envelop is also considered [24]. Some grasping applications use optical tracking systems to mitigate positional inaccuracies [25]. Energy conservation is considered in [26] using magnetism and high holding forces. In-house cuboid object transportation has been considered [27] although autonomous transportation remains a problem. Many of these works have compliance and adjustability in mind, however efficient setup adaptability is not explicitly addressed.

Soft robotics [28] is being integrated with grippers to minimize contact forces in high speed grasping [29]. Coupling this with 3D printed technology [30] has forwarded an improved design strategy for aerial transportation. However, a large DOF system is complex while modeling and optimization can be difficult. Linkage designs remain common in manipulations [31, 32], where this highly developed area is found in UAV perching [33] and gripping [34]. The scalability and adaptability of linkage designs provide 6-DOF solutions for precision grasping [35, 36] using parallel actuation [25]. Although in these cases, a high degree of actuation makes adaptability challenging. In [27], a miniature parallel jaw gripper was designed to transport cuboid like objects although, the complete system including the gripper, localization system, and quadcopter were unable to successfully autonomously grasp object due to positional deviations.

In light of this discussion, we adopt a 3D printed four-bar linkage design into a rigid gripper controlled by a single servo motor. The parallel concept is implemented in the form of parallel plates capable of transporting standard factory objects. The linkage setup makes it scalable and the low degree of actuation allows for repeatable and adaptable implementation. The overall system including the drone, light weight gripper, and time synchronized data transmission system is targeted to allow for fully autonomous indoor pick-and-place tasks and is presented in Chapter 4. Extensive experimental studies are carried out to verify the functions of the gripper and the seamless integration with the Clover drone.

2.1.2 Control Solutions

Successful payload grasping requires active interaction, especially in UAV pick-and-drop applications. Research indicates that traditional PID control approaches have shown limitations when applied to these tasks.

When the payload's characteristics are well-known, the standard PID controller or re-design of the PID controller often proves adequate for managing the additional payload. This concept is exemplified in [37], where a navigation system equipped with a gripper is used to transport a predetermined payload. Through the integration of a PID controller alongside a feedforward term, effective compensation for the nonlinear effects inherent in altitude dynamics, including the influence of the known payload, is achieved. Furthermore, a study in [38] investigates stability bounds of the flight controller for PID control on a helicopter. The study addresses the changes in mass moments of inertia resulting from the act of grasping objects.

An unknown load can be compensated with the integral term I of PID [39], however increasing loads run the risk of instability. Most scenarios involve grasping objects of an unknown load requiring an adaptive or robust control scheme that can autonomously stabilize the quadcopter at varying altitudes under varying loads. An Adaptive Robust Controller (ARC) was proposed in [39] to deal with parametric uncertainty in altitude control. The controller estimates the mass of the vehicle to use in the feedforward path. In the feedback path, Proportional Derivative (PD) control and a robust control technique are used. It was proven in simulation that this method outperformed classical PID control and could estimate the payload mass fairly well. The varying payload mass problem was approached in [40] with add and drop applications. The author designed and implemented a Model Reference Adaptive Control (MRAC) controller that adapts the control parameters to achieve an acceptable convergence to the desired output tracking.

A fractional order sliding mode controller in [41] based on the back-stepping method is designed to attenuate wind disturbances, load variation, and moment of inertia changes. The load magnitude is estimated through an adaptive correction coefficient applied to the mass parameter. Simulation outcomes underscore the controller's robust performance even in the presence of disturbances. Similarly, the scope of this thesis centers on addressing multiple disturbances, encompassing load fluctuations, wind gusts, and the aerodynamic forces encountered during low altitude flight, commonly known as the 'ground effect'. The ground effect phenomenon is explored in the subsequent Section 2.1.3.

2.1.3 Multirotor under Ground-Effect

Recently, multirotor vehicles have been found operating close to hard surfaces in mapping, manipulation, and facility operations. A key challenge when operating in confined spaces is handling reactive proximity effects, such as in-ground-effect (IGE). This effect arises from the aerodynamic interaction of the down-wash from the rotors with solid boundaries [42], resulting in increased thrust production that can destabilize the quadcopter and compromise performance. An illustration of this aerodynamic interaction can be seen in Fig. 2.2.

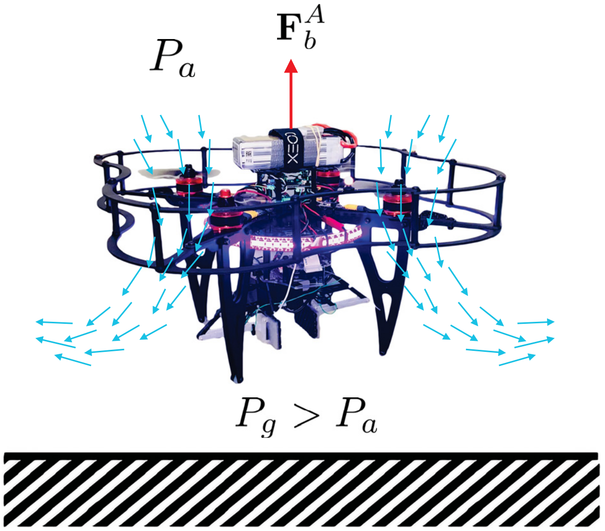


Figure 2.2: COEX Clover under ground-effect.

It has become an apparent challenge in low altitude dynamic grasping [43] with extensive modelling conducted in [44]. However, the majority of manipulation and grasping takes place at low speed or hover [45] which is the focus of this work. In many cases, ground effect is intentionally avoided during grasping tasks, as discussed in [46, 47]. This avoidance is primarily motivated by a desire to simplify the system and concentrate on other research aspects. Nonetheless, maintaining stability under these conditions remains crucial, particularly for rigid gripping tasks. Therefore, the upcoming Section 2.1.4 will investigate control methods specifically targeted to address this stability requirement.

2.1.4 Ground Effect Compensation and Disturbance Handling

Extensive research has been dedicated to developing mathematical models for systematic analysis of ground effect [44, 48, 49]. A common IGE model used in control applications is from [49], which is an adaptation of the Cheeseman and Bennett helicopter model [48]. However, altitude singularities are outlined in [50] and forward velocity limitations.

To this end, various control techniques have been used to compensate these effects. One approach involves PID gain scheduling [51], where gains are determined based on the operating conditions. However, this method only guarantees local stability and heavily relies on an IGE model. Adaptive methods such as MRAC can improve altitude performance under ground effect [51], however in this case it relies on reference ground effect models in the form of a linear function or a series of radial basis functions that were experimentally determined. Moreover, this method struggles to adapt to arbitrary disturbances in multi-disturbance environments. Adaptive backstepping control [52] adjusts the empirical unknown parameters in the IGE model to handle model uncertainty and improve IGE flight performance, although significant disturbances like load variation are not considered. Similarly, an adaptive sliding mode controller is presented in [53] that uses an adaptive law to determine a lumped parameter including the empirical IGE model coefficient and unknown system mass for the model based controller. The combination of this technique with the robustness of the SMC compensates for modeling errors, thus enhancing flight performance IGE with varying loads. However, performance may vary within low disturbance zones depending on the gain selection.

Using an empirical IGE model as a compensator in the controller design through the vertical thrust channel has improved the stability and tracking performance [49, 53, 54]. In this case the model reduces the control input proportionally to the ground effect kinetics acting on the quadrotor. Combining this with robust sliding mode control [53, 54] suppresses modeling error, improves tracking, and ensures stability through proven theoretical analysis. Although, the performance response of the multirotor IGE and out-of-ground-effect (OGE) may vary with gains that do not adjust with the fluctuating disturbances.

Data-driven approaches such as those presented in [55] have demonstrated their capability to encompass a variety of disturbance sources within the control loop, as highlighted by research in [56]. Nevertheless, these methodologies often overlook velocity considerations within the Gaussian models, which can constrain acceleration predictions and the handling of disturbances not encountered in the training data, such as external wind perturbations. Gaussian processes, as discussed in [57], are employed to address the challenges posed by variable payloads and ground effects during low altitude grasping. In this context, the body frame velocity is incorporated as a modeling feature. To further enhance performance, a Nonlinear Model Predictive Controller (NMPC) is integrated. However, it's important to note that external disturbances are typically treated as constants in these approaches, necessitating extensive data sampling and retraining for adaptation across different platforms, as indicated by findings in [58].

Table 2.1: Comparison of control strategies for disturbance handling.

Control Method: Sub method	Details	**Pros** and **Cons**
Adaptive: Backstepping [52], sliding mode [53], robust [39], MRAC [40, 59]	Adapt the controller to mass variation, or the thrust coefficient, usually defined as a model compensator with an empirical coefficient to be adaptively determined, or to external forces in the presence of the ground.	**P**: Adaptive methods can handle varying platforms better compared to empirically determined ones, does not trade performance for modeling accuracy. **C**: Has trouble adapting to arbitrary disturbances.
Gain scheduling and gain redesign: PID [37, 38, 51]	Gains are scheduled based on known operating conditions or determined from an IGE model.	**P**: Uses non-model based simple linear control. **C**: Unable to adapt to changing disturbances, performance is confined to a restricted domain predicted by a limited model which makes it undesirable for most real-time applications.
Empirical model with Robust control: sliding mode [54]	Empirical IGE model is used to compensate GE forces and a robust controller can cope with parametric/modelling uncertainty.	**P**: Robust against disturbances and parametric uncertainty. **C**: Varying performance with varying disturbance levels, chattering phenomena.
Observer or estimator-based: IESO [60], NDO [50], HOSMO [61, 62, 63]	Uses an observer/estimator to observe/estimate the unmodeled external disturbances and feed it into the control input for compensation.	**P**: Estimation of states and disturbances, great transient response, little modelling required. **C**: Sensitive to noise (observer), computationally expensive (estimator).
Learning Methods: Gaussian [55, 56] + NMPC [57]	Determine data driven models using Gaussian process regression to improve control	**P**: Reliable performance improvement in a repeated setting. **C**: Requires sophisticated data sampling, trouble adapting to new environments and disturbances.

Observer-based methods increase robustness under an accumulation of arbitrary disturbances. An integral extended state observer (IESO) [60] aids IGE compensation with a control scheme that cannot handle the disturbance by itself. A nonlinear disturbance observer (NDO) [50] is applied to the quadcopter system to observe and estimate the IGE including both the forces and torques it causes on the system. These methods primarily aim to improve the flight IGE rather than focusing on manipulation tasks. However, they incorporate observers that can estimate various forms of disturbances, including wind gusts and varying payloads.

The HOSMO is widely used robust observer that is capable of providing state information and external disturbance estimation. References such as [61, 62, 63] use it to estimate and compensate time varying external wind disturbances on the system. It has not been tested in UAV manipulating tasks, which is part of the contribution of this thesis.

A summarized overview of control methods can be found in Table 2.1. It should be noted that many of these methods can be categorized into multiple sections, where the topic focus with reference to application and controller development varies significantly across different literature sources.

2.2 Background Theory

This section introduces the fundamental background theory required to understand the controller and observer development in subsequent sections along with their theoretical proofs.

2.2.1 Lyapunov Stability

The Lyapunov stability conditions are a set of mathematical conditions that allow us to determine the stability properties of an equilibrium point or a trajectory of a dynamical system. These conditions rely on the existence of a Lyapunov function, which is a scalar function that describes the energy of the physical system.

The basis of Lyapunov theory can be described as follows. Consider the autonomous nonlinear system $\dot{x} = f(x)$. It is said to have a Lyapunov function $V(x)$ if in a ball $\mathbf{B}_{R_o}$, denoting a spherical region and domain of attraction defined by $||x|| < R_o$ in state space, it is positive definite $V(x) > 0$; has continuous partial derivatives; and its time derivative along any state trajectory of the system is negative semi-definite $\dot{V}(x) \leq 0$. Local asymptotic stability is guaranteed for the equilibrium point $x = 0$ if the following conditions can be met for $V(x)$ [64]:

- $V(x)$ is positive definite ($V(x) > 0$ locally in $\mathbf{B}_{R_o}$)
- $\dot{V}(x)$ is negative definite ($\dot{V}(x) < 0$ locally in $\mathbf{B}_{R_o}$)

The stability at the origin $x = 0$ can be extended globally ($\mathbf{B}_{R_o}$ extended to the whole state-space) if $V(x)$ is *radially unbounded*. This occurs if the following conditions are met [64]:

- $V(x) \to \infty$ as $||x|| \to \infty$ (x tends to infinity in any direction).

A few other stability definitions are used within this thesis and they are listed below, where the origin $x = 0$ for the system is (locally) globally [65]:

- *Finite-time stable*, if all trajectories starting in (a neighborhood of $x = 0$) $\mathbb{R}^2$ converge to $x = 0$ in finite time.
- *Robustly stable*, if all trajectories starting in (a neighborhood of $x = 0$) $\mathbb{R}^2$ converge to $x = 0$ under perturbations vanishing at the origin.
- *Exactly stable*, if all trajectories starting in (a neighborhood of $x = 0$) $\mathbb{R}^2$ converge to $x = 0$ in finite time, under perturbations that are non vanishing at the origin.

Nonlinear controller design using Lyapunov theory can be done in two main ways. The first method involves hypothesizing a Lyapunov function then searching or a control law that will satisfy the stability conditions. An example can be found in [66] where adaptive gains for the recursive HOSMO (the HOSMO in this thesis is non-recursive) are derived from the Lyapunov function candidate to satisfy the stability conditions. In contrast, the adaptive gains for the AHOSMO in this thesis are assumed in a certain form and a Lyapunov function is found to justify the choice.

2.2.2 Exact and Robust Observer

An exact differentiator is developed in [67] for some input $\sigma(t)$ meaning is provides the true derivative $\dot{\sigma}(t)$ absent of noise. As stated in [68] the differentiator can also be called robust for $\sigma(t)$ if the output tends uniformly to the true derivative $\dot{\sigma}(t)$ when the input tends uniformly to $\sigma(t)$ in the absence of noise.

The Super-Twisting algorithm used in this work is a discontinuous exact and robust second order algorithm. The observer is a higher third order system being able to exactly estimate the first and second order derivatives of the system, or the velocity and bounded external perturbations.

2.2.3 Lipschitz Condition and Diffeomorphism

Lipschitz Condition

Consider a function $f(t,x)$ that is continuous on $[t_0, t_0+\delta t]$, $(\delta t \geq 0)$, it is said to be locally Lipschitz if

$$\begin{gathered} ||f(t,x)|| \leq c, \\ ||f(t,x_1) - f(t,x_2)|| \leq L||x_1 - x_2||, \end{gathered} \tag{2.1}$$

for $c, L \geq 0$ such that $x, x_1, x_2 \in B_r(x_0)$ where $B_r(x_0) = \{x \in \mathcal{X} |\ ||x - x_0|| \leq r\}$ for any $t \in [t_0, t_0 + \delta t]$ where $\mathcal{X}$ is a Banach space (any concrete space of functions). Any such L is referred to a Lipschitz constant for (2.1) [69]. Any function with a bounded first derivative must be Lipschitz, where f is locally continuous, and if no value of L satisfies (2.1) then the function is non-Lipschitz and its derivative is not bounded. Conditions in (2.1) must be fulfilled for all $x_1, x_2 \in \mathcal{X}$ corresponding to $r \to \infty$ to be globally Lipschitz.

Diffeomorphism

A diffeomorphism is used within the observer gain definition to robustify it against disturbances. The formal definition is given as [70]:

- A diffeomorphism Φ: $\mathcal{L} \to \mathcal{Z}$ with $\mathcal{L}, \mathcal{Z} \subseteq \mathbb{R}^n$ is a one-to-one (bijection) continuous/smooth differentiable map or manifold, and its inverse Φ^{-1} exists and is smooth.

It is easy to check whether it is a local diffeomorphism by using the [64, Lemma 6.2].

2.2.4 Homogeneity

It is important to recall some definitions on homogeneity (see [71] for continuous systems). The definitions are as follows

- A function $V : \mathbb{R}^n \to \mathbb{R}$, $x \in \mathbb{R}^n$ is *homogeneous of degree* $q \in \mathbb{R}$ if $V(\kappa^{r_1} x_1, ..., \kappa^{r_n} x_n) = \kappa^q V(x)$, $\forall \kappa \in \mathbb{R}$, $\kappa > 0$ with weights $\mathbf{r} = [r_1, ..., r_n]^T \in \mathbb{R}^n$.

- The vector field $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$, $\mathbf{f}(x) = [f_1(x), ..., f_n(x)]^T$ (respectively, the vector-set field $\mathbf{F}(x) \subset \mathbb{R}^n$) is *homogeneous of degree* q if for every f_i is homogeneous of degree $q + r_i$.

- A differential inclusion $\dot{x} \in F(x)$, is *homogeneous of degree* q if its vector-set field F is homogeneous of of degree q.

A homogeneous Lyapunov function is used to prove the stability of the non adaptive observer in later sections. This was eventually generalized to a family of homogeneous Lyapunov functions in other literature [72, 73].

A class of homogeneous systems

According to [73] the Lyapunov function for the proposed observer must belong to a particular class of functions Generalized Forms (GF). This is defined as follows. Let $F : \mathbb{R}^n \to \mathbb{R}$ be a homogeneous function of degree m with a homogeneity weight vector $\mathbf{r} = [r_1, ..., r_n]^T \in \mathbb{Q}^n$ and let the numbers $\kappa \in \mathbb{R}$ and $\rho \in \mathbb{Q}$. F is said to be a GF if it is only expressed as sums and products of the following form:

$$F_i = \sum_{j=1}^{n} \kappa_j \lfloor x_j \rceil^{\rho_j} + \sum_{k=1}^{s} \kappa_k \prod_{j=1}^{n} \lfloor x_j \rceil^{\rho_{k,j}}. \tag{2.2}$$

It is common to see these functions appear in finite-time observers and controllers such as the Super-Twisting Algorithm and High Order Sliding Mode Observer/Controller (observer in this thesis).

2.3 Contributions Summary

The main control application contributions of this thesis are summarized as follows.

Develop an adaptive robust control strategy that achieves stable position tracking under time varying perturbations. Refine this control problem to stable altitude control in a multi-disturbance UAV application involving ground effect and varying loads from low altitude grasping tasks. The accumulated disturbances and process dynamic uncertainty is compensated for with an AHOSMO and high performance trajectory tracking is accomplished with a STSMC that is robust against these conditions. The chattering phenomena is attenuated by using the super twisting algorithm paired with disturbance cancellation from the AHOSMO. This control design is novel in the following ways. First, an adaptive law is proposed for the HOSMO, enabling the observer to estimate and compensate bi-directional disturbances. This adaptive approach minimizes the control effort in both low and high disturbance zones, commonly found in aerial pick-and-place tasks. As a consequence, optimal performance is achieved within IGE and OGE regions. The adaptive law minimizes noise sensitivity that is common in high gain observers. Detailed Lyapunov stability analysis is provided to highlight that finite time convergence properties are maintained with the adaptive law. To the best of our knowledge, the HOSMO has been limited to unidirectional adaptive laws assuming constant or increasing disturbances, which is not the case in most UAV applications. Second, we combine the AHOSMO with the STSMC removing the single gain condition from this controller-observer pair [74]. Likewise, to our best knowledge this controller-observer pair has been limited to flight improvement outside of manipulation tasks. Finally, we demonstrate the proposed control algorithm in multi-phase simulations and experiments to rigorously analyze the benefits of the key aspects of our control design relative to the widely used PID control within a targeted condition framework involving ground-effect and payload handling.

Chapter 3

System Modeling

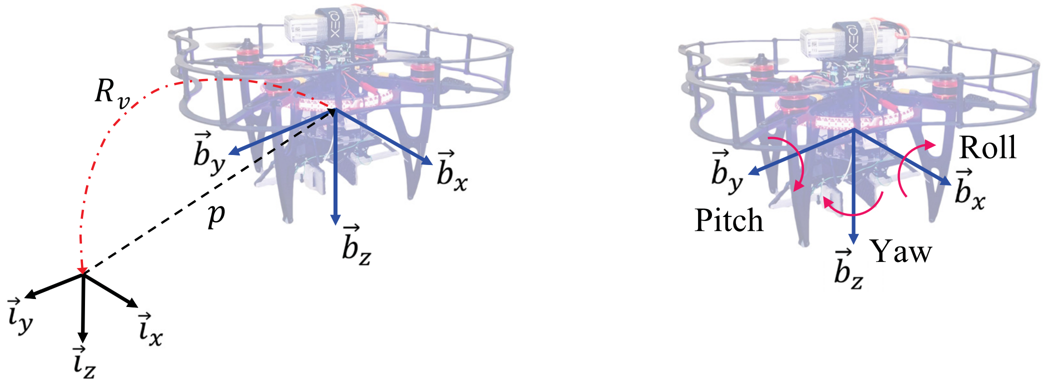
This chapter will derive the nonlinear model of a quadrotor UAV along with providing an empirically determined ground effect model, which will be used in subsequent sections for simulating the vehicle and designing controllers. The quadrotor used in this project is the COEX Clover 4.2. This quadcopter is configured in the X shape, where the front of the vehicle is positioned between two of the propellers, as illustrated in Fig. 3.1(a).

The chapter begins by defining different coordinate frames and providing an overview on quaternions, which are used to represent the vehicle's attitude. Finally, the chapter describes the equations of motion for the 6DOF and the various forces and moments acting on the vehicle including wind disturbances, varying payloads, and ground effect.

3.1 Coordinate Frames

Consider the 6DOF COEX Clover quadcopter in Fig. 3.1(a), the unit vectors on the Clover are located at the center of mass and make the body-fixed reference frame forming the rotation matrix $\mathbf{R}_v = [\vec{b}_x \quad \vec{b}_y \quad \vec{b}_z] \in SO(3)$. This matrix gives the transformation from the body-fixed reference frame to the North-East-Down (NED) inertial reference frame $\{\vec{i}_x \ \vec{i}_y \ \vec{i}_z\}$. The modelling assumes a non-rotating, flat earth, which is valid considering the quadrotor's operation over short distances. Typically, the origin of the inertial frame aligns with the quadrotor's takeoff location, where the x-axis points North, the y-axis aligns with East, and the z-axis coincides with the Down direction. The body frame's orientation relative to the inertial frame involves both a displacement from the origin and a rotation, fully describing the quadcopter's 6-DOF.

All of the high-level algorithm development discussed in Chapter 4 for autonomous flight testing is conducted in the East-North-Up (ENU) reference frame. However,



(a)

(b)

Figure 3.1: (a) Coordinate frames; (b) Body reference frame angular terminology.

existing software plugins automatically implement the reference frame transformation to the NED frame which low-level controller development operates in.

3.2 Quadrotor Modelling

A quadrotor can be modelled as a rigid body with 6DOF, which includes three translational DOF and three rotational DOF. This modeling assumes that the quadrotors motion is purely translational and rotational without any significant deformation or flexibility.

This thesis is focused on designing and implementing an adaptive robust control strategy within the position control module in the PX4 flight stack, to track reference positions and their rates therefore the position dynamics are the main modelling focus. However, both attitude representations, in the form of Euler angles [75] and quaternions are referenced in this work where the output from the proposed control methods are converted to quaternion setpoints for the angle controllers. Therefore, a brief overview of both are given.

3.2.1 Euler Angles

Describing the attitude of a rotating body with a body-fixed axis is approached through the use of Euler angles ψ, θ, and ϕ which correspond to yaw, pitch and roll respectively. This can be done by applying three consecutive rotations mapping linear space $\vec{i} \in \mathbb{R}^3$ to linear space $\vec{b} \in \mathbb{R}^3$.

Three Euler angle rotations are applied in a predefined order to rotate from the inertial to body frame. The Euler 3-2-1 sequence is used to parameterize the rotations for this work. An illustration can be found in Fig. 3.1 where the resulting rotation matrix $\mathbf{R}_v$ from the body-fixed frame to the inertial reference frame is given as

$$\mathbf{R}_v = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix}. \tag{3.1}$$

Using the inverse mapping condition $\mathbf{R}_{bi} = (\mathbf{R}_v)^T = (\mathbf{R}_v)^{-1}$, one can obtain the transformation matrix from the inertial frame to the body fixed reference frame.

3.2.2 Quaternions

Quaternions are hyper-complex mathematical entities used to represent rotations in three-dimensional space, where every rotation of a rigid body is equivalent to a single rotation about a fixed axis. Consider a unit vector $\vec{r} = [r_x \ r_y \ r_z]^T$ and a rotation about this vector by some value β in Fig. 3.2.

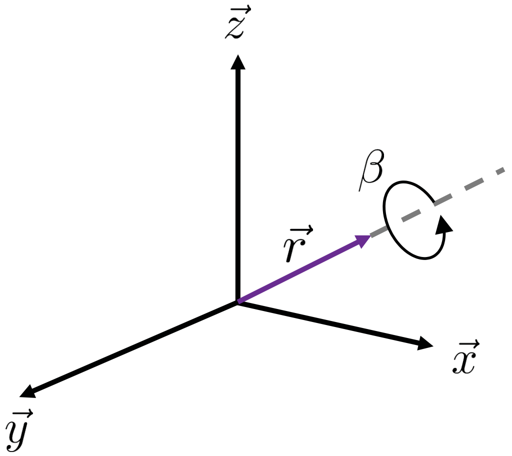


Figure 3.2: Rotation about a unit vector.

Based on this rotation, a unit quaternion $\mathbf{q} = [q_0 \ q_1 \ q_2 \ q_3]^T$ is defined as

$$\mathbf{q} = [q_0 \ q_1 \ q_2 \ q_3]^T = \begin{bmatrix} q_0 \\ \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\beta}{2}\right) \\ \vec{r}\sin\left(\frac{\beta}{2}\right) \end{bmatrix} \in \mathbb{S}^3, \tag{3.2}$$

where $\mathbb{S}^3 = \{\mathbf{q} \in \mathbb{R}^4 \mid \mathbf{q}^T\mathbf{q} = 1\}$. Quaternions consist of two components, the first being a scalar q_0 which gives the rotation angle β, and a vector $\mathbf{q}_v = [q_1 \ q_2 \ q_3]^T$ giving the quaternion imaginary components.

The norm satisfies the following

$$||\mathbf{q}|| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1, \tag{3.3}$$

making it a unitary length.

Inverse Quaternion

The conjugate of a quaternion is defined as $\mathbf{q}^* = [q_0 \ \ -\mathbf{q}_v]^T$ which can be thought of as the inverse rotation of a quaternion that describes the transformation from one coordinate frame to another $\mathbf{q}^* = \mathbf{q}(\vec{r}, -\beta)$. The formal definition of the inverse is given as $\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{||\mathbf{q}||}$ and for a unit quaternion the inverse is given as

$$\mathbf{q}^{-1} = \mathbf{q}^* = \begin{bmatrix} q_0 \\ -\mathbf{q}_v \end{bmatrix}. \tag{3.4}$$

Quaternion Multiplication

Multiplication of two quaternions $\mathbf{q}_a$ and $\mathbf{q}_b$ is performed by using the the Kronecker product, denoted as $\otimes$, where $\mathbf{q}_b = [q_{b,0} \ \ -\mathbf{q}_{b,v}]^T$ represents one rotation and $\mathbf{q}_a = [q_{a,0} \ \ -\mathbf{q}_{a,v}]^T$ represents another rotation. The combined rotation is given as [76]

$$\mathbf{q}_b \otimes \mathbf{q}_a = \begin{bmatrix} q_{b,0}q_{a,0} - \mathbf{q}_{b,v}^T\mathbf{q}_{a,v} \\ q_{b,0}\mathbf{q}_{a,v} + q_{a,0}\mathbf{q}_{b,v} + \mathbf{q}_{b,v} \times \mathbf{q}_{a,v} \end{bmatrix}, \tag{3.5}$$

where × represents the cross product, and quaternion multiplication is non-commutative.

3.2.3 Kinetics

The kinetics of a quadrotor involves the study of the forces and torques acting on it, and how it responds to various inputs. Kinetics is concerned with how these forces and torques interact with the quadrotor's mass and inertia, and how they affect the quadrotor's motion and stability.

With the quadcopter assumed to being a rigid body, the Newton-Euler equations can be used to describe its dynamics as shown

$$\mathbf{F}_b = m\dot{\mathbf{V}}_b + \mathbf{\Omega}_b \times (m\mathbf{V_b}), \tag{3.6}$$

$$\mathbf{M}_b = \mathbf{I}\dot{\mathbf{\Omega}}_b + \mathbf{\Omega}_b \times (\mathbf{I}\mathbf{\Omega}_b), \tag{3.7}$$

where

$$\mathbf{F}_b = [F_b^x \;\; F_b^y \;\; F_b^z]^T, \tag{3.8}$$

$$\mathbf{M}_b = [M_b^x \;\; M_b^y \;\; M_b^z]^T, \tag{3.9}$$

are the forces and moments acting on the quadcopter in the body frame. Term $\mathbf{\Omega}_b \in \mathbb{R}^3$ is the angular velocity in the body-fixed reference frame and $\mathbf{V}_b \in \mathbb{R}^3$ is the linear velocity in the body frame given as

$$\mathbf{V}_b = [V_b^x \;\; V_b^y \;\; V_b^z]^T, \tag{3.10}$$

$$\mathbf{\Omega}_b = [\Omega_b^x \;\; \Omega_b^y \;\; \Omega_b^z]^T. \tag{3.11}$$

The total mass of the quadcopter is labeled by $m \in \mathbb{R}$ and $\mathbf{I} \in \mathbb{R}^{3\times 3}$ is the mass moment of inertia matrix given by

$$\mathbf{I} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}. \tag{3.12}$$

The quadcopter is assumed to be symmetrical about the XZ and YZ planes. Therefore, the inertia matrix simplifies to

$$\mathbf{I} \approx \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \tag{3.13}$$

3.3 Forces and Moments

There is a wide variety of forces and moments acting on the vehicle including the thrust from actuators, gravity, aerodynamic, and payload. The total forces and moments are given by

$$\mathbf{F}_b = \mathbf{F}_b^T + \mathbf{F}_b^G + \mathbf{F}_b^A + \mathbf{F}_b^P, \tag{3.14}$$

$$\mathbf{M}_b = \mathbf{M}_b^T + \mathbf{M}_b^G + \mathbf{M}_b^A + \mathbf{M}_b^P, \tag{3.15}$$

where the subscripts T, G, A, and P stand for thrust, gravity, aerodynamics, and payload effects respectively.

3.3.1 Thrust

It has been shown that the thrust force generated by each motor propeller pair denoted as T_i where $i = \{1, 2, 3, 4\}$ is proportional to the square of its rotational speed $k\omega_i$ [44]. The proportionality constant k can be experimentally determined for a specific quadcopter.

Another method includes identifying a first order lag model [77] with its dynamics given as

$$\dot{T}_i = \frac{-T_i + T_{iR}}{\tau}, \tag{3.16}$$

where T_{iR} is the reference thrust and τ is the model time constant for each motor.

Virtual actuators are defined to allow extensive actuator configurations within the aircraft and follow a fixed-wing UAV convention. They are referred to as the thrust, aileron, elevator, and rudder respectively and for a quadrotor in the cross configuration, such as the Clover, they are defined as

$$\delta_T = T_1 + T_2 + T_3 + T_4, \tag{3.17a}$$

$$\delta_A = \frac{1}{\sqrt{2}}(-T_1 + T_2 + T_3 - T_4), \tag{3.17b}$$

$$\delta_E = \frac{1}{\sqrt{2}}(T_1 - T_2 + T_3 - T_4), \tag{3.17c}$$

$$\delta_R = T_1 + T_2 - T_3 - T_4. \tag{3.17d}$$

A mixing matrix can be defined which transforms the virtual actuator commands to the real actuator commands. From (3.17a)–(3.17d), the mixing matrix for the cross configuration quadrotor is

$$\begin{bmatrix} \delta_T \\ \delta_A \\ \delta_E \\ \delta_R \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 1 & 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}. \tag{3.18}$$

The resulting forces and moments produced by the actuators from these commands are

$$\mathbf{F}_b^T = \delta_T \vec{b}_z \tag{3.19}$$

$$\mathbf{M}_b^T = d\delta_A \vec{b}_x + d\delta_E \vec{b}_y + R_N \delta_R \vec{b}_z, \tag{3.20}$$

where d is the distance from the actuator to the center of mass, and R_N is the virtual yaw moment arm coefficient. This coefficient can be determined from rotor drag point force method, blade element theory, or from experimental results. Each method requires some form of data and physical evaluation of the target system.

Unlike many research drones, the COEX Clover is a newer one on the market with little to no modelling evaluation on the physical system. In the context of this work, MATLAB/Simulink simulations are used as a proof of concept for controller development. Given the focus on controller assessment, a comprehensive modeling effort, especially involving a detailed motor model, was beyond the scope of this testing phase. However, the absence of detailed modeling presents an avenue for future research and development endeavors. To support simulation-based evaluations, the Gazebo simulator developed by COEX replicates the looks and performance of the real Clover drone, however, the physical characteristics adopt the properties of the Iris quadrotor, including motor plugin values such as rotor drag and a time constant.

3.3.2 Ground Effect and Aerodynamic Forces

Modeling and compensating for aerodynamic forces and moments on a vehicle pose significant challenges due to their inherent complexity. Consequently, many studies in the literature tend to overlook the modeling of these forces in order to simplify the system. However, recent advancements in high-speed performance applications have underscored the necessity of considering factors such as rotor drag [78]. This thesis directs its attention towards the aerodynamic interaction referred to as ground effect, which manifests at low altitudes as previously discussed in Section 2.1.3. Additionally, the research presents a common wind model used for position dynamic simulation testing.

Ground Effect

Ground effect results in a higher thrust output at a given thrust command (see Fig. 3.3), which can have destabilizing effects at low altitudes.

Empirical thrust measurements helped develop an IGE model for multirotors, as documented in [44, 49, 58, 59], and is mathematically represented in the following

$$G(z) \approx \frac{1}{1 - \rho\left(\frac{r}{4z_r}\right)^2}, \tag{3.21}$$

where $z_r = z_m - z_d$ is the rotor altitude, or for the motion capture system, the difference between the OptiTrack marker readings z_m and constant height offset from the markers and rotors z_d. Term r is the rotor radius and ρ is a correction coefficient adapting the single rotor model from [48] to a multirotor model that varies with

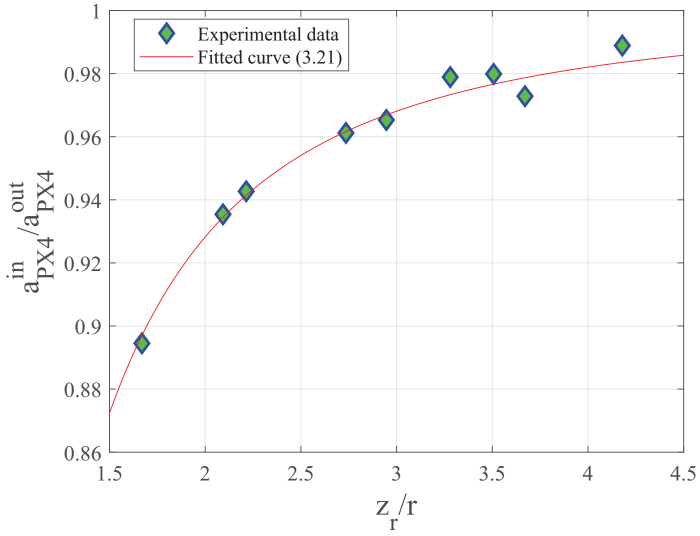


Figure 3.3: Ground effect vs. z_r/r for Clover hovering experiments ($r = 0.0635m$) from PX4 logged data analysis.

Table 3.1: Empirical ground effect model parameter from various sources.

Quadrotor	ρ
COEX Clover	4.6
Flycker MH750 [49]	8.6
Hummingbird [44]	3.4
CrazieFlie [59]	3.315

platform [59]. It has been illustrated that a prior knowledge of $G(z)$ allows for the possibility to compensate the harmful impacts of ground effect [58]. Sophisticated experimental procedures are conducted to estimate ρ to minimize the mean square error in (3.21) [49] and accurate results require measurement of the output thrust generated through setup [59] or a motor model [44]. An approximate method involving logged input acceleration a_{PX4}^{in} and filtered output acceleration a_{PX4}^{out} from the PX4 uLogs was used for the Clover results in Fig. 3.3 (more details in Appendix A) to provide a simulation ground effect model with comparable results to other literature in Table 3.1.

In contrast to other approaches, the controller design in this study avoids relying on a ground effect model. Instead, a robust controller and an adaptive observer are devised to effectively address the uncertainties associated with this phenomenon, thereby providing a comprehensive solution for system disturbances. Consequently, the ground effect is represented as an additional aerodynamic force, considered as an external system disturbance ξ_z, within the vertical dynamics of the system.

Dryden Wind Gust Model

A Dryden wind gust model [79] is used to produce time varying disturbances for the position dynamics. This is calculated with

$$v_w(t) = v_w^0 + \sum_{i=1}^{n} a_i sin(\Omega_i t + \psi_i), \tag{3.22}$$

where $v_w(t)$ is a time-dependent estimate of the wind vector in the inertial frame with Ω_i and ψ_i being the randomly selected frequencies and phase shifts respectively, n is the number of sinusoids, a_i is the amplitudes of the sinusoids and v_w^0 is the static wind vector.

As mentioned in [80], aerodynamic modelling proves to be quite difficult when aiming to express the forces and moments on the vehicle in terms of the vehicle velocity relative to the free stream. This is done through quadrotor aerodynamics, which develops thrust and moment relations from blade element theory [81] utilizing geometry, rotor thrust coefficients, and rotor torque coefficients. This theory can also be used to determine the virtual yaw moment arm coefficient [82] discussed in Section 3.3.1.

This thesis applies (3.22) directly into the position dynamics represented as an unknown external disturbance $\boldsymbol{\xi}$ in the inertial frame to be compensated for with the proposed control method.

Remark 1. *These assumptions do not encompass the full physical characteristic influence from wind gusts on a physical system, however, the performance improvements from this implementation will demonstrate viability and point towards improvements on real-time robotic systems. Also, the Gazebo simulator, being the next stage of testing, incorporates realistic wind gust plugins to evaluate the performance under these conditions before hardware testing. The disturbance moments produced are handled by the PID attitude control module in PX4.*

Payload

Consider the payload grasped by the rigid gripper in Fig. 3.4. Unlike a suspended payload, or the use of a robotic manipulator, the mass is rigidly connected to the

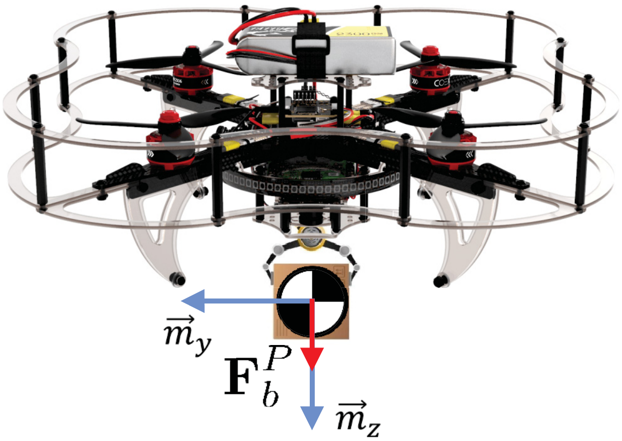


Figure 3.4: Rigid gripper grasping target payload with the COEX Clover [5].

quadrotor via a gripper and it can be modelled as a point mass attached to the body of the quadcopter. This assumption is valid when the size of the payload is small compared to the size of the quadcopter, and in this case the grasped payload mainly affects the vertical axis of the vehicle as the weight is combined with the multi-rotor.

With these assumptions, the additional payload is considered a variation of the total system mass which is treated as an additional unmodelled payload dynamic within the vertical thrust channel, or an external disturbance represented by ξ_z.

3.3.3 Kinematics

Quadrotor kinematics encompasses the study of the quadrotor's motion and positioning within three-dimensional space. This domain includes the characterization of both the quadrotor's translational and rotational motion, along with its velocity, acceleration, and orientation. The attitude of a rigid body is often defined by a rotation between a reference coordinate frame and a body-fixed coordinate frame. This transformation enables access to inertial linear velocity and position, both of which are significant aspects of interest.

Attitude Dynamics

To establish a comprehensive understanding of the quadrotor's kinematic states, it is crucial to establish a relationship between quaternions and other variables. Specifically, the alteration rate of quaternions can be associated with the quadrotor's body angular velocity as outlined in Equation (3.23)

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \otimes \bar{\mathbf{\Omega}}_b = \frac{1}{2}\begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix}\begin{bmatrix} 0 \\ \Omega_b^x \\ \Omega_b^y \\ \Omega_b^z \end{bmatrix}, \tag{3.23}$$

where $\bar{\mathbf{\Omega}}_b = [0 \ \mathbf{\Omega}_b]^T$ and $\mathbf{q} = [q_0 \ q_1 \ q_2 \ q_3]^T$ is the normed quaternion attitude vector.

The final rotational dynamics are given as

$$\mathbf{I}\dot{\mathbf{\Omega}}_b = -\mathbf{\Omega}_b \times (\mathbf{I}\mathbf{\Omega}_b) + \mathbf{M}_b^T + \mathbf{M}_b^D. \tag{3.24}$$

The term $\mathbf{M}_b^T \in \mathbb{R}^3$ is the control moment vector, $\mathbf{M}_b^D \in \mathbb{R}^3$ is the moment produced by external disturbances and $-\mathbf{\Omega}_b \times (\mathbf{I}\mathbf{\Omega}_b)$ accounts for the conservation of angular momentum.

The attitude, represented by quaternions, can be used to transform body frame dynamics to the inertial frame ones and vice versa. Consider a vector $\mathbf{H} \in \mathbb{R}^3$, which is represented by $\mathbf{H}_i$ in $\vec{i}$ and $\mathbf{H}_b$ in $\vec{b}$. The transformation from $\vec{b}$ to $\vec{i}$ is defined as

$$\begin{bmatrix} 0 \\ \mathbf{H}_i \end{bmatrix} = \mathbf{q} \otimes \begin{bmatrix} 0 \\ \mathbf{H}_b \end{bmatrix} \otimes \mathbf{q}^*. \tag{3.25}$$

The Euler angles in (3.1) suffer from singularities which can be a problem with quadrotor acrobatic maneuvers. Quaternions avoid this shortcoming and also reduce floating-point calculations benefiting onboard micro-controllers [76].

Position Dynamics

In the body frame, the forces defined in (3.8) are equal to gravity $\mathbf{F}_b^G$, the total thrust from the rotors $\mathbf{F}_b^T$ and the external aerodynamic $\mathbf{F}_b^A$ and payload $\mathbf{F}_b^P$ forces

$$m\dot{\mathbf{V}}_b + \mathbf{\Omega}_b \times (m\mathbf{V_b}) = \mathbf{F}_b^G + \mathbf{F}_b^T + \mathbf{F}_b^A + \mathbf{F}_b^P.$$

In the inertial frame, the $\mathbf{\Omega}_b \times (m\mathbf{V_b})$ term is nullified. Considering the ground effect model (3.21), then the translational dynamics are given by

$$\dot{\mathbf{p}} = \mathbf{v}, \tag{3.26}$$

$$\dot{\mathbf{v}} = g\mathbf{e}_3 - G(z)f\mathbf{R}_v\mathbf{e}_3 + \boldsymbol{\xi}(\mathbf{p}, \dot{\mathbf{p}}), \tag{3.27}$$

where $\mathbf{p} = [x\ y\ z]^T$ and $\mathbf{v} = [v_x\ v_y\ v_z]^T$ are the position and velocity in the inertial frame, respectively, f is the mass normalized collective thrust, $\mathbf{e}_3 = [0\ 0\ 1]^T \in \mathbb{R}^3$, where $-f\mathbf{R}\mathbf{e}_3 \in \mathbb{R}^3$ is the total thrust in the inertial reference frame, and $\boldsymbol{\xi} \in \mathbb{R}^3$ includes external disturbances, measurement uncertainty, and uncertainty in the process dynamics.

This work assumes $G(z)$ is an unknown aerodynamic force that increases thrust and can be included in the system uncertainty $\boldsymbol{\xi}(\mathbf{p}, \dot{\mathbf{p}})$ giving each position dynamic as

$$\ddot{x}(t) = fb_z^1 + \xi_x(x, \dot{x}), \tag{3.28}$$

$$\ddot{y}(t) = fb_z^2 + \xi_y(y, \dot{y}), \tag{3.29}$$

$$\ddot{z}(t) = g - fb_z^3 + \xi_z(z, \dot{z}), \tag{3.30}$$

with a superscript indicating an individual element of $\vec{b}_z$.

3.4 Summary

In this chapter, the mathematical model of a quadrotor with a rigid load transportation system was derived with various real world disturbances discussed. The control problem is refined to handling position dynamic tracking under force disturbances, and a block diagram of this is represented in Fig. 3.5. The derived models provide a base for simulating similar systems with different parameters and disturbance considerations to aid in the controller design.

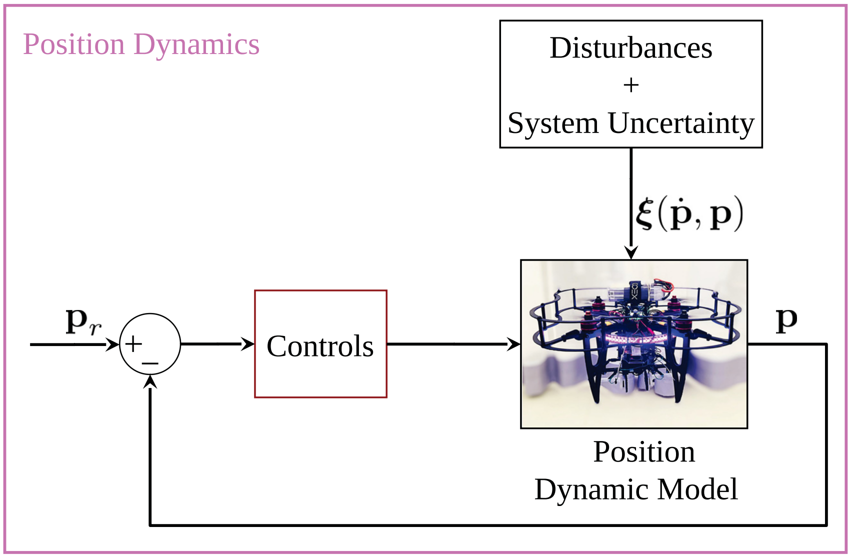


Figure 3.5: Position dynamic simulation architecture for controller prototyping and verification.

Chapter 4

System Overview

This chapter presents a comprehensive analysis of the system employed in this thesis research, encompassing software, simulation, and experimental components. It covers various aspects, including high-level control modules, sensors, simulation environments, experimental test setups, and drone load transportation designs. By delving into these areas, a thorough understanding of the entire system is achieved.

4.1 PX4 Flight Control Stack

PX4 is an open-source, full-featured, and flexible autopilot system used for controlling unmanned vehicles, such as drones, airplanes, and rovers. It is designed for high-performance computation and real-time control, making it ideal for complex aerial and ground robotic applications. PX4 consists of a modular architecture consisting of several modules, including the flight control module, communication module, and sensor module. The basic architecture of the PX4 flight stack is shown in Fig. 4.1.

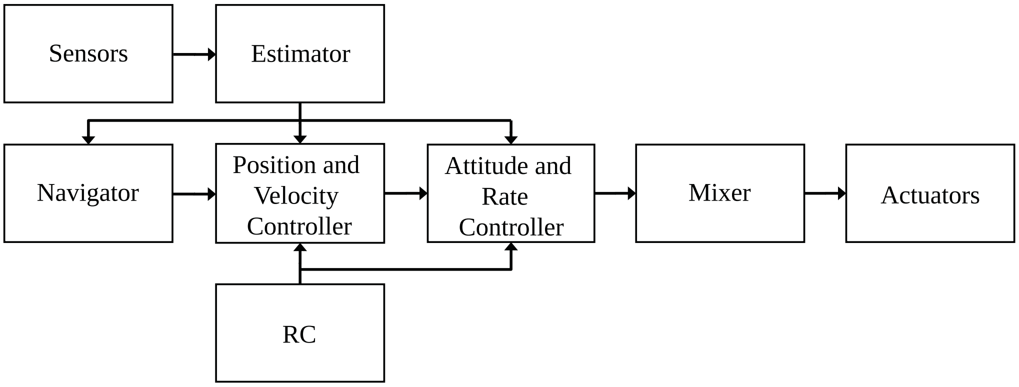

Sensors
Estimator
Navigator
Position and Velocity Controller
Attitude and Rate Controller
Mixer
Actuators
RC

Figure 4.1: PX4 system architecture.

These modules communicate with each other through a middleware called uORB.

It is a lightweight, high-performance, publish-subscribe communication system designed specifically for real-time and embedded applications making the system asynchronous.

4.1.1 Estimator

PX4 uses an Extended Kalman Filter (EKF) responsible for combining the measurements from various sensors to estimate the vehicle's state, including its position, velocity, attitude, and sensor biases. The EKF is designed to handle different time delays and data rates by making use of a delayed time horizon which is common in UAV applications. Typically, the estimator runs at 1 kHz and publishes the estimated states at 250 Hz.

4.1.2 Controllers

PX4 employs classical linear PID controllers for unmanned vehicle control. The controller architecture follows a cascaded loop structure, which includes inner attitude and outer translational controller modules. The inner attitude controllers operate at a rate of 250 Hz, which is in sync with the estimated states obtained from the EKF. The outer translational controllers operate at a rate of 50 Hz. This architecture is discussed in depth in Chapter 5.

4.1.3 Simulation

PX4 supports SITL and Hardware-in-the-Loop (HITL) simulations for unmanned vehicle system development and testing. It runs on various platforms like the NuttX Real-Time Operating System (RTOS) and other POSIX-compliant platforms such as Linux, enabling code testing on a computer before deployment. Using the Gazebo simulator, PX4 performs realistic physics simulations. In SITL, both the model and flight controllers run on the development machine, while HITL runs the model on the machine and flight controllers on the hardware. SITL tests algorithms during development, while HITL ensures their compatibility with constrained resources on the target hardware. This thesis focuses on SITL testing, but successful HITL has been achieved with modified firmware.

4.1.4 PX4 Modifications

The following modifications were made within the PX4 firmware for this project:

- The cascaded position control module was modified to use the proposed control methods in Chapter 5.

- Custom uORB topics were made to log a large variety of data for the analysis of the control improvement in Chapter 5 including observer estimation, observer estimation error, acceleration control input, and trajectory tracking error.

- Meta parameters were created to allow the user to tune the controller/observer gains through QGroundControl.

The modifications were made using PX4 firmware version 1.13.0, and this version was subsequently flashed onto the flight controller. Notably, this version corresponds to the one utilized by the Gazebo simulator, ensuring transferability of the modifications between the simulation and the physical drone.

4.2 Software Components

This section explores the high level software applications used in this work such as QGroundControl, ROS, and MAVROS. The discussion provides insights into the functionality and significance of these software applications, shedding light on their roles in enabling efficient communication and control through the implementation of the MAVLink protocol.

4.2.1 QGroundControl

QGroundControl is the open source ground control station used with PX4 to manage and monitor the COEX Clover in simulation and experiment. It provides a user-friendly interface for configuring and controlling the Clover, as well as visualizing flight data in real-time.

One of the key features of QGroundControl is its ability to perform sensor calibration and component calibration. This includes calibrating sensors such as accelerometers, gyroscopes, and magnetometers, as well as calibrating components such as motors and radio controllers. These calibrations ensure that the Clover is operating correctly and can help to prevent issues such as drifting or instability during flight.

QGroundControl also provides tools for modifying firmware such as setting controller gains and configuring the onboard EKF for sensor fusion which is an integral part of this work. Flight log analysis requires access to the ulog files, which are binary log files used by PX4 to store flight data and can be downloaded using QGroundControl. This data includes telemetry, sensor readings, and flight control commands, among other things. All of the results from SITL simulation and experiment are obtained from these logs.

QGroundControl can connect to the flight controller with a direct USB connection, or over WIFI with either a TCP or UDP Protocol. The wireless connection is illustrated in Fig. 4.2.

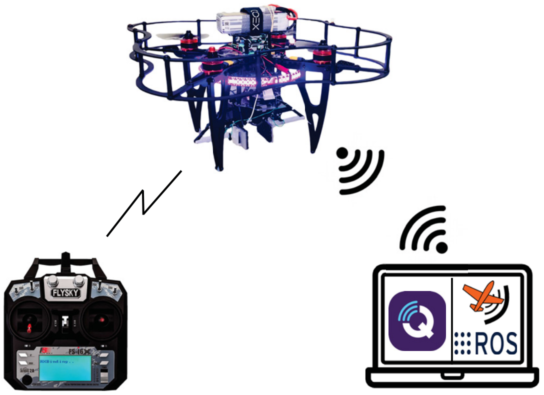


Figure 4.2: Practical flight setup with RC controller and ground station computer.

4.2.2 ROS

Robot Operating System (ROS) is an open-source framework for building robotic applications. It provides a flexible and powerful platform for developing and deploying robot software. ROS is not an operating system in the traditional sense but rather a collection of software libraries and tools that can run on various operating systems such as Linux and macOS.

One of the key features of ROS is its communication framework for robotic applications. ROS uses a publish-subscribe message system, as shown in Fig. 4.3 to enable communication between different components of a robotic system. Nodes are individual processes that perform specific tasks and communicate with each other by publishing and subscribing to topics. Topics are named buses over which messages are sent and received.

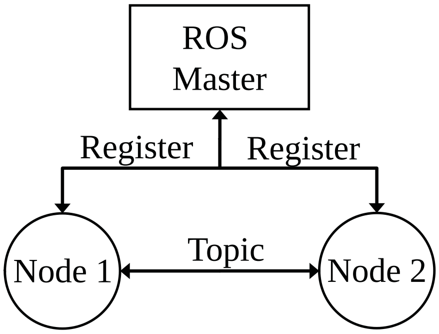


Figure 4.3: ROS communication framework.

ROS provides a wide range of libraries and tools for developing robotic applications, including libraries for computer vision, motion planning, and simulation. In this work it is used for high-level external control of the vehicle to provide advanced functionality in the areas that QGroundControl lacks with relation to control.

4.2.3 MAVLink and MAVROS

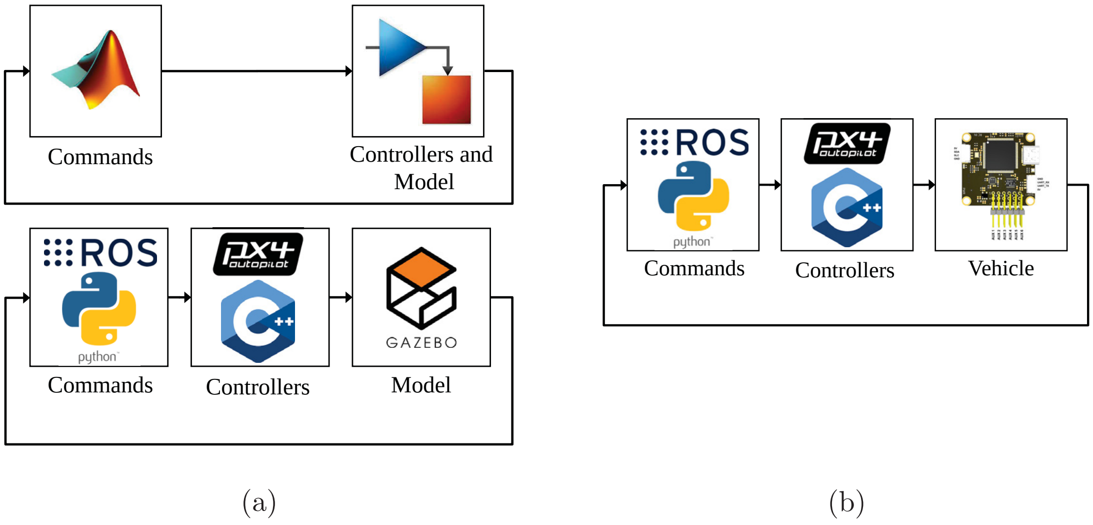
The Micro Air Vehicle Link (MAVLink) communication protocol is a lightweight and widely adopted protocol used for communication between unmanned systems, including drones, and ground control stations. It provides a common language and structure for exchanging telemetry, control commands, and other system information.

MAVLink uses a packet-based protocol where data is organized into messages. It can operate over various communication channels such as serial links, UDP, or even wireless protocols. In ROS, the MAVROS (MAVLink + ROS) package provides a MAVLink bridge, enabling communication between ROS nodes and the COEX Pix flight controller. Serving as middleware, MAVROS effectively translates ROS messages into MAVLink messages and vice versa. This translation capability allows for efficient communication and data transmission to control the drone, receive telemetry data, and implement higher-level autonomy algorithms presented in this work.

4.3 Simulation Environment

Two simulation environments are used (Fig. 4.4(a)), namely MATLAB/Simulink and a PX4 powered Gazebo simulator. PX4 is developed in the C++ programming language, making the design and testing of control algorithms slow and tedious. Therefore, making use of MATLAB/Simulink provides an environment for rapid development and implementation of the proposed control methods. This stage of testing is essentially a 'proof of concept' where quick development and testing can be completed to verify the desired characteristics before implementing in C++ and Python based environments such as PX4 and ROS.

When the control algorithm concept is accomplished in MATLAB/Simulink, it is then converted to C++ for integration into the PX4 environment and tested in the Clover Gazebo simulator (Fig. 4.5). When verified, the modified PX4 firmware is built and flashed to the COEX Pix on the COEX Clover for practical flights tests (Fig. 4.4(b)).



(a)

(b)

Figure 4.4: (a) MATLAB/Simulink and Gazebo simulation architecture block diagrams; (b) Practical system flight architecture diagram.

Remark 2. *While the COEX Clover Gazebo Simulator takes on the physical characteristics of the Iris drone, it provides a safe and reliable testing environment before hardware implementation. The same firmware that controls the drone also powers this simulator and when verified in this environment, it will most likely work in a similar fashion on hardware.*

4.3.1 MATLAB and Simulink Simulation

The quadrotor position dynamics including the ground effect disturbance model derived in Chapter 3 are modelled in Simulink where the proposed control methods in Chapter 5 are tested. It includes simple Gaussian noise models for the states, such as the Box-Muller transform, which includes high-frequency noise and a sensor bias.

An overview of the PX4 control structure is provided in Section 5.5 where a simplified version of the cascaded PID position control module is implemented in MATLAB/Simulink to provide a base comparison in the controller development.

4.3.2 COEX Clover Gazebo Simulation

Gazebo is an open-source, 3D physics-based simulator widely used in robotics research and development. The two main components include the physics engine to simulate the dynamics and interactions of objects within the simulated environment that accurately models forces, collisions, and constraints. The other is the rendering engine to generate visually rich and detailed 3D graphics of the simulated world.

Gazebo uses graphical modelling with links and joints that can range from simple geometric shapes to complex, articulated robots. Therefore the differential equation modelling in Chapter 3 is not implemented directly. The Clover model in Fig 4.5 is described using Xacro (XML Macros) files used as a preprocessor to generate Simulation Description Format (SDF) files which is an Extensible Markup Language (XML) format describing robot environments and objects.

Gazebo supports a wide range of sensors, including cameras, GPS, LiDAR, depth sensors, and Inertial Measurement Unit (IMU). The localization sensors in the COEX Clover simulator (Fig. 4.5) include a LiDAR range finder for altitude feedback and a camera for computer vision localization using ArUco markers and optical flow with the OpenCV library. These are the same ones used on the real Clover drone. Sensor noise, drift and biases are specified with noise models implemented by PX4 to replicate real-world sensors.

Gazebo offers a plugin system to enable the integration of custom controllers, algorithms, additional sensors, and interaction between objects and the environment. This enhances the capabilities of the simulator to match specific research or development requirements. A custom plugin was created to apply a force to the center

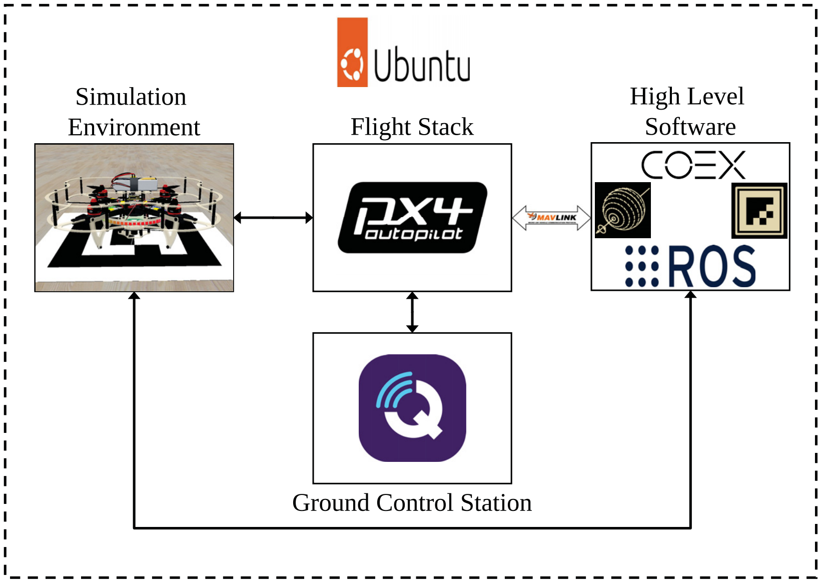


Figure 4.5: COEX Clover PX4 powered Gazebo simulation architecture.

of mass of the Clover in the inertial reference frame. This force was variable and described by a Dryden wind gust model in Section 3.3.2.

Overall, Gazebo offers a versatile and realistic simulation environment with modular architecture. Considering the PX4 control system powers the Clover drone in SITL, it provides an ideal setup before the hardware implementation.

Simple Offboard Trajectory Generator

The 'simple offboard' trajectory generation module is a high level ROS package system developed by COEX for a simple trajectory generation. It allows for high-level trajectory commands using MAVROS and reference frame transformations. This module was modified to include linear waypoint trajectory generation using feedforward components such as position, velocity, and acceleration which helps improve tracking performance of the classical PID controllers. To keep the Clovers heading aligned with the trajectory, yaw and yaw rate are also provided. All of these feedforward setpoints are needed for the model based non-linear controller derived in Section 5.3.

Complex Trajectory Generator ROS Node

Similar to the 'simple offboard' module, a ROS node is created to execute advanced trajectory generation of complex trajectories. The generated setpoints include position, velocity, acceleration, yaw, and yaw rate to allow for optimized tracking and control performance. The trajectories generated are smooth, such as the Lemniscate of Bernoulli in Section 6.2.2, meaning it is differentiable and continuous everywhere and these components are available. This smoothness characteristic is required because higher-order derivatives are needed for the model based non-linear controller used in this thesis.

4.3.3 Discussion

The methodologies of these two simulation environments exhibit significant disparities. In MATLAB/Simulink, the quadrotor model's differential equations and simplified disturbance models, outlined in Chapter 3, are employed, rendering all states measurable for analysis. Additionally, a simplified version of the PX4 cascaded controllers is utilized. On the other hand, the Gazebo environment leverages graphical modeling and a physics engine to simulate the model, along with the comprehensive PX4 flight control stack encompassing more realistic system responses and implementation.

These differences lead to the inability to make direct comparisons between controller gain values. However, system response characteristics with the same controllers and perturbations will remain the same in each environment making it possible to highlight improvements from one control method to another.

When a developer aims to achieve results that are directly comparable between MATLAB/Simulink, Gazebo simulations, and hardware testing, it becomes imperative to establish a comprehensive model of the target hardware. This necessity was addressed in [77], where advanced modeling techniques were used. These techniques included the identification of key parameters such as the multirotor's mass, mass moment of inertia characteristics, aerodynamic properties, payload specifics, and motor models. Notably, the PX4 control system was meticulously reconstructed as part of this process, facilitating transferable testing capabilities.

4.4 COEX Clover

The quadrotor utilized in this thesis for experimental validation is the COEX Clover 4.2, located within the Advanced Control and Mechatronics Laboratory at Dalhousie University. It was assembled from an array of components provided by COEX.

The COEX Clover drone platform is an innovative and versatile educational kit designed to introduce students and enthusiasts to the world of drones and robotics. It provides a comprehensive open-source framework that allows users to explore various aspects of drone technology, including programming, hardware, and customization. Like many drones used in research it consists of open source off-the-shelf components that allow us to modify and test new designs and algorithms. This platform was significantly modified and adapted with the purpose of testing advanced control algorithms for aerial load transportation tasks. The final COEX Clover quadrotor capable of accomplishing this is presented in Fig. 4.6 with labeled components.

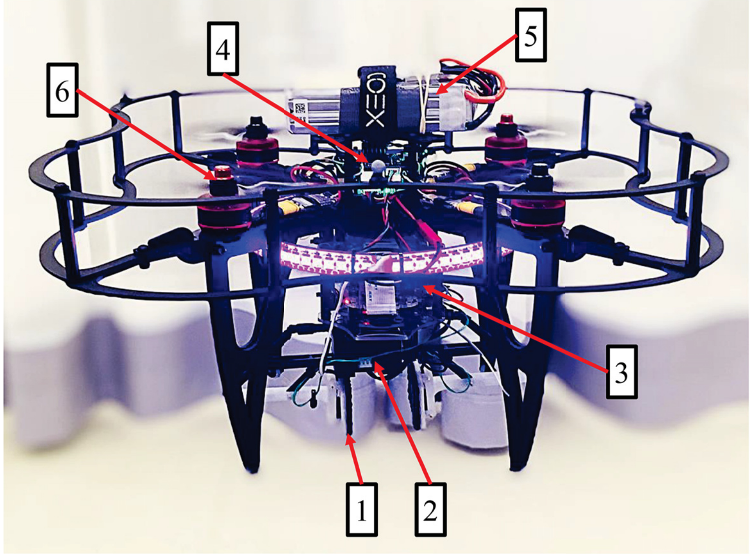


Figure 4.6: Clover Clover 4.2 platform with integrated gripper mechanism. Gripper mechanism: 1. 4-bar linkage rigid gripper mechanism, 2. Dynamixel XL330-M288-T Servomotor, Hidden: Robotis U2D2. Clover platform: 3. Raspberry Pi 4 Model B, 4. COEX Pix flight controller, 5. Tattu 2300 mAh 4S 45C LiPo battery, and 6. COEX BR2306 2400-kV motors.

The entire system responsible for autonomous missions can be divided into three distinct subsystems as in Table 4.1. These components are discussed in further detail with the following sections.

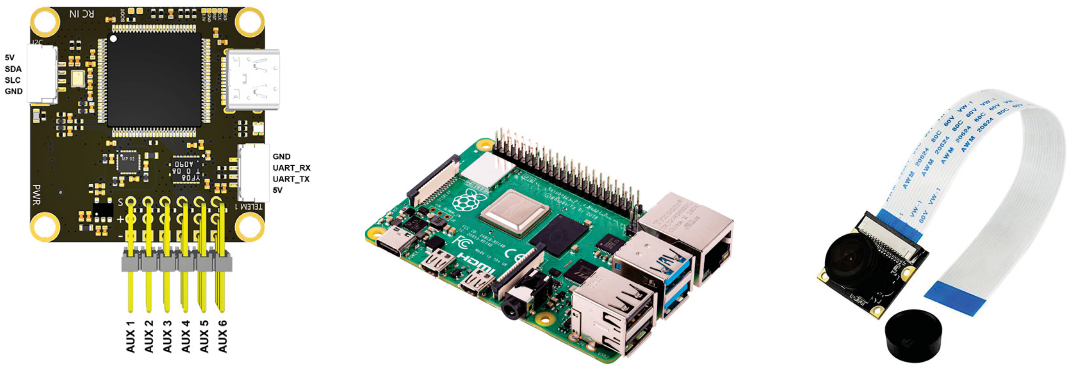
Table 4.1: Each subsystem and its corresponding components.

Clover	Gripper	Offboard
Raspberry Pi 4B (Companion computer)	Dynamixel XL330-M288-T Servomotor	4 OptiTrack Flex 13 cameras (Motion capture data)
COEX Pix (Flight Controller)	Robotis U2D2 (USB communication)	Laptops/PCs (System analysis, data transfer and logging)

4.4.1 Flight Controller

The COEX Clover drone is equipped with a programmable flight controller, which serves as the brain of the drone. Users can write and upload their own code to control the flight behavior, sensor integration, and data processing of the drone. As a result, users gain the capability to explore and experiment with advanced features such as autonomous flight, navigation, and other functionalities.

The on-board flight controller for the Clover is the COEX Pix. This flight controller is a modified Pixracer which is part of a board family optimized for small racing quads and planes. A diagram of the flight controller can be seen in Fig. 4.7(a).



(a) (b) (c)

Figure 4.7: (a) COEX Pix flight controller on-board the Clover 4.2 [6]; (b) Raspberry Pi Model 4b [7]; (c) Raspberry Pi 4 Cam OV5647 Wide Angle Lens [8].

A variety of onboard sensors are included which includes an IMU, a magnetometer and a barometer. The flight controller communicates with the Raspberry Pi 4B over a universal asynchronous receiver-transmitter (UART) serial connection. The responsibility of the companion computer is discussed in the following Section 4.4.2.

4.4.2 Onboard Computer

The Raspberry Pi in Fig. 4.7(b), serves as the COEX Clover drone's onboard companion computer. It is responsible for processing sensor feedback, computation, and communication between ROS and the flight controller via the MAVLink protocol. It also provides actuation commands for the onboard rigid gripper mechanism. The Raspberry Pi empowers autonomous mission execution such as complex trajectory tracking and grasping tasks conducted in Chapter 6.

The Clover image for the Raspberry Pi contains all of the necessary software for working with Clover and programming autonomous flights. The Clover platform is based on a Raspbian Operating System using an open Debian-repository with ROS Noetic and prebuilt binary packages for the ARMhf architecture.

4.4.3 Image Sensor

A downward facing camera (Fig. 4.7(c)), positioned on the bottom of the quadrotor, is used to provide localization feedback through various computer vision modules contained within the Clover image. An OpenCV library is presinstalled in the Clover image with additional libraries for converting from ROS messages to OpenCV images and vice versa. The camera module is designed for easy integration with onboard computers, presenting an inexpensive solution. It boasts a 5-megapixel OV5647 sensor capable of capturing images at a maximum resolution of 2592 × 1944 pixels, accompanied by a 1.3mm focal length [83].

4.5 4-bar Rigid Gripper Mechanism

For the rigid gripper mechanism attached to the bottom of the Clover [84], we explored a variety of mechanical kinematic designs before taking inspiration from the four-bar configuration [85]. The main design characteristics considered are:

- Total Weight;
- Scalability/Adaptability and Robustness;
- Fabrication;

- Secure Transportation.

Using a multi-linkage mechanism allows for increased scalability and adaptability within the Clovers physical constraints. The total workspace is reduced compared to lower linkage systems [86]. Based on a maximum takeoff weight of 1 kg provided by COEX, it was determined an additional 300 g could be added including the gripper and target object. The final gripper design can be seen in Fig. 4.8 in its open position. It is actuated by a servomotor connected to the onboard Raspberry Pi. The gripper has three distinct components: a pushbar mechanism, a four-bar linkage, and a modular gripping face.

- **Push-bar:** A push-bar mechanism was constructed to convert the horizontal plane motor rotation into the vertical plane motion. The push-bar is driven through a ball joint by a centerpiece that was designed to connect to the Dynamixel motor via four connection pins. The push bar connects to the end link of the four-bar linkage by a second ball joint.

- **Four-bar:** The four-bar linkage was built using two pairs of equal-length linkages, each set at a length of 3 cm. This configuration guaranteed that the opposing linkages remained parallel to each other regardless of the motor's position, allowing for parallel gripping surfaces.

- **Modular gripping face:** The gripper's face was designed to be modular, allowing for adaptive end effector designs. This design feature also provides for design adjustments to suit different drones and environmental requirements.

4.5.1 Parallel Plates

The four-bar linkage configuration offers modularity through dimension adjustments. To capitalize on parallel actuation within the present setup, parallel plates were used for the interaction between the gripper and objects. This ensures continuous contact between the gripping surface and the object, regardless of the motor's position.

4.5.2 Fabrication

The frame and linkages of the gripping mechanism are 3D printed out of PETG as it provides a strong, durable, and resistive material for reliable transportation under varying loads. The center piece was 3D printed out of resin because the strength and rigidity of the material allowed four small pins to be 3D printed into the piece and connected to the motor. This rigidity also allowed the ball joints in the gripper to be reliable and not prone to disengaging. The pins used in all of the linkages were made of small 1.2 mm brass rods. A rubber mesh was used on the parallel plate gripping surface as increased friction is needed with flat surface contact for secure grasping.

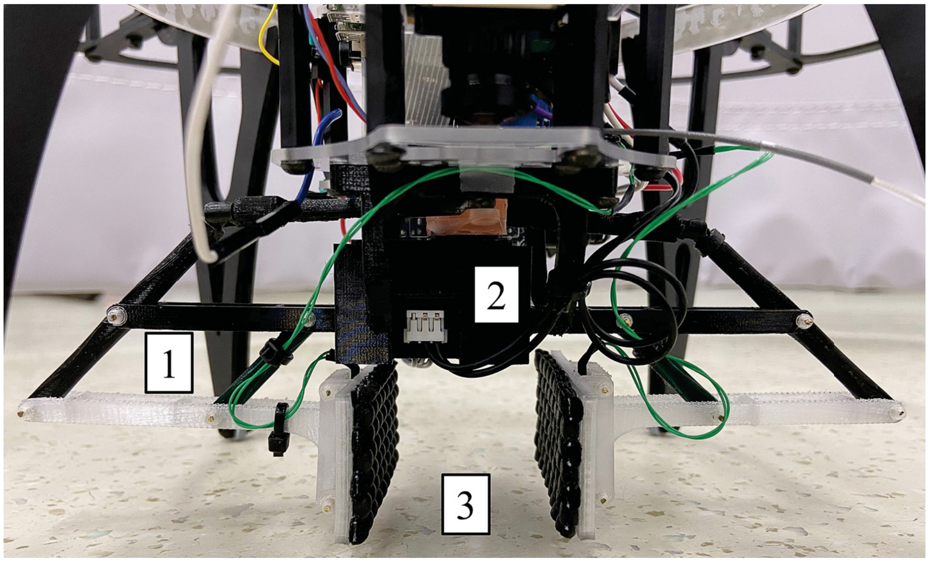


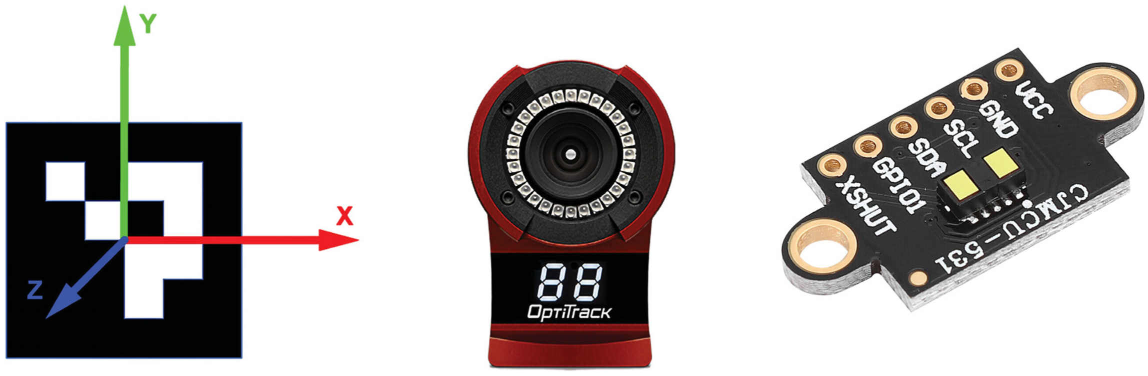
Figure 4.8: Mechanical gripper; 1) four-bar linkage; 2) servomotor; 3) parallel plate actuation.

4.5.3 Software Design and Control Architecture

The gripping mechanism is actuated by a python control module on the Raspberry Pi using position-torque mode with defined applied torque and encoder positions. This is ideal for articulated robots and grippers. The internal control architecture features a PID controller with feedforward components to improve profile tracking. Communication between the Raspberry Pi and Dynamixel motor was accomplished through the use of a U2D2 converter.

4.6 Indoor Localization

Multirotor localization encompasses the crucial task of precisely determining the position and orientation of a multirotor vehicle within its surroundings. This process



(a) (b) (c)

Figure 4.9: (a) ArUco marker OpenCV localization method [9]; (b) Flex 13 OptiTrack motion capture camera [10]; (c) CJMCU-531 VL53L1X Range Finder Optical Ranging Sensor Module [11].

relies on a combination of sensors and advanced algorithms to achieve accurate estimations of the vehicle's location. In order to streamline the focus on controller development for autonomous flight, the research implemented the following sensor-based methods both in simulation and experimental settings, effectively removing the concern for precise positioning.

4.6.1 ArUco Markers and Optical Flow

An initial form of localization for experiment involved using ArUco markers shown in Fig. 4.9(a), which is a common vision-based pose estimation method. An OpenCV library [87] runs onboard the Raspberry Pi (Fig. 4.7(b)) that allows the Clover to detect them with its downward facing camera (Fig. 4.7(c)) where algorithms estimate the pose of the marker in 3D space by analyzing its 2D projection on the camera image. Two modes of operation are preconfigured in the Clover image including

- Single marker detection and navigation.
- Map-based navigation.

The first method enables Clover to detect and navigate relative to a single marker, while the second method allows Clover to detect a pre-defined marker map layout onboard, enabling navigation relative to an entire map of markers. These methods can be used in combination, alongside optical flow analysis. Optical flow involves analyzing the motion of pixels in consecutive frames captured by the drone's camera feed, leveraging OpenCV algorithms to estimate the drone's movement and position. However, it's important to note that this method localizes the drone relative to itself rather than an external reference frame, resulting in limited accuracy.

4.6.2 LiDAR Range Finder

The SITL simulator uses marker based navigation (see Fig. 4.9(a)) in conjunction with a LiDAR laser range finder that utilizes laser beams to measure altitude by emitting short pulses of light and measuring the time it takes for the light to bounce back after hitting a surface.

The range finder model STM VL53L1X in Fig. 4.9(c) can measure distances from 0 to 4 m while ensuring high measurement accuracy. This sensor was used as the main source of altitude determination in the Gazebo simulator tests.

4.6.3 Motion Capture System

To address the requirement of precise pose feedback during indoor pick-and-place load transportation missions, and considering the obstruction caused by the rigid gripper mechanism impeding the field-of-view of downward facing camera (see Fig. 4.8), the computer vision localization methods were substituted with an OptiTrack motion capture system Fig. 4.9(b).

The OptiTrack motion capture system offers exceptional accuracy for drone localization. It is known for providing sub-millimeter level precision and low latency, enabling real-time tracking of the drone's movements. OptiTrack reflective markers are placed on the Clover to track the rigid body, they are also placed on the target object for pick-and-place missions in Section 4.8.2 to provide pose feedback of the target object to the Clovers onboard server.

Overall, the OptiTrack motion capture system provides a robust solution for drone localization, showcasing its high accuracy and real-time tracking capabilities. This system eliminates poor localization during hardware testing. An educational document on integrating the motion capture system with the COEX Clover platform was created as a CopterHack 2023 project based on this work and can be found in the following Gitbook: https://0406hockey.gitbook.io/mocap-clover/.

4.7 Network Topology

Visual feedback for the Clover and the target object are provided by an Optitrack Motion Capture System in Fig. 4.12. During flight, Optitrack cameras capture the tracker motions, then stream the data at a rate of 120 Hz. For the setup in Fig. 4.10, pose data is then fed to the Raspberry Pi using a custom User Datagram Protocol (UDP) client-server written in C++ that runs a thread and waits on data packets provided by the NatNet IP multicasting server. These data packets are then received by the client socket on the Raspberry Pi, processed into pose data and sent to the flight controller using ROS. A ground station computer is used for a variety of purposes that are listed:

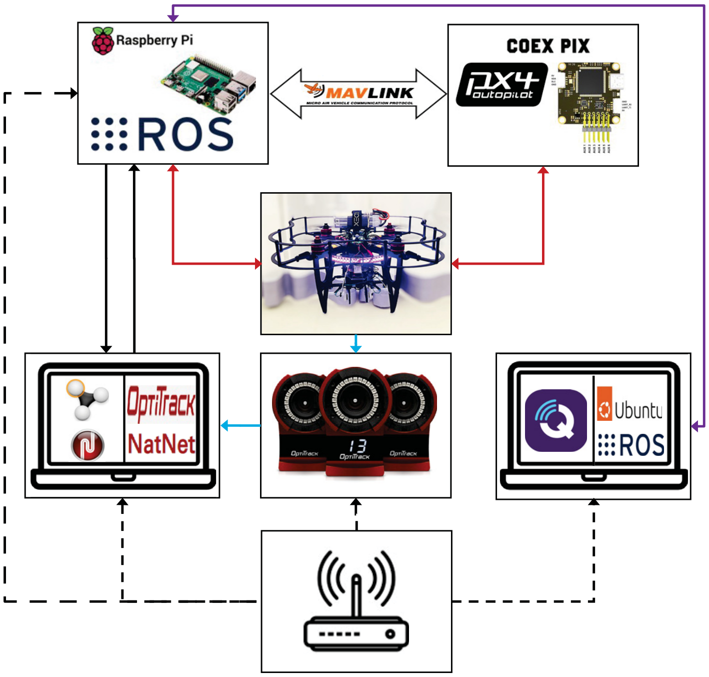


Figure 4.10: Experimental setup topology. Legend: Black dotted line is the provided local network; Black solid line is the UDP client-server drone pose transmission; Light blue line is the pose data transmission; Red line is hardware connections; Purple line is communication via secure shell protocol and ROS network communication; MAVLink arrow is communication via a MAVLink protocol.

- Connects to the Raspberry Pi via secure shell protocol (SSH) to activate the custom UDP client as well as a python script that starts the autonomous grasping mission.

- Operates as a self-check device by analyzing onboard functions and the communication between MAVROS and the PX4 using Clovers self-check function.

- Communicates with the Clover over a multimachine ROS network. This is used to analyze the EKF external pose data fusion by inspecting MAVROS topics in Rviz.

- It acts as a ground station for flight data logging and analysis, PX4 firmware modification and controller tuning using QGroundControl.

Each computer in the setup interacts using a time synchronized local network using a network time protocol server on the motion capture system computer with a dynamic host configuration protocol server for communication.

4.8 Verification of System Components

The practical flight setup utilizing the discussed components is further examined by conducting hovering tests and indoor autonomous pick-and-place missions, providing a comprehensive illustration of its application abilities within facility based operations.

4.8.1 Gripper Setup and Tests

The gripper was tasked to hold a variety of tools. The process involved placing the Clover on the ground, centering the object within the gripper, gripping it, then manually translating the Clover 0.5 m in each direction before landing again. This procedure was performed 10 times for each object in Fig. 4.11. The object mass and gripper performance results are listed in Table 4.2.

Based on the results, the gripper was able to provide a sufficient parallel plate clamping force to transport each object. Additionally, hover flight tests were conducted under these conditions with a video in Section 4.8.2. It was determined the Clovers onboard control and stability could manage relatively stable hovering up to 151 g, where 185 g showed frequent oscillations and 241 g compromised stability although this far exceeds the design limitations of 180 g. This highlights the limitations

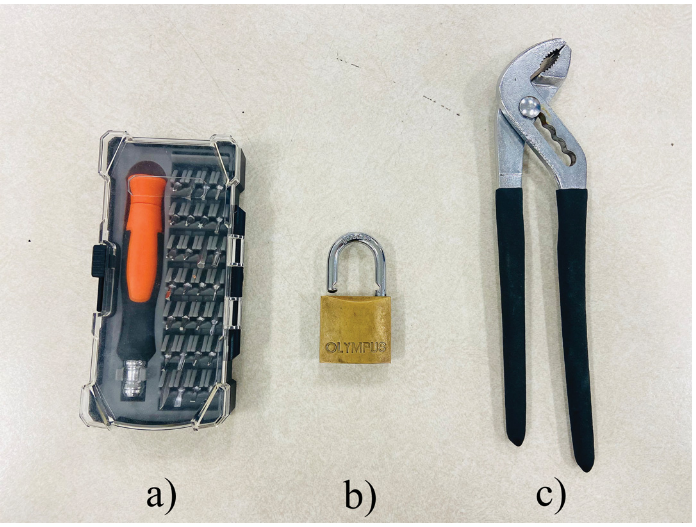


Figure 4.11: Testing objects found in factories of varying shape, size, and material. a) Toolbox; b) Lock; c) Pliers.

of PID controllers which use the integral term to compensate for additional unknown loads [39].

Another concern arises when dealing with the reliability of contact forces while transporting objects of different shapes and sizes using parallel plate friction. This challenge can be mitigated by shifting from complete force closure to form closure grasping, as discussed in [88]. In the future, it's important to explore the possibility of replacing the current parallel plates with those that facilitate form closure. This modification aims to enhance safety and reliability when handling heavier objects.

Table 4.2: Gripper performance grasping varying objects.

Object	**Mass (grams)**	**Success Rate**
Lock	151	100% (10/10)
Pliers	241	100% (10/10)
Toolbox	185	100% (10/10)

4.8.2 Complete System Testing with Autonomous Experiment

The experimental environment in Fig. 4.12 has a drop zone labelled by an ArUco marker, this is the starting and end point of the mission. The Styrofoam target object was labeled with OptiTrack facial markers to provide a target pose for the Clovers onboard server. The coordinate system is set as an X East, Y North, and Z Up system whereby the x-axis corresponds to longitudinal translation and y-axis corresponds to the lateral translation relative to the target object. ROS uses the same coordinate system and transforms data to the NED frame for the PX4 control system.

The tracking performance is quantified using the Mean Absolute Error (MAE) metric. Further information on this data analytic technique can be found in Section 6.1.

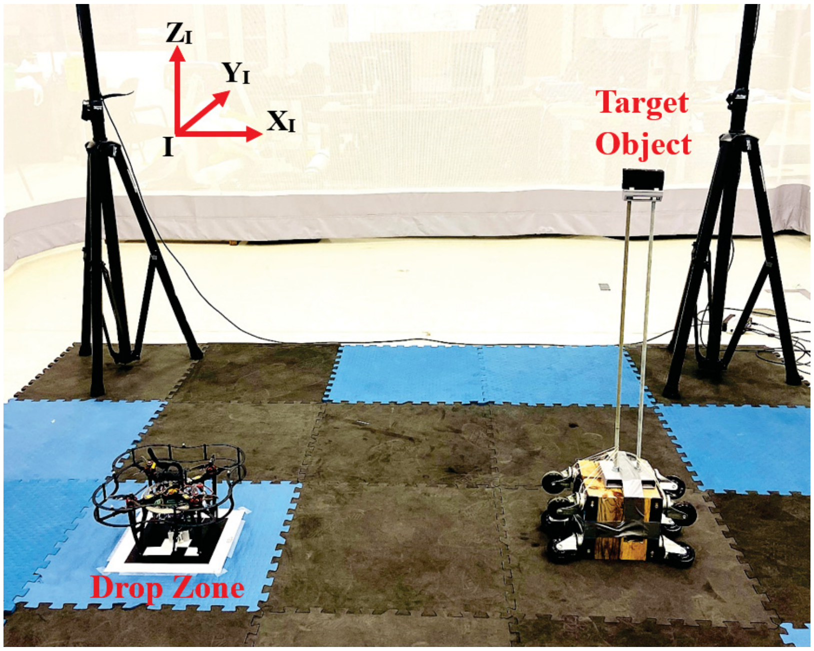


Figure 4.12: Experimental setup including the Clover, drop zone, target object and coordinate system of motion capture volume.

Results

A series of twenty autonomous grasping missions were conducted to analyze the performance of the gripper and hardware design framework. These autonomous missions are separated into three phases:

- **Phase 1:** Takeoff 1.42 m above the drop zone $\{-0.32, -0.024, 1.42\}$ m at 0.5 m/s. Track to a waypoint $\{0.835, -0.014, 1.42\}$ m located directly above the target object at 0.25 m/s and align with 0°yaw reference.

- **Phase 2:** Descend to {0.835, –0.014, 1.145} m at 0.1 m/s while maintaining 0° yaw reference. Close the gripper to grasp the object before ascending above the pickup location {0.835, –0.014, 1.42} m at 0.5 m/s.

- **Phase 3:** Track back to a waypoint directly above the drop zone {–0.32, –0.024, 1.42} m at 0.5 m/s.

    1. Descend and land into the drop zone before releasing the target object for a successful mission.

    2. For a failed target object retrieval, descend and land into the drop zone.

The position and heading of the Clover during each phase of a successful mission can be seen in Fig. 4.13. The first phase was used to analyze the drone's tracking performance provided by position and yaw setpoints. The second phase was used to analyze the drone's ability to carefully descend, keep the target object aligned in the grasping volume by maintaining yaw and retrieve the block. The final phase was used to evaluate in-flight load transportation and delivery. The success rate of each section can be seen in Table 4.3.

Table 4.3: Success rate of each phase through 20 iterations in the autonomous grasping mission.

Phase	**Success Rate**
1	100% (20/20)
2	55% (11/20)
3	1) 91% (10/11) 2) 100% (9/9)

Overall, the autonomous grasping missions were not as successful as desired. The first phase was accomplished with every attempt where the Clover accomplished sufficient tracking performance. The MAE for position during Phase 1 was {5.2, 0.81, 7.9} cm. The elevated errors with x-position and z-position were from the Clover not tracking the changing setpoints as tightly, although for this application it was more than enough. Also, the Clover was able to maintain a zero-degree yaw towards the end of Phase 1 and into the beginning of Phase 2 for gripper alignment.

The second phase was the most critical one as it involved controlled grasping of the target object. From Table 4.3, it had a 55% success rate and the MAE in position

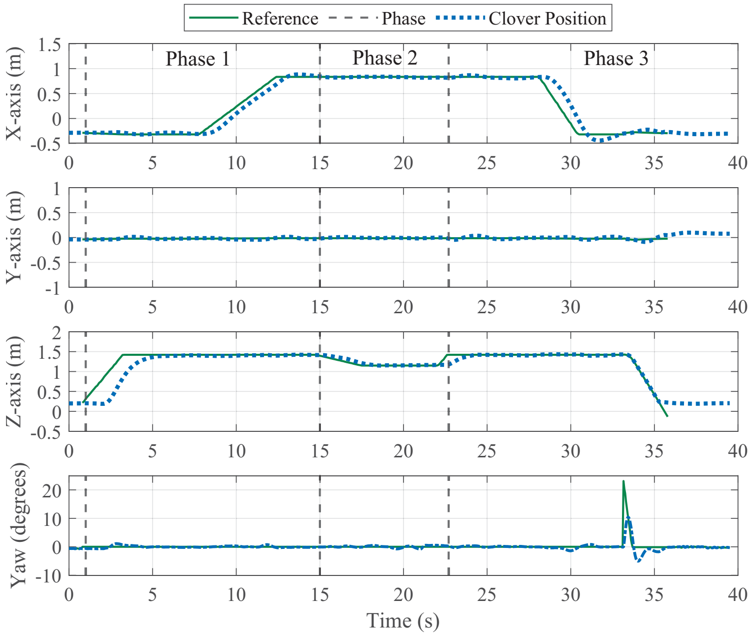


Figure 4.13: Profiles of the Clover position and heading tracking during the autonomous mission.

for Phase 2 was {0.81, 0.96, 3.74} cm and the MAE in yaw was 0.28°. While this illustrated a precise controlled descent specifically in x and y, many times the target object was missed. From observation, missed grasps were mainly caused by small deviations in the y-direction. The y-direction corresponds with the lateral portion of the grippers grasping volume being 4 cm wide in its open position. The maximum error in the y-direction was 3 cm during this phase therefore having a relatively small work envelope puts a great deal of pressure on the drones performance to ensure a successful grasp.

The third phase seen a 10/11 success rate as the Clover was able to transport the target object back to the drop zone. The styrofoam weighed 5 g so it had no flight dynamic influence the onboard controllers could not handle.

The video of the experiment can be seen in the following link: `https://youtu.be/P4aa8GVaKMM`.

4.8.3 Limitations and Discussion

Based on our observations, the primary source of failure was deviations in the horizontal plane. This resulted in the Clover descending alongside the target object, which fell outside the grasping volume, leading to missed grasps. Ensuring a consistent horizontal setpoint throughout the descent allowed the object to enter the control volume and be successfully retrieved. Notably, the y-direction exhibited a MAE of 0.96 cm during Phase 2, posing a challenge for consistently aligning a 2 cm wide block with a 4 cm wide grasping volume. To address this, increasing the work envelope, particularly by slightly extending the gripper's open position, could enhance robustness against positional errors in this configuration. The scalability of the linkage design makes such adjustments feasible.

The experimental tests involved providing position setpoints exclusively to the PX4 control system. While suitable for general linear waypoint tracking, enhanced trajectory tracking can be achieved by introducing feedforward setpoints. This approach is particularly advantageous for complex trajectory tracking scenarios. Likewise, when coupled with high-precision pose feedback from the motion capture system, precise tracking during the approach to the target object would see improvements.

The downwash generated by the Clover would exhibit different impacts if the target object were positioned closer to the ground. Firstly, there is the "thrust stealing" effect [29], and secondly, the influence of ground effects [89]. These factors can destabilize the quadcopter during low-altitude flight due to uncertainties in drone dynamics (unmodeled dynamics). While these concerns were largely mitigated with the elevated target object, minor wind disturbances arose as the Clover descended gently towards the stand, leading to increased deviations in the xy-plane.

This uncertainty is not accounted for with the PX4 PID controllers. Control methods such as adaptive [53] or robust [61] will be needed to compensate while operating under time varying external disturbances.

4.9 Summary

This chapter discussed the components and tools used in this thesis work. The toolchain consists of a variety of software components including the PX4 firmware used for flight control and ROS used for high level algorithm development and control of the system. QGroundControl serves as the ground station software used to monitor flight, modify firmware, and calibrate sensors. MATLAB/Simulink is used for rapid development of controllers and algorithms and Gazebo is used for pre-flight algorithm testing within the PX4 environment.

The COEX Clover 4.2 quadrotor within the Advanced Mechatronics Laboratory at Dalhousie University is used along with the integrated gripper mechanism design for aerial pick-and-place tasks. A variety of sensors and localization methods were discussed that were used within the simulation and the hardware system. Autonomous missions were conducted in a motion capture system volume to verify the effectiveness of the gripper design and overall system components.

Chapter 5

Flight Controller Design

The focus of this chapter is to derive and design a robust control system for the multirotor to perform various real world UAV applications in the presence of lumped varying disturbances such as complex trajectory tracking and load transportation pick-and-place operations. Lyapunov stability is used to ensure a stable performance under these conditions.

Recall, this work focuses on the influence of position tracking and control under time varying perturbations and assumes the attitude control module can track the setpoints provided. Therefore, effective tuning of the PID controllers is needed, where an adaptive auto-tuning analysis of the Clover is implemented and discussed.

5.1 Problem Formulation

In this thesis, a HOSMO is used to determine the unknown states and lumped disturbances. To design a HOSMO, the quadrotor dynamics, represented in (3.26), are divided into three second-order subsystems, out of which x, y, and z subsystems together constitute the translational subsystem. The HOSMO is designed to estimate the translational velocity $\mathbf{v} = [v_x \ v_y \ v_z]^T$ and the time-varying disturbances associated with the x, y position, and altitude z dynamics defined as $\boldsymbol{\xi}(\mathbf{p}, \dot{\mathbf{p}})$. In this section, the HOSMO design for a general second order system is presented, which can be extended to the targeted translational dynamics on similar lines. Hence, consider the general second order system model

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f(x) + h(x)u + \xi(x) \\ y &= x_1. \end{cases} \tag{5.1}$$

In the second order system above, $\mathbf{x}(t) = [x_1 \ x_2]^T \in \mathbb{R}^2$ is the system state vector,

$f(x) \in \mathbb{R}$ and $h(x) \in \mathbb{R}$ are known functions and $y = x_1$ is the measurable system output. The lumped disturbance ξ is unknown with ξ_s being its upper bound, i.e. $|\xi| \leq \xi_s$.

5.2 Higher Order Sliding Mode Observer

The HOSMO designed to estimate the dynamics x_2 along with the system disturbance ξ of system (5.1) is given

$$\begin{cases} \dot{\hat{x}}_1 & = \hat{x}_2 + \lambda_1 |\tilde{x}_1|^{2/3} sign(\tilde{x}_1) \\ \dot{\hat{x}}_2 & = \hat{x}_3 + \lambda_2 |\tilde{x}_1|^{1/3} sign(\tilde{x}_1) + f(x) + h(x)u \\ \dot{\hat{x}}_3 & = \lambda_3 sign(\tilde{x}_1), \end{cases} \tag{5.2}$$

where $\hat{x}_i,\ i = 1, ..., n+1 = 1, 2, 3$, represents the estimated state variables and disturbance. The estimation errors $\tilde{x}_1, \tilde{x}_2, \tilde{x}_3$ are defined as

$$\begin{aligned} \tilde{x}_1 &= x_1 - \hat{x}_1, \\ \tilde{x}_2 &= x_2 - \hat{x}_2, \\ \tilde{x}_3 &= x_3 - \hat{x}_3, \end{aligned} \tag{5.3}$$

respectively. Let $x_3 = \xi$, and the initial value of $\hat{x}_3$ is assumed to be zero for simplicity. From (5.1) and (5.2), the estimation error dynamics can be written as

$$\begin{cases} \dot{\tilde{x}}_1 & = \tilde{x}_2 - \lambda_1 |\tilde{x}_1|^{2/3} sign(\tilde{x}_1) \\ \dot{\tilde{x}}_2 & = -\hat{x}_3 - \lambda_2 |\tilde{x}_1|^{1/3} sign(\tilde{x}_1) + \xi_z \\ \dot{\hat{x}}_3 & = \lambda_3 sign(\tilde{x}_1). \end{cases} \tag{5.4}$$

Considering a new estimation error variable $\tilde{x}_3 = \xi - \hat{x}_3$, where the disturbance ξ is assumed to be Lipschitz and $|\dot{\xi}(t)| \leq \Delta,\ \Delta \geq \Delta_0$ (Δ_0 defined in [72, Lemma 2.1]) then the error dynamics can be written as

$$\begin{cases} \dot{\tilde{x}}_1 & = \tilde{x}_2 - \lambda_1 |\tilde{x}_1|^{2/3} sign(\tilde{x}_1) \\ \dot{\tilde{x}}_2 & = \tilde{x}_3 - \lambda_2 |\tilde{x}_1|^{1/3} sign(\tilde{x}_1) \\ \dot{\tilde{x}}_3 & = \dot{\xi}_z - \lambda_3 sign(\tilde{x}_1). \end{cases} \tag{5.5}$$

The above equation has the form of a non-recursive exact robust differentiator. Therefore, the errors $\tilde{x}_1$, $\tilde{x}_2$, and $\tilde{x}_3$ will converge to zero in a finite time $t > t_0$, absent of noise, if the gains λ_1, λ_2, and λ_3 are chosen appropriately relative to Δ. Therefore, the observer is able to find real-time robust estimations of x_1, x_2 and x_3 being exact in the absence of measurement noise. This has been proven with a quadratic Lyapunov function [68].

A diffeomorphism $y = Lx, \ 0 < L \in \mathbb{R}$ is applied to system (5.2) to robustify against disturbances $\xi(t)$ and was used in [90] for differential gain scaling. This gain scaling is used here in the following form:

$$\lambda_i = \alpha_i L^{\frac{1}{n-i+2}}, \tag{5.6}$$

which was made possible because of the homogeneity properties of the differentiators. Assuming α_i, $i = 1, 2, ..., n+1$, of differentiator (5.2) provide exact nth-order differentiation with $L = 1$ i.e. in the perturbed case ($\Delta \neq 0$). Then according to [90, Proposition 4] definition (5.6) is valid for any $L > 0$ with accuracy

$$|x_i - \hat{x}_i| \leq \mu_i L^{i/(n+1)} \epsilon^{(n-i+1)/(n+1)}, \quad i = 0, ..., n, \tag{5.7}$$

for some $\mu_i \geq 1$. The term ϵ is the maximal measurement-noise magnitude of the Lebesgue-measurable noises. It is easy to see from [90, Theorem 6] that higher order differentiators provide for much better accuracy on the lower order derivatives motivating higher order sliding modes for controllers and observers.

The parameters α_i can be rewritten in terms of Levant's parameters $\alpha_{1,3} = [3 \ 1.5 \ 1.1]^T$ proposed in [90, 91]. According to [68, Corollary 2] finite time stability with gain scaling occurs for $L \geq 1$ in the perturbed case ($\Delta \neq 0$) and with $L > 0$ in the unperturbed case ($\Delta = 0$).

In practical system applications, specifically in low altitude flight, the knowledge of this bound is unknown. Consequently, users often opt for conservative gain selections, inadvertently exacerbating chattering effects. This phenomenon is illustrated in [61], where a smoothing function was used to alleviate chattering. For these reasons, a robust adaptive law is presented in the next section to allow the gains to adapt with varying disturbances, thereby minimizing the control effort and chattering phenomena.

5.2.1 Adaptive law

According to [72, Theorem 2.2] where $|\dot{\xi}(t)| \leq \Delta, \forall t \geq 0$ and $L = \Delta/\Delta_0$ the trajectories of (5.2) converge to the origin in finite time with scaled parameters. Thus, the parameter to be tuned is L. This observation leads to the simple idea that increasing until an adequate value has been reached will provide a solution when the perturbation bound Δ is unknown. Various techniques have been implemented that let the gain grow until $|\tilde{x}| = 0$ for the ideal case [92, 93]. However, this condition is limited in practical settings due to the presence of uniformly bounded measurement noise, encompassing both high-frequency noise and low-frequency drift. Consequently, achieving $|\tilde{x}| = 0$ becomes unattainable, and the adaptive gain would grow without bound. To address this, the condition $|\tilde{x}|$ belonging to a small neighborhood of zero, also known as a threshold or dead-zone, replaces the $|\tilde{x}| = 0$ criterion. This alteration ensures adaptability halts when necessary.

A deadzone [94] is commonly used to robustify adaptive laws to high frequency noise [95] with high gain observers being sensitive to high frequency noise [96]. It has been applied to the HOSMO in [97], replacing the constant adaptive rate by one that depends on observer error. This adjustment allows for a faster convergence during the initial transient and an extra degree of freedom in gain tuning.

An inherent limitation of existing laws lies in their unidirectional nature, making them effective only in scenarios where perturbations remain constant or increase. However, in pick-and-drop applications, disturbances are often most significant close to the ground during the grasping phase under varying conditions. As the flight progresses, the size of in-flight perturbations may diminish, resulting in excessive control efforts and chattering. To tackle this challenge, we introduce the following adaptive law for the HOSMO

$$\dot{L} = \begin{cases} K_a|\tilde{x}_1|sign(|\tilde{x}_1| - \Lambda_0), & if\ L \geq 1 \\ \mu, & if\ L < 1. \end{cases} \tag{5.8}$$

The gain condition $L \geq 1$ keeps the gain above its initial value for finite time stability in the perturbed system [68, Corollary 2]. Parameter μ introduces a small rate increase to keep $L \geq 1$ for $\forall t \geq 0$ and Λ_0 is a threshold bound set by the user discussed in Section 5.4.1. Gain K_a provides an additional DOF to tune the convergence rate of the observer during the transient stage.

The idea behind this adaptation law is similar to the dead zone methods: when the system estimation is not accurate, it could be due to a too much low gain with respect to the uncertainties and perturbations therefore L adapts to lower $|\tilde{x}_1|$ within the $|\tilde{x}_1| < \Lambda_0$ boundary. However, instead of stopping the adaption law in the dead zone, the proposed law (5.8) allows the gain L to decline (while $|\tilde{x}_1| < \Lambda_0$) within the threshold. In other words, the gain L will be kept at the smallest level that allows the given accuracy of HOSMO stabilization under noisy position measurements. Also, it allows an adequate gain with respect to uncertainties/perturbations that increase and decrease in the realm of UAV pick-and-drop applications.

Remark 3. *In a system without noise, the bound can be set* $\Lambda_0 = 0$ *which simplifies the adaption law allowing finite time convergence of the estimation errors to zero.*

5.3 Super Twisting Sliding Mode Controller

One of the main drawbacks of the first order sliding mode control is the chattering phenomenon that appears in the control law. This phenomenon arises due to the presence of unmodeled system dynamics combined with the high value of the discontinuous control gain. To address this problem, authors in [98] presented the concept of High Order Sliding Mode (HOSM) where HOSM was offered as an alternative to the conventional sliding mode. By opinion of the authors, HOSM can delete the requirement to have relative degree be equal to one as with conventional sliding mode and attenuate the effects of chattering phenomena [99]. An integral aspect of HOSM is the twisting algorithm, which operates on superior derivatives of the sliding variable. This concept has led to the development of the Super-Twisting algorithm, which serves as a first-order differentiator [67], an observer [100], and a control strategy [61, 74]. As a lower-order version (second-order) of the HOSMO (third-order), this algorithm is discontinuous, exact, and robust for signals with bounded second derivatives. In this section, it will be employed as the controller.

The goal is to design a super-twisting sliding mode controller based on the estimated state information for the system (5.1) to track a reference position in the presence of external disturbances, modelling uncertainties and noisy measurements.

For this system, the position component x_1 is measurable and available while the velocity x_2 will be estimated with the observer $\hat{x}_2$. To achieve second-order sliding mode, the following convergence condition should be verified

$$s(x) = \dot{s}(x) = 0,$$

where $s(x)$ represents the sliding manifold of the following form

$$s = c_1 e_1 + \hat{e}_2 = 0, \tag{5.9}$$

where $c_1 > 0$, $e_1 = x_1 - {x_1}^d$, and $\hat{e}_2 = \hat{x}_2 - \dot{x}_1^d$. The desired position and velocity are denoted by ${x_1}^d$ and ${x_2}^d = \dot{x}_1^d$ respectively.

Taking the time derivative of the sliding manifold (5.9) and substituting the position dynamics (5.1) and observer dynamics (5.2) gives

$$\dot{s} = c_1\hat{x}_2 - c_1\dot{x}_1^d + \underbrace{\int_0^t \lambda_3 sign(\tilde{x}_1)\,\mathrm{d}t}_{\hat{x}_3} + \lambda_2|\tilde{x}_1|^{1/3} sign(\tilde{x}_1) + f(x) + h(x)u - \dot{x}_2^d. \tag{5.10}$$

The desired acceleration in the inertial frame is set as $a_r = h(x)u$ with the control input chosen such that the second order sliding mode occurs in finite time. This is given as

$$\begin{aligned} a_r = -c_1\hat{x}_2 + c_1\dot{x}_1^d - \int_0^t \lambda_3 sign(\tilde{x}_1)\,\mathrm{d}t - \lambda_2|\tilde{x}_1|^{1/3} sign(\tilde{x}_1) - f(x) + \dot{x}_2^d \\ - k_1|s|^{1/2} sign(s) - \int_0^t k_2 sign(s)\,\mathrm{d}t. \end{aligned} \tag{5.11}$$

The control law (5.11) consists of two parts, one being the equivalent control and the other being the switching control denoted as u_{eq} and u_h respectively. The equivalent control ensures that the system reaches the sliding surface and the switching control, designed on the Super-Twisting algorithm, is responsible for forcing the trajectory to slide along the sliding surface despite the bounded uncertainties and external disturbances [62]. Hence, from (5.11), one can obtain the expressions for u_{eq} and u_h as follows

$$u_{eq} = -c_1\hat{x}_2 + c_1\dot{x}_1^d - \int_0^t \lambda_3 sign(\tilde{x}_1)\,\mathrm{d}t - \lambda_2|\tilde{x}_1|^{1/3} sign(\tilde{x}_1) - f(x) + \dot{x}_2^d, \tag{5.12}$$

$$u_h = -k_1|s|^{1/2} sign(s) - \int_0^t k_2 sign(s)\,\mathrm{d}t. \tag{5.13}$$

The substitution of super-twisting control law (5.11) into (5.10) gives the reaching law

$$\dot{s} = -k_1|s|^{1/2} sign(s) - \int_0^t k_2 sign(s)\,\mathrm{d}t. \tag{5.14}$$

Defining $\eta = -k_2 \int_0^t sign(s)\,\mathrm{d}t$ one can write (5.13) in the super twisting form as

$$\begin{cases} u_h = -k_1|s|^{1/2} sign(s) + \eta \\ \dot{\eta} = -k_2 sign(s). \end{cases} \tag{5.15}$$

5.4 Stability Analysis

Consider the reaching law (5.14) in the super twisting form

$$\begin{cases} \dot{s} = -k_1|s|^{1/2} sign(s) + \eta \\ \dot{\eta} = -k_2 sign(s). \end{cases} \tag{5.16}$$

Now consider (5.9), from (5.1) and (5.2) the sliding manifold can be rewritten as

$$s = x_2 - \tilde{x}_2 - \dot{x}_1^d + c_1 e_1,$$

and knowing $\dot{x}_1 \to x_2$ after some time t_0, one can write

$$\dot{x}_1 = s + \tilde{x}_2 + \dot{x}_1^d - c_1 e_1. \tag{5.17}$$

The closed loop dynamics with the STSMC controller and AHOSMO observer can be seen as

$$\Xi : \begin{cases} \dot{x}_1 & = s + \tilde{x}_2 + \dot{x}_1^d - c_1(x_1 - x_1^d) \\ \dot{s} & = -k_1|s|^{1/2}sign(s) + \eta \\ \dot{\eta} & = -k_2 sign(s), \end{cases} \tag{5.18}$$

$$\Pi : \begin{cases} \dot{\tilde{x}}_1 & = \tilde{x}_2 - \lambda_1|\tilde{x}_1|^{2/3}sign(\tilde{x}_1) \\ \dot{\tilde{x}}_2 & = \tilde{x}_3 - \lambda_2|\tilde{x}_1|^{1/3}sign(\tilde{x}_1) \\ \dot{\tilde{x}}_3 & = \dot{\xi} - \lambda_3 sign(\tilde{x}_1). \end{cases} \tag{5.19}$$

For the observer error dynamics presented in (5.19), and controller error dynamics presented in (5.18), the following notation will be used in the stability Sections 5.4.1 and 5.4.2: for a real variable $\tilde{x} \in \mathbb{R}$ and a rational number $\rho \in \mathbb{Q}$, $\lfloor \tilde{x} \rceil^\rho = |\tilde{x}|^\rho sign(\tilde{x})$ where $\lfloor \tilde{x} \rceil^1 = \lfloor \tilde{x} \rceil = \tilde{x}$ and $\lfloor \tilde{x} \rceil^0 = sign(\tilde{x})$.

5.4.1 Analysis of Subsystem Π

It was discussed earlier in Section 5.2 that the estimation error of system Π converges to zero in finite time [68, Theorem 1] for the non adaptive gains where one could substitute $\tilde{x}_1 = \tilde{x}_2 = 0$ into (5.5) absent of measurement noise. Building upon this observation, the subsequent Theorem has been formulated to address practical applications.

Theorem 5.4.1. *Consider the nonlinear uncertain system* (5.1) *under time varying perturbations* $|\xi(t)| \leq \Delta$. *The HOSMO* (5.2) *designed to estimate the state* x_2, *and lumped disturbances* ξ *will maintain its finite-time convergence properties when combined with the adaptive law in* (5.8) *with error dynamics* (5.19) *converging to an interval* $|\tilde{x}_1| \leq \Lambda_0$ *in the presence of Lebesgue-measurable input noises.*

Proof. According to [68, Corollary 2] it can be demonstrated that the Lyapunov functions for the designed sliding-mode differentiators appropriately scale with the value of the gain L, with more details provided in [72]. Therefore, given a Lyapunov function for a specific value of the observer gain L, one can obtain a Lyapunov function for any value of L where $L \geq 1$ for $\Delta \neq 0$ and $L > 0$ for $\Delta = 0$ by adjusting the parameters accordingly.

For this reason, the value of L used to define the Lyapunov function for the non-adaptive part is omitted in the following proof. The Lyapunov function details for the non-adaptive differentiator can be found in [68].

Consider the following Lyapanov function:

$$V(\tilde{x}, L) = V_0(\tilde{x}) + \frac{1}{4}(L - L^*)^4, \tag{5.20}$$

and

$$V_0(\tilde{x}) = \boldsymbol{\vartheta}^T \boldsymbol{\Gamma} \boldsymbol{\vartheta}, \tag{5.21}$$

where $\boldsymbol{\vartheta} = \begin{bmatrix} \lfloor \tilde{x}_1 \rceil^{2/3} & \tilde{x}_2 & \lfloor \tilde{x}_3 \rceil^2 \end{bmatrix}^T$ and

$$\boldsymbol{\Gamma} = \begin{bmatrix} \gamma_1 & \frac{1}{2}\gamma_{12} & 0 \\ -\frac{1}{2}\gamma_{12} & \gamma_2 & -\frac{1}{2}\gamma_{23} \\ 0 & -\frac{1}{2}\gamma_{23} & \gamma_3 \end{bmatrix}.$$

This definition implies that $V_0(\tilde{x})$ is positive definite and radially unbounded if and only if $\boldsymbol{\Gamma} > 0$. The following conditions must be satisfied to ensure $\boldsymbol{\Gamma} > 0$ [68, Theorem 1]

$$\gamma_1 > 0, \quad \gamma_1\gamma_2 > \frac{1}{4}\gamma_{12}^2, \quad \gamma_1(\gamma_2\gamma_3 - \frac{1}{4}\gamma_{23}{}^2) > \frac{1}{4}\gamma_1{}^2\gamma_3.$$

Referencing the HOSMO error dynamics (5.19) it is possible to show the negative definiteness of $\dot{V}_0(\tilde{x})$ in [68, Theorem 1] for every value of the derivative of perturbation $|\dot{\xi}(t)| \leq \Delta$ with the following conditions

$$\gamma_{12} > 0, \quad \gamma_{23} > 0, \quad \alpha_3 L > \Delta \quad \forall t \geq 0, \tag{5.22}$$

for some positive value of α_3. Taking the derivative of the Lyaponuv function (5.20) along the trajectories of the perturbed system and using the adaptive gain conditions in (5.8) gives

$$\begin{aligned} \dot{V}(\tilde{x}, L) &= \dot{V}_0(\tilde{x}) + \dot{L}(L - L^*)^3 \\ &= \dot{V}_0(\tilde{x}) + K_a|\tilde{x}_1| sign(|\tilde{x}_1| - \Lambda_0)(L - L^*)^3. \end{aligned} \tag{5.23}$$

According to [72], estimating the convergence time requires solving the

$$\dot{V}(\tilde{x}, L) \leq -\kappa V^{\delta}(\tilde{x}, L), \tag{5.24}$$

differential inequality where $0 < \kappa$, $\delta \in \mathbb{R}$, $\delta < 1$, and $V(\tilde{x}, L)$ is positive definitive and $\dot{V}(\tilde{x}, L)$ is negative definite. It is straightforward from (5.20) that $V(\tilde{x}, L)$ is positive definite and we will study the negative definiteness of $\dot{V}(\tilde{x}, L)$ in the following cases.

From [73], we obtain the GF $W(\tilde{x}) : \mathbb{R}^n \to \mathbb{R}$ such that

$$W(\tilde{x}) = -(\partial V(\tilde{x})/\partial \tilde{x})\dot{\tilde{x}} = -\dot{V}(\tilde{x}), \tag{5.25}$$

then the differential inequality (5.24) becomes

$$W(\tilde{x}) \geq \kappa V^{\delta}(\tilde{x}), \tag{5.26}$$

where $\delta = 3/4$ because $V^{3/4}(\tilde{x})$ and $W(\tilde{x})$ must be equivalent (where they are homogeneous with homogeneity degree $\delta_{\dot{V}} = \delta_{\dot{W}} = 3$). The homogeneity of $V(\tilde{x})$ with degree $\delta_V = 4$ and weights [3, 2, 1] gives the function

$$\Phi(\tilde{x}) = \frac{W(\tilde{x})}{V^{3/4}(\tilde{x})}, \tag{5.27}$$

with homogeneous degree $\delta_\Phi = 0$. This means for every $\kappa > 0$ that

$$\Phi(\kappa^3 \tilde{x}_1, \kappa^2 \tilde{x}_2, \kappa \tilde{x}_3) = \kappa^0 \frac{W(\tilde{x})}{V^{\frac{3}{4}}(\tilde{x})} = \Phi(\tilde{x}), \tag{5.28}$$

implying all values of $\Phi(\tilde{x})$ are taken on the unity homogeneous ball $B_h = \{\tilde{x} \in \mathbb{R}^3 \mid |\tilde{x}_1|^{\frac{2}{3}} + |\tilde{x}_2| + |\tilde{x}_3|^2 = 1\}$.

Because of this, $\Phi(\tilde{x})$ can be bounded on the homogeneous ball B_h and globally as

$$0 < \eta_{min} \leq \Phi(\tilde{x}) = \frac{W(\tilde{x})}{V^{\frac{3}{4}}(\tilde{x})} = \frac{-\dot{V}(\tilde{x})}{V^{\frac{3}{4}}(\tilde{x})} \leq \eta_{max}, \tag{5.29}$$

for $\eta_{min}, \eta_{max} > 0$. This implies satisfying the inequality $\dot{V}(\tilde{x}) \leq -\eta_{min} V^{\frac{3}{4}}(\tilde{x})$ guarantees finite time convergence of the estimation.

Remark 4. *On B_h the function $V(\tilde{x})$ is continuous and different from zero and $-\dot{V}(\tilde{x})$ is different from zero and continuous everywhere except $\{\tilde{x}_1 = 0\}$. On the discontinuity point the Lyapunov function becomes $+\infty$ highlighting a positive minimum for $\Phi(\tilde{x})$ calculated with*

$$\eta = \min_{\tilde{x} \in B_h} \Phi(\tilde{x}).$$

Based on these conditions, (5.23) can be rewritten as

$$\dot{V}(\tilde{x},L) \leq -\eta V^{\frac{3}{4}}(\tilde{x}) + K_a|\tilde{x}_1|sign(|\tilde{x}_1| - \Lambda_0)(L - L^*)^3.$$

From [101, Lemma 2], it is possible to assume that there exists a positive constant L^* such that $L(0) = 1$, $L \leq L^*$ and $\forall t > 0$. It yields

$$\dot{V}(\tilde{x},L) \leq -\eta V^{\frac{3}{4}}(\tilde{x}) - \underbrace{K_a|\tilde{x}_1|sign(|\tilde{x}_1| - \Lambda_0)|L - L^*|^3}_{\Upsilon}. \tag{5.30}$$

Case 1:

Suppose that $|\tilde{x}_1| > \Lambda_0$, this gives

$$\dot{V}(\tilde{x},L) \leq -\eta V^{\frac{3}{4}}(\tilde{x}) - K_a\Lambda_0|L - L^*|^3,$$

recalling Jensen's inequality for two variables [102] where $a \geq 0$, $b \geq 0$ and $0 < p < q$ for $(a^q + b^q)^{\frac{1}{q}} \leq (a^p + b^p)^{\frac{1}{p}}$. Setting $q = 4/3$ where $p = 1$ gives

$$\begin{aligned}\dot{V}(\tilde{x},L) &\leq -[\eta^{\frac{4}{3}}V(\tilde{x}) + 4(K_a\Lambda_0)^{\frac{4}{3}}(\frac{1}{4}|L - L^*|^4)]^{\frac{3}{4}} \\ &\leq -\varrho V^{\frac{3}{4}}(\tilde{x},L),\end{aligned} \tag{5.31}$$

where $\varrho = min\{\eta, 4^{\frac{3}{4}}K_a\Lambda_0\}$, therefore finite time convergence to a domain $|\tilde{x}_1| \leq \Lambda_0$ is guaranteed from any initial condition $|\tilde{x}_1(0)| > \Lambda_0$.

Case 2:

Suppose that $|\tilde{x}_1| < \Lambda_0$. Function Υ in (5.30) can be negative, meaning $\dot{V}(\tilde{x},L)$ would be sign indefinite and closed-loop stability of the observer cannot be concluded. Therefore, $|\tilde{x}_1|$ can increase over Λ_0 where $\dot{V}(\tilde{x},L) \leq -\varrho V^{\frac{3}{4}}(\tilde{x},L)$ and V starts decreasing. However, decrease of $V(\tilde{x},L)$ can be achieved via an increase of L which happens during the transient phase. With a properly selected Λ_0, then $|\tilde{x}_1| < \Lambda_0$ remains as long as $\alpha_3 L > |\dot{\xi}(t)|$, which leads to Remark 5. Parameter Λ_0 must be selected large enough to avoid the influence of measurement noise ($|\tilde{x}_1|$ never stays lower than Λ_0 leading to large oscillations and possible divergence) but not too large to avoid low controller accuracy [101]. Guidelines for choosing Λ_0 are given in [103].

In our application, an initial value can be chosen based on the covariance of the sensor noise from the EKF within PX4. Subsequent experimental evaluation can further refine this value, setting it slightly above the measurement noise level of the corresponding position measurement output. This refinement aims to achieve steady-state observer gains.

Remark 5. *As discussed in [104, Section 3.2] the system in (5.5) cannot remain in pure sliding mode under arbitrary noises, as it would require infinitely large gains. Instead, assuming Λ_0 was selected properly i.e. the measurement noise is bounded by Λ_0, then the system will remain within the vicinity of the $\tilde{x}_2$ axis of width Λ_0, with $\tilde{x}_1 + \Lambda_0 = 0$ as the sliding surface. When $|\tilde{x}_1| \leq \Lambda_0$, the adaptive gains will dynamically reduce beginning a fine tuning phase. As a result, the overshoot of the trajectory can be further weakened until it leaves this domain [105].*

□

Remark 6. *While the Lyapunov function provided in [68] suffices in displaying the second order differentiators stability (positive definiteness of the function and negative definiteness of its derivative), it still has its disadvantages [72]. The first being it is not differentiable, and the second being it was obtained by solving non linear and hardly solvable inequalities in the function coefficients and differentiator parameters. Inspiration is taken from [73] where a smooth and differentiable Lyapanuv function is developed in [72] to obtain a new family of gains to go along with the ones proposed in [90]. These gains are much easier to compute compared to the ones above as the task involves solving linear inequality systems and the stability is reduced by using Polya's Theorem [106, 107].*

5.4.2 Analysis of Subsystem Ξ

Consider the closed loop system dynamics from (5.18):

$$\begin{cases} \dot{x}_1 = s + \tilde{x}_2 + \dot{x}_1^d - c_1(x_1 - x_1^d) \\ \dot{s} = -k_1|s|^{1/2}sign(s) + \eta \\ \dot{\eta} = -k_2 sign(s). \end{cases} \tag{5.32}$$

The s and η sub-dynamics in (5.32) form the super twisting algorithm, which leads to Theorem 5.4.2.

Theorem 5.4.2. *Suppose that for the system in (5.1), where the perturbation ξ is assumed to be Lipschitz and $|\dot{\xi}(t)| \leq \Delta$ with gains $k_1 > 0$ and $k_2 > 0$. Then, the super-twisting sliding mode controller (5.11) guarantees a second-order sliding motion ($s = \dot{s} = 0$) in a finite time with the tracking errors e_1 and $\hat{e}_2$ achieving boundedness in the presence of Lebesgue-measurable noises and converging asymptotically to zero absent of these noises by suitably selecting $c_1 \in \mathbb{R}^+$.*

Proof. Detailed analysis of the sliding motion proof can be found in [108, 109]. Where, because being paired with the higher order sliding mode observer that estimates perturbations, the proof now takes on the form of the unperturbed super-twisting algorithm case presented in the following. Consider the new state vector $\boldsymbol{\zeta}$ defined as

$$\boldsymbol{\zeta} = [\lfloor s \rceil^{1/2} \quad \eta]^T, \tag{5.33}$$

and the derivative vector is given by

$$\dot{\boldsymbol{\zeta}} = \frac{1}{|\zeta_1|} A\boldsymbol{\zeta}, \quad A = \begin{bmatrix} -\frac{k_1}{2} & \frac{1}{2} \\ -k_2 & 0 \end{bmatrix}$$

where $|\zeta_1| = |s|^{1/2}$. The following candidate Lyapunov function is considered [109]

$$\begin{aligned} V_2 &= \boldsymbol{\zeta}^T P \boldsymbol{\zeta} \\ &= 2k_2|s| + \frac{1}{2}\eta^2 + \frac{1}{2}\left(k_1\lfloor s \rceil^{1/2} - \eta\right)^2, \end{aligned} \tag{5.34}$$

with $P = P^T$ defined as

$$P = \frac{1}{2}\begin{bmatrix} 4k_2 + k_1^2 & -k_1 \\ -k_1 & 2 \end{bmatrix}. \tag{5.35}$$

Since V_2 is continuous but not locally Lipschitz (on the set $S = \{(s, \eta) \in \mathbb{R}^2 \mid s = 0\}$) due to the term $\lfloor s \rceil^{1/2}$ usual versions of Lyapunov theorem cannot be used here [110]. However, function V_2 is shown to be absolutely continuous along the state trajectories of (5.32) implying it is differentiable almost everywhere. Moreover, V_2 can still be used as a Lyapunov function on the same lines as the theorem of Zubov [69, Theorem 20.2] if it can be shown that it decreases monotonically along the trajectories of the system, and converges to zero (negative definite almost everywhere). Which has been proven in [108].

Taking the time derivative of (5.34) along the trajectories of the system (5.32) gives

$$\begin{aligned} \dot{V}_2 &= \dot{\boldsymbol{\zeta}}^T P \boldsymbol{\zeta} + \boldsymbol{\zeta}^T P \dot{\boldsymbol{\zeta}} \\ &= -|s|^{-1/2} \boldsymbol{\zeta}^T Q \boldsymbol{\zeta}, \end{aligned} \tag{5.36}$$

with $Q = Q^T$ defined as

$$Q = \frac{k_1}{2} \begin{bmatrix} 2k_2 + k_1^2 & -k_1 \\ -k_1 & 1 \end{bmatrix}. \tag{5.37}$$

The matrix $P = P^T > 0$ is the unique symmetric and positive definite solution of the Algebraic Lyapunov Equation (ALE)

$$A^T P + PA = Q, \tag{5.38}$$

that relates P and Q. Noting that A is Hurwitz if $k_1 > 0$ and $k_2 > 0$, it has been proven in [108, Theorem 1] that the stability of the equilibrium $s = 0$ is completely determined by the stability of matrix A, specifically, V_2 is positive definite, and $\dot{V}_2$ is negative definite. This holds true when $Q > 0$ (positive definite), as readily apparent from (5.36). Consequently, a collection of equivalent statements has been formulated, illustrating the stability of the super twisting algorithm [65, Theorem 1].

Therefore, the second-order sliding mode phenomena will occur and s and $\dot{s}$ will converge to zero in a finite time. □

Once $s = 0$, the closed loop dynamics (5.32) will reduce to equation (5.39) and the boundedness of the tracking error e_1 can be attained

$$\dot{e}_1 = -c_1 e_1 + \tilde{x}_2. \tag{5.39}$$

Proof. Considering another candidate Lyapunov function as

$$V_3 = \frac{1}{2} e_1^2. \tag{5.40}$$

Taking the time derivative of (5.40) gives

$$\begin{aligned}\dot{V}_3 &= e_1\dot{e}_1 = e_1(-c_1e_1 + \tilde{x}_2) = -c_1e_1^2 + e_1\tilde{x}_2 \\ &\leq -|e_1|(c_1|e_1| - \Lambda_1),\end{aligned} \tag{5.41}$$

where Λ_1 is the bound of $\tilde{x}_2$. It can be concluded from (5.41) that if $\tilde{x}_2$ is bounded and $c_1 > 0$ then e_1 is bounded. Finite time convergence of $\tilde{x}_2$ was proven in Section 5.4.1, therefore $\tilde{x}_2$ is bounded to an interval Λ_1 and the tracking stability of e_1 is maintained. □

Remark 7. *Existing literature that implemented this controller-observer pair [62, 63] in the quadcopter system did not consider system noises and the resulting observer estimation error failing to reach zero in finite time.*

It should be noted that the tracking accuracy depends on the measurement noise in the system, which directly influences the boundedness level set by the user using the threshold value Λ_0 in the adaptive gain for the observer. As mentioned in [92, Remark 5], employing larger observer gains amplifies the impact of estimation noise, rendering it challenging to achieve satisfactory outcomes with excessively high gains for the observer-based controller. This rationale underscores the utilization of the adaptive law presented in this thesis. Conversely, when measurements are devoid of noise, the estimation error's finite-time convergence goes to zero, resulting in asymptotic tracking.

The Noiseless Case:

With the estimation error converging to zero in finite time, one can substitute $\tilde{x}_2 = 0$ into (5.39) which will reduce to equation (5.42) and the asymptotic stability of tracking error e_1 can be attained

$$\dot{e}_1 = -c_1e_1. \tag{5.42}$$

Proof. Considering candidate Lyapunov function from (5.40)

$$V_3 = \frac{1}{2}e_1^2.$$

Taking the time derivative of (5.40) and substituting (5.42) gives

$$\dot{V}_3 = e_1\dot{e}_1 = e_1(-c_1e_1) = -c_1e_1^2. \tag{5.43}$$

Therefore, e_1 is asymptotically stable by suitably selecting $c_1 > 0$ where $\dot{V}_3 < 0$. This further ensures the convergence of x_1 to x_1^d. □

STSMC Convergence Time

Function (5.34) is a *strong* Lyapunov function that is continuous but not differentiable at $s = 0$. It is positive definite and radially unbounded, if $k_2 > 0$ with

$$\lambda_{min}\{P\}||\boldsymbol{\zeta}||_2^2 \leq V_2 \leq \lambda_{max}\{P\}||\boldsymbol{\zeta}||_2^2, \tag{5.44}$$

where $||\boldsymbol{\zeta}||_2^2 = |s| + \eta^2$ is the Euclidean norm of $\boldsymbol{\zeta}$ and $\lambda_{min}\{P\}$ is the minimum eigenvalue of P. This gives

$$\dot{V}_2 = -\frac{1}{|s|^{1/2}}\boldsymbol{\zeta}^T Q\boldsymbol{\zeta} \leq -\frac{1}{|s|^{1/2}}\lambda_{min}\{Q\}||\boldsymbol{\zeta}||_2^2. \tag{5.45}$$

Using (5.44), (5.45) and the fact that

$$|s|^{1/2} \leq ||\boldsymbol{\zeta}||_2 \leq \frac{V_2^{1/2}}{\lambda_{min}^{1/2}\{P\}},$$

it follows that

$$\dot{V} \leq -\gamma V^{1/2}(s),$$

where

$$\gamma = \frac{\lambda_{min}^{1/2}\{P\}\lambda_{min}\{Q\}}{\lambda_{max}\{P\}}.$$

The estimation of the convergence time can be determined since the differential equation

$$\dot{v} - \gamma v^{1/2}, \quad v(0) = v_0 \geq 0,$$

has the solution

$$v(t) = \left(v_0^{1/2} - \frac{\gamma}{2}t\right)^2. \tag{5.46}$$

It follows from the comparison Lemma [111] that $V_2(t) \leq v(t)$ when $V_2(s_0) \leq v_0$. From (5.46), one obtains that $V_2(s(t))$, and therefore $s(t)$, converges to zero in finite time with an upper bound of the time T defined as

$$T = \frac{2V_2^{1/2}(s_0)}{\gamma}. \tag{5.47}$$

5.4.3 Application of AHOSMO based STSMC for Position Control of Quadcopter

The quadcopters translational dynamics presented in Equations (3.28) - (3.30) can be represented with the following equations

$$\begin{cases} x &= v_x \\ \dot{v}_x &= b_z^1 f + \xi_x(x, \dot{x}) \end{cases} \tag{5.48}$$

$$\begin{cases} y &= v_y \\ \dot{v}_y &= b_z^2 f + \xi_y(y, \dot{y}) \end{cases} \tag{5.49}$$

$$\begin{cases} z &= v_z \\ \dot{v}_z &= g + b_z^3 f + \xi_z(z, \dot{z}). \end{cases} \tag{5.50}$$

The desired acceleration in the inertial frame is set as

$$\begin{aligned} \mathbf{a}_r &= \begin{bmatrix} b_z^1 f & b_z^2 f & b_z^3 f \end{bmatrix}^T \\ &= \begin{bmatrix} a_r^x & a_r^y & a_r^z \end{bmatrix}^T . \end{aligned} \tag{5.51}$$

In order to track a desired trajectory x_d, y_d, and z_d we will utilize the super-twisting controller based on the adaptive higher order sliding mode observer developed in Sections 5.3 and 5.2 respectively. The observer dynamics (5.2) applied to the translational and altitude dynamics (5.48) are given by

$$\begin{cases} \dot{\hat{x}}_1 &= \hat{x}_2 + \lambda_{x1}|\tilde{x}|^{2/3} sign(\tilde{x}) \\ \dot{\hat{x}}_2 &= \hat{x}_3 + \lambda_{x2}|\tilde{x}|^{1/3} sign(\tilde{x}) + a_r^x \\ \dot{\hat{x}}_3 &= \lambda_{x3} sign(\tilde{x}) \end{cases} \tag{5.52}$$

$$\begin{cases} \dot{\hat{y}}_1 &= \hat{y}_2 + \lambda_{y1}|\tilde{y}|^{2/3} sign(\tilde{y}) \\ \dot{\hat{y}}_2 &= \hat{y}_3 + \lambda_{y2}|\tilde{y}|^{1/3} sign(\tilde{y}) + a_r^y \\ \dot{\hat{y}}_3 &= \lambda_{y3} sign(\tilde{y}) \end{cases} \tag{5.53}$$

$$\begin{cases} \dot{\hat{z}}_1 &= \hat{z}_2 + \lambda_{z1}|\tilde{z}|^{2/3} sign(\tilde{z}) \\ \dot{\hat{z}}_2 &= \hat{z}_3 + \lambda_{z2}|\tilde{z}|^{1/3} sign(\tilde{z}) + g + a_r^z \\ \dot{\hat{z}}_3 &= \lambda_{z3} sign(\tilde{z}) \end{cases} \tag{5.54}$$

Now we can design the STSMC based on the estimated state information. For that, consider the sliding surface for each translation dynamics

$$\begin{cases} s_x = c_{1x}(x - x^d) + \hat{x}_2 - v_x^d \\ s_y = c_{1y}(y - y^d) + \hat{y}_2 - v_y^d \\ s_z = c_{1z}(z - z^d) + \hat{z}_2 - v_z^d \end{cases} \tag{5.55}$$

where c_{1x}, c_{1y}, $c_{1z} > 0, \in \mathbb{R}^+$. The acceleration control for (5.48) is given by

$$\begin{aligned} a_r^x = -c_{1x}\hat{x}_2 + c_{1x}\dot{x}^d - \int_0^t \lambda_{3x} sign(\tilde{x})\, \mathrm{d}t - \lambda_{2x}|\tilde{x}|^{1/3} sign(\tilde{x}) + \dot{v}_x^d \\ - k_{1x}|s_x|^{1/2} sign(s_x) - \int_0^t k_{2x} sign(s_x)\, \mathrm{d}t, \end{aligned} \tag{5.56}$$

$$\begin{aligned} a_r^y = -c_{1y}\hat{y}_2 + c_{1y}\dot{y}^d - \int_0^t \lambda_{3y} sign(\tilde{y})\, \mathrm{d}t - \lambda_{2y}|\tilde{y}|^{1/3} sign(\tilde{y}) + \dot{v}_y^d \\ - k_{1y}|s_y|^{1/2} sign(s_y) - \int_0^t k_{2y} sign(s_y)\, \mathrm{d}t, \end{aligned} \tag{5.57}$$

$$\begin{aligned} a_r^z = -c_{1z}\hat{z}_2 + c_{1z}\dot{z}^d - \int_0^t \lambda_{3z} sign(\tilde{z})\, \mathrm{d}t - \lambda_{2z}|\tilde{z}|^{1/3} sign(\tilde{z}) - g + \dot{v}_z^d \\ - k_{1z}|s_z|^{1/2} sign(s_z) - \int_0^t k_{2z} sign(s_z)\, \mathrm{d}t, \end{aligned} \tag{5.58}$$

where the position controllers above guarantee that x, y, and z track asymptotically to x^d, y^d, and z^d respectively.

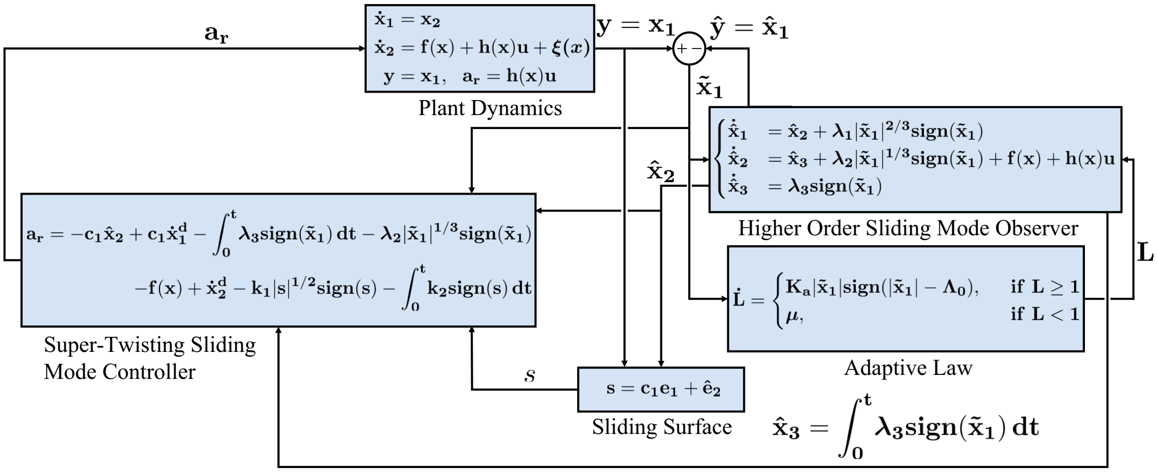

$\mathbf{a_r}$
$\dot{x}_1 = x_2$
$\dot{x}_2 = f(x) + h(x)u + \xi(x)$
$y = x_1, \quad a_r = h(x)u$
Plant Dynamics
$y = x_1$
$\hat{y} = \hat{x}_1$
+ −
$\tilde{x}_1$
$\dot{\hat{x}}_1 = \hat{x}_2 + \lambda_1|\tilde{x}_1|^{2/3}sign(\tilde{x}_1)$
$\dot{\hat{x}}_2 = \hat{x}_3 + \lambda_2|\tilde{x}_1|^{1/3}sign(\tilde{x}_1) + f(x) + h(x)u$
$\dot{\hat{x}}_3 = \lambda_3 sign(\tilde{x}_1)$
Higher Order Sliding Mode Observer
$\hat{x}_2$
$a_r = -c_1\hat{x}_2 + c_1\dot{x}_1^d - \int_0^t \lambda_3 sign(\tilde{x}_1)\,dt - \lambda_2|\tilde{x}_1|^{1/3}sign(\tilde{x}_1)$
$-f(x) + \dot{x}_2^d - k_1|s|^{1/2}sign(s) - \int_0^t k_2 sign(s)\,dt$
Super-Twisting Sliding Mode Controller
L
$\dot{L} = \begin{cases} K_a|\tilde{x}_1|sign(|\tilde{x}_1| - \Lambda_0), & \text{if } L \geq 1 \\ \mu, & \text{if } L < 1 \end{cases}$
Adaptive Law
s
$s = c_1e_1 + \hat{e}_2$
Sliding Surface
$\hat{x}_3 = \int_0^t \lambda_3 sign(\tilde{x}_1)\,dt$

Figure 5.1: Block diagram of the super-twisting sliding mode controller based on an adaptive higher order sliding observer.

5.4.4 Gain Setting

The selection of gains for the super-twisting sliding mode controller relies on the derivative of the perturbation, constrained as $|\dot{\xi}(t)| < \Delta$, leading to values such as $k_1 = 1.5\sqrt{\Delta}$ and $k_2 = 1.1\Delta$ [112]. This practice has prompted the exploration of adaptive gains in many studies [113, 114]. A straightforward explanation for this approach is rooted in the closed-loop system's behavior when using the output feedback variant of the STSMC. The closed-loop system can be represented as follows

$$\begin{cases} \dot{s} = -k_1|s|^{1/2}sign(s) + \eta \\ \dot{\eta} = -k_2 sign(s) + \dot{\xi}, \end{cases}$$

where tracking is managed under the external disturbance term. In contrast, the dynamics presented in (5.32) eliminates $\dot{\xi}$ from the super-twisting algorithm with the AHOSMO. Consequently, the gain condition shifts to the observer in (5.22), embedded within the closed-loop dynamics of (5.19). This shift underscores the motivation behind the adaptive gain featured in Section 5.2.1.

When integrating a lower-order sliding mode observer, such as the Super-Twisting Observer (STO), which exclusively estimates states, with the output feedback super twisting controller, a situation arises where second-order sliding mode cannot be achieved unless the controller incorporates discontinuity with the term $\lambda_2\text{sign}(\tilde{x}_1)$, as elaborated in [74]. This configuration mandates the consideration of two distinct gain conditions [74, 92].

A summary of conditions is provided in Table 5.1.

Table 5.1: Comparison of gain conditions for the proposed control system with other observer and non-observer variations.

Control Method	Number of gain conditions	Gain condition summary
Output-feedback STSMC	1	Design the controller gain k_2 based on the max bound of the derivative of the disturbance $\dot{\xi}(t)$.
STSMC-STO	2	Design the controller gain k_2 based on the max bound of the derivative of the disturbance $\dot{\xi}(t)$ and the observer gain λ_2 based on the explicit max bound of the direct disturbance $\xi(t)$.
STSMC-HOSMO	1	The gain condition from the controller is transferred to the observer where one needs to tune λ_3 according to the first derivative of the disturbance (see (5.22)). Therefore, it is said the HOSMO removes a gain condition from the controller with the higher order term [74] illustrated in Section 5.4.2.
STSMC-AHOSMO	0	The adaptive gain design in Section 5.2.1 is targeted to auto-tune the observer gain based on the disturbance bound conditions. This removes the gain condition for the user allowing for a more practical control system.

5.4.5 Chattering Smoothing Functions

The well known chattering phenomena with sliding mode control refers to the rapid and high-frequency switching behavior of the control signal or system states near the sliding surface. It is characterized by the control signal rapidly switching between two values or the system states oscillating excessively around the desired trajectory. The main reason cause is from the discontinuous switching function *sign* resulting in an infinitely large switching frequency [115] defined as

$$sign(s) = \begin{cases} 1 & if\ s > 0 \\ 0 & if\ s = 0 \\ -1 & if\ s < 0. \end{cases} \tag{5.59}$$

This cannot be realized by real physical systems resulting in high speed chattering being applied. Consider the super-twisting algorithm presented in (5.32) and repeated here

$$\begin{cases} \dot{s} = -k_1|s|^{1/2}sign(s) + \eta \\ \dot{\eta} = -k_2 sign(s). \end{cases} \tag{5.60}$$

The s-dynamics $k_1|s|^{1/2}sign(s)$ plotted in Fig. 5.2 has a derivative of infinity at $s = 0$ illustrating a high degree of variation of the sliding variable s at this point [116]. This characteristic is greatly increased due to unmodelled dynamics where k_1 and k_2 should be increased to maintain the tracking performance in these conditions [62].

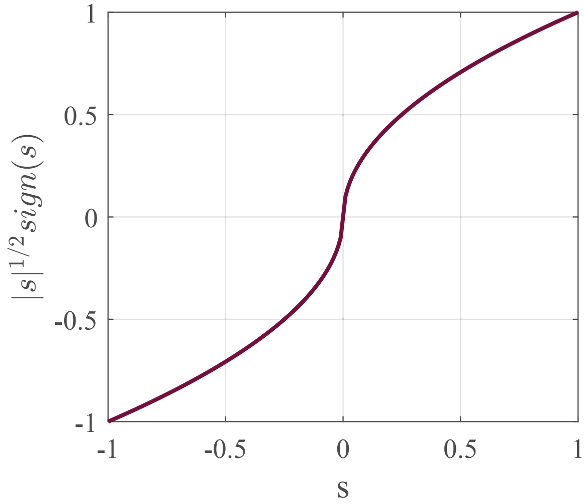


Figure 5.2: Generalized $|s|^{1/2}sign(s)$ function from the super-twisting algorithm.

A major component of this work is alleviating this chattering through a variety of methods as listed

1. Utilizing high-order sliding mode control, more specifically second order sliding mode with the super-twisting algorithm, can delete the requirement to have relative degree to be equal to one for conventional sliding mode and attenuate the chattering phenomena [117].

2. Reduce the effect of the unmodelled disturbances with the implementation of an AHOSMO allowing for reduced gain selection and chattering.

3. Minimize the gain selection on the observer by implementing an adaptive law that compensates bi-directional disturbances. The adaptive law adjusts itself downward once the disturbance is effectively compensated and tracking reaches a satisfactory level. This dual action aims to not only reduce control effort but

also mitigate the heightened sensitivity to system noise, a common characteristic of high gain observers, which ultimately reduces chattering.

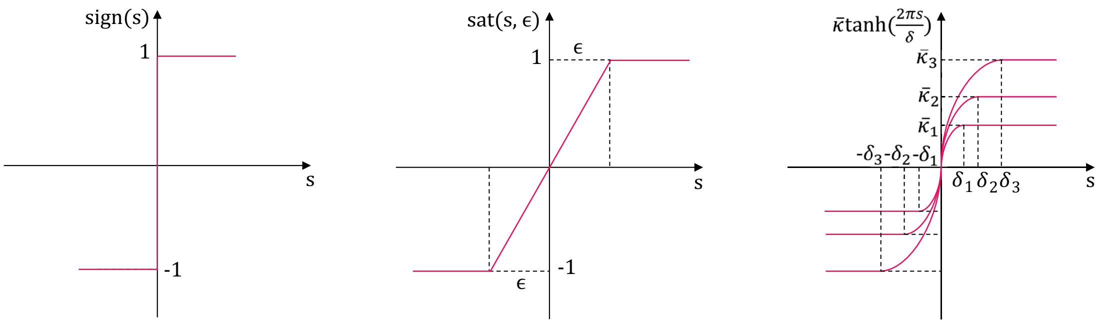
However, while this undesirable characteristic is attenuated with the proposed methods, the application of the *sign* function to real systems still carries the risk of chattering, which could potentially result in damage to electromechanical systems. To address this concern, the concept of boundary layers for the sliding surface is employed [64, 116, 118]. This boundary layer is created by replacing the $sign(s)$ function with either a $sat(s, \epsilon)$ (Fig. 5.61(b)) function given as

$$sat(s,\epsilon) = \begin{cases} sign(s), & |s| \geq \epsilon \\ \frac{s}{\epsilon}, & |s| < \epsilon, \end{cases} \tag{5.61}$$

where $\epsilon > 0$ denotes the boundary layer width, or a $tanh(s, \delta)$ (Fig. 5.61(c)) function given as

$$F(s,\delta) = \begin{cases} sign(s), & |s| \geq \delta \\ \bar{\kappa} tanh(\nu s), & |s| < \delta. \end{cases} \tag{5.62}$$

Parameter $\nu = 2\pi/\delta$, where δ represents the thickness of the boundary layer and $\bar{\kappa}$ is typically set to 1. Increasing δ tends to yield a more pronounced effect on chattering suppression. However, this decision comes with a trade-off: as δ increases, the response speed and robustness of the control algorithm tend to decrease. A smoothened super-twisting algorithm using (5.61) is illustrated in Fig. 5.4.



(a) (b) (c)

Figure 5.3: (a) Discontinuous $sign(s)$ function; (b) Saturation $sat(s, \epsilon)$ smoothing function; (c) Hyperbolic tangent $tanh(s)$ smoothing function.

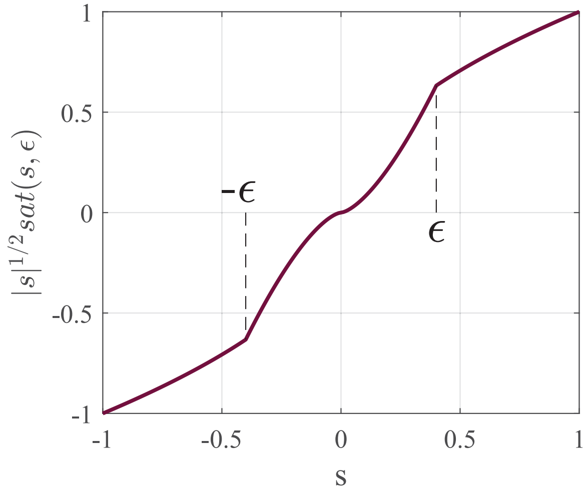


Figure 5.4: Smoothened super twisting algorithm with $sat(s, \epsilon)$ where $\epsilon = 0.4$.

For this thesis, each *sign* component within the control input (5.11) is replaced with the function in (5.62) including the disturbance and velocity estimation from the observer and the super-twisting algorithm, each term is listed:

- $\int_0^t \lambda_3 tanh(\nu\tilde{x}_1)\,\mathrm{d}t$, disturbance estimation component.
- $\lambda_2|\tilde{x}_1|^{1/3}tanh(\nu\tilde{x}_1)$, velocity estimation component.
- $-k_1|s|^{1/2}tanh(\nu s) - \int_0^t k_2 tanh(\nu s)\,\mathrm{d}t$, super-twisting algorithm or switching control component.

The switching function under the integral for the disturbance estimation and super-twisting algorithm would attenuate the chattering but cannot eliminate it [116].

5.5 PX4 Control System Architecture

PX4 makes of a cascaded loop architecture, consisting of linear PID controllers which is popular in multirotor dynamic control. This architecture comprises two key modules: the inner attitude control module and the outer position control module. The attitude control module operates in the body frame and consists of angular rate and angle controllers. The position control module operates in the inertial frame and consists of linear velocity and position controllers. The control system presented including the STSMC-AHOSMO is meant to replace the position control module.

The control system architecture is shown in Fig. 5.5. A description of the symbols depicted in Fig. 5.5 are given in Table 5.2.

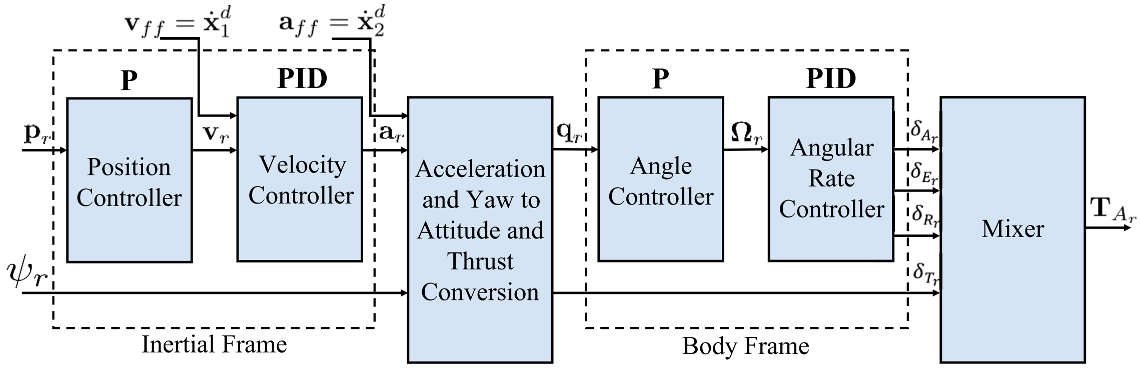


Figure 5.5: PX4 cascaded control system architecture.

Table 5.2: Control signals of the PX4 control architecture.

Control Signal	Description
$\mathbf{p}_r$	Inertial position reference
ψ_r	Yaw reference
$\mathbf{v}_r$	Inertial linear velocity reference
$\mathbf{v}_{ff}$	Inertial linear velocity feedforward
$\mathbf{a}_r$	Inertial linear acceleration reference
$\mathbf{a}_{ff}$	Inertial linear acceleration feedforward
$\mathbf{q}_r$	Unit quaternion reference
$\mathbf{\Omega}_{b_r}$	Body angular rate reference
δ_{A_r}	Aileron reference
δ_{E_r}	Elevator reference
δ_{R_r}	Rudder set point
δ_{T_r}	Total thrust reference
$\mathbf{T}_{A_r}$	Actuator input reference

5.5.1 PX4 Integration

Deriving the controllers to this point has been done by determining acceleration commands to ensure sufficient tracking of setpoints for the translational dynamics. The reason is the PX4 architecture ensures controllers can be used for various UAV platforms with little modifications. To make this possible, it must use a control system that is model-free and not reliant on modelling parameters such as mass or rotor configurations. PX4 accomplishes this by using normalized commands such as normalized accelerations, forces, and thrusts. The virtual aileron, elevator and rudder commands sent to the mixer are normalized to [-1, 1], and the virtual actuator $\delta_{T_{PX4}}$ and motor actuators $\mathbf{T}_{A_{PX4}}$ are normalized to [0, 1]. The actual thrust output from each rotor can be determined with the maximum thrust from each rotor T_{MAX} with

$$0 \leq T_i \leq T_{MAX}, \tag{5.63}$$

where i represents the individual rotor. The thrust produced by each rotor is calculated with

$$\mathbf{T}_A = T_{MAX}\mathbf{T}_{A_{PX4}}. \tag{5.64}$$

The normalized thrust commands are determined from a mixing matrix within PX4. From the code base, this matrix for a quadrotor in the X-configuration was found to be

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}_{PX4} = \begin{bmatrix} 1 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 1 \\ 1 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 1 \\ 1 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -1 \\ 1 & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -1 \end{bmatrix} \begin{bmatrix} \delta_T \\ \delta_A \\ \delta_E \\ \delta_R \end{bmatrix}_{PX4} . \tag{5.65}$$

The mixer used within PX4 contains actuator saturation, non-linearities, and attitude authority commands which are not illustrated in (5.65). It can also be noticed that the PX4 mixer matrix differs from the one developed in Section 3.3.1.

5.5.2 Acceleration and Yaw to Attitude and Thrust Conversion

The acceleration commanded from the inertial linear velocity controller is equivalent to the thrust vector of the vehicle in the body frame, as shown in Fig. 5.6. The desired acceleration of the quadrotor's body in the inertial frame of reference [119] is

$$\mathbf{a}_r = \mathbf{a}_{fb} + \mathbf{a}_{ff} - \mathbf{a}_{rd} + g\mathbf{e}_3.$$

The term $\mathbf{a}_{rd}$ represents accelerations due to rotor drag, and are ignored in this case, however flights at high speed should not ignore these considerations. An example of this can be found in [120]. As a result, each acceleration component is expressed as follows

$$\mathbf{a}_r = \mathbf{a}_{fb} + \mathbf{a}_{ff} + g\mathbf{e}_3. \tag{5.66}$$

Term $\mathbf{a}_{ff}$ is the feedforward acceleration component and $\mathbf{a}_{fb}$ is the feedback acceleration component. For the STSMC in (5.11), they are given as

$$\begin{cases} a_{ff} = \dot{x}_2^d \\ a_{fb} = -c_1\hat{x}_2 + c_1\dot{x}_1^d - \displaystyle\int_0^t \lambda_3 sign(\tilde{x}_1)\,\mathrm{d}t - \lambda_2|\tilde{x}_1|^{1/3} sign(\tilde{x}_1) \\ \qquad\qquad - k_1|s|^{1/2} sign(s) - \displaystyle\int_0^t k_2 sign(s)\,\mathrm{d}t. \end{cases} \tag{5.67}$$

Therefore the STSMC encapsulates all of the relevant accelerations discussed.

The desired orientation $\mathbf{q}_r$ for the inner control modules is determined from the acceleration commands produced from the linear velocity controllers allowing the vehicle to track a reference velocity. This transformation is described in [121]. The authors calculate the desired attitude denoted by $\vec{b}_{x_r}$, $\vec{b}_{y_r}$, and $\vec{b}_{z_r}$, from the commanded acceleration $\mathbf{a}_r$ as

$$\vec{b}_{z_r} = -\frac{\mathbf{a}_{r_{PX4}}}{|\mathbf{a}_{r_{PX4}}|}, \tag{5.68}$$

$$\vec{b}^{\mathcal{E}}_{y_r} = [-\sin\psi_r \quad \cos\psi_r \quad 0]^T, \tag{5.69}$$

$$\vec{b}_{x_r} = \frac{\vec{b}^{\mathcal{E}}_{y_r} \times \vec{b}_{z_r}}{||\vec{b}^{\mathcal{E}}_{y_r} \times \vec{b}_{z_r}||}, \tag{5.70}$$

$$\vec{b}_{y_r} = \vec{b}_{z_r} \times \vec{b}_{x_r}, \tag{5.71}$$

where $\mathcal{E}$ is an intermediate frame as a result from rotating the inertial frame about the $\vec{i}_z$ axis by the reference yaw angle, ψ_r. This transformation is illustrated in Fig. 5.6.

For a given reference command x_1^d, a total collective thrust f_{PX4} along the $\vec{i}_z$ axis in the inertial frame is calculated with the following

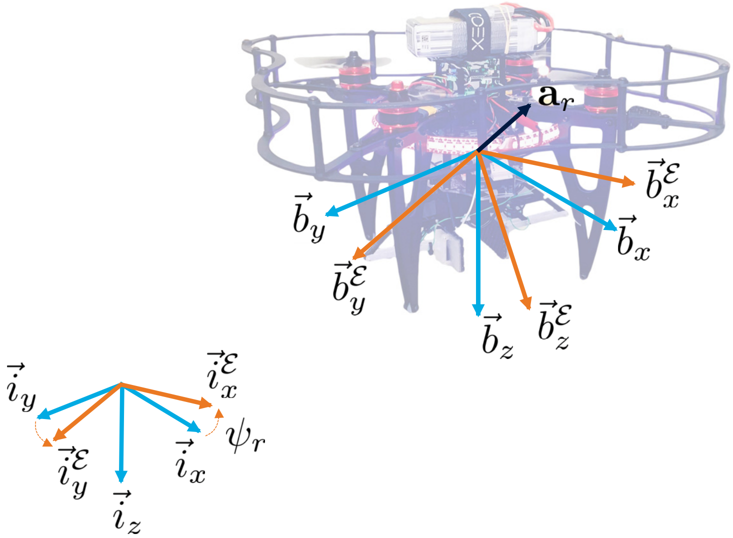

$\mathbf{a}_r$
$\vec{b}_x^{\mathcal{E}}$
$\vec{b}_y$
$\vec{b}_x$
$\vec{b}_y^{\mathcal{E}}$
$\vec{b}_z$
$\vec{b}_z^{\mathcal{E}}$
$\vec{i}_y$
$\vec{i}_x^{\mathcal{E}}$
ψ_r
$\vec{i}_y^{\mathcal{E}}$
$\vec{i}_x$
$\vec{i}_z$

Figure 5.6: Illustration of the commanded inertial acceleration.

$$f_{PX4} = (a_r^z + g)\left(\frac{f_{PX4}^{HT}}{g}\right) - f_{PX4}^{HT}, \tag{5.72}$$

with f_{PX4}^{HT} being the normalized thrust $[0, 1]$ required to hover with level orientation determined from an onboard zero order EKF. It is assumed to be equal to standard gravity as a unit conversion for acceleration commands. The thrust is restricted to a range defined by the user, where the minimal is set greater than zero to avoid the possibility of free fall. The commanded inertial acceleration is described by the vector

$$\mathbf{a}_{r_{PX4}} = [a_r^x \quad a_r^y \quad -g]^T. \tag{5.73}$$

Notice how the above vector differs from the acceleration command vector (5.51), (5.66), and (5.67) where the a_{ff}^z and a_{fb}^z components are not initially considered in $\mathbf{a}_{r_{PX4}}$. PX4 does this for a few important reasons that are listed:

- Decouple the vertical and horizontal position/velocity control.
- Typical multirotor designs prioritize generating thrust commands along the $\vec{b}_z$ direction, which allows for rapid dynamic changes in the $\vec{b}_z$ axis by manipulating collective motor thrust f. In contrast, achieving horizontal displacement

requires the entire airframe to rotate using differential thrust to align with the desired $\vec{b}_z$ axis before thrust can be effectively applied in that direction.

- This design strategy is particularly advantageous in scenarios involving very fast and impractical attitude setpoint changes, especially when accompanied by low thrust commands such as during descent. By focusing on adjusting thrust along the $\vec{b}_z$ axis, the design minimizes the need for excessive and potentially unfeasible adjustments in attitude, enhancing overall control stability and performance.

The desired rotation matrix is then described by

$$\mathbf{R}_{v_r} = [\vec{b}_{x_r} \quad \vec{b}_{y_r} \quad \vec{b}_{z_r}]^T. \tag{5.74}$$

A reference quaternion is calculated from this rotation matrix and passed to the angle controllers illustrated in Fig. 5.5. The virtual actuator thrust command is determined with the following

$$\delta_{T_{PX4}} = -||f_{PX4}\vec{b}_{z_r}||, \tag{5.75}$$

where f_{PX4} is projected along the $\vec{b}_{z_r}$ axis.

5.5.3 PX4 Auto-Tuning

The commands from the position control module, being an acceleration $\mathbf{a}_r$, are converted to an attitude setpoint in the form of a quaternion $\mathbf{q}_r$ for the inner control modules including the attitude and angular rate controllers. These controllers are the most important for stable and responsive flight, therefore to ensure sufficient performance they must be well tuned.

Angle controllers allow the vehicle to follow a given, roll, pitch, and yaw angle. PX4 implements a quaternion based angle control law from [122] which is a proportional controller with non-linear components and this work assumes it tracks the desired setpoints $\mathbf{q}_r$. Linear PID controllers are used to track angular rate setpoints $\mathbf{\Omega}_{b_r}$. The controllers command the virtual actuators δ_{i_r} where $i = \{A, E, R\}$ of the vehicle.

The COEX Clover platform provides working gains for the controllers, however, upon the integration of the gripper mechanism, the system mass moment of inertia changed and re-tuning was required. The auto-tuning module makes use of an indirect adaptive control scheme [123] represented in Fig. 5.7.

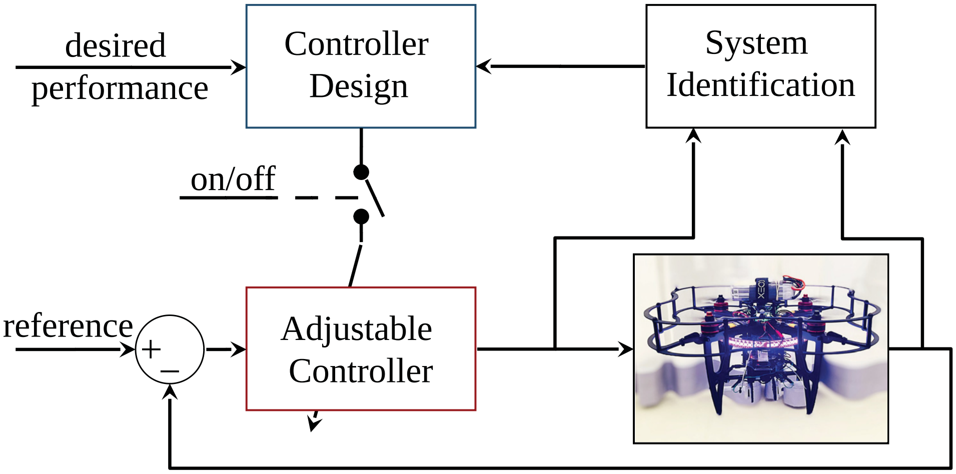


Figure 5.7: Indirect adaptive controller gain auto-tuning block diagram.

The direct adaptive control method (such as model reference adaptive control), bases its adaption mechanism on the error between the plant and reference model to drive this error to zero asymptotically by adjusting the controller parameters online. However, indirect adaptive control is done in two main stages:

1. estimate the plant parameters online;

2. estimate the controller parameters online using model based tuning techniques.

Auto-tuning uses the indirect adaptive scheme by temporarily enabling the adaption algorithm to tune the controllers.

An Auto-Regressive eXogenous (ARX) model [124] of the following form is used for system identification

$$y(k) + \sum_{i=1}^{n_a} a_i y(k-i) = \sum_{j=0}^{n_b} b_j u(k-n_k-j) \quad k = 0, 1, 2, 3, \ldots, \tag{5.76}$$

and the transfer function form is

$$A(q^{-1})y(k) = q^{-n_k} B(q^{-1})u(k) + A(q^{-1})E(k), \tag{5.77}$$

where

$$A(q^{-1}) = \sum_{i=0}^{n_a} a_i q^{-i} = 1 + a_1 q^{-1} + a_2 q^{-2} + \ldots + a_{n_a} q^{n_a}, \tag{5.78}$$

$$B(q^{-1}) = \sum_{i=0}^{n_b} b_i q^{-i} = b_0 + b_1 q^{-1} + b_2 q^{-2} + \ldots + b_{n_b} q^{n_b}, \tag{5.79}$$

with q^{-i} being the backward shift operator, n_a being the order of $A(q^{-1})$ or number of poles, n_b being the order of $B(q^{-1})$ or number of zeros, n_k being the delay or number of sampling intervals related to deadtime $n_k dt$, u being the system input, y being the system output, and E being the white noise error term. From the PX4 source code, it was determined $n_a = 2$, $n_b = 2$ and $n_k = 1$ for the identified ARX system model.

The a_i and b_i parameters are estimated for this model using the efficient Recursive Least Squares (RLS) algorithm

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} \hat{a}_1 & \hat{a}_2 & \hat{b}_0 & \hat{b}_1 & \hat{b}_2 \end{bmatrix}^T. \tag{5.80}$$

This avoids the computationally heavy matrix inversion used with the Least squares method that solves the parameters from a batch of data and assumes constant parameters as opposed to time varying ones. The RLS algorithm is presented in the following

1. Initialize variables

   - Parameter estimation: $\hat{\boldsymbol{\theta}}(0) = \mathbf{0}$,
   - Forgetting factor: $\lambda = 1 - \frac{dt}{\tau}$,
   - β, value to initialize $\mathbf{P}(0) = \beta \mathbf{I}$.

2. Constructing the design vector/data regression vector:

$$\boldsymbol{\Phi}(k) = \begin{bmatrix} -y(k-1) \\ \vdots \\ -y(k-n_a) \\ u(k-n_k) \\ u(k-n_k-1) \\ \vdots \\ u(k-n_k-n_b) \end{bmatrix}. \tag{5.81}$$

3. Covariance matrix $\mathbf{P}(k)$ update:

$$\mathbf{P}(k) = \left(\mathbf{P}(k-1) - \frac{\mathbf{P}(k-1)\boldsymbol{\Phi}(k)\boldsymbol{\Phi}(k)^T\mathbf{P}(k-1)}{\lambda + \boldsymbol{\Phi}(k)^T\mathbf{P}(k-1)\boldsymbol{\Phi}(k)} \right) \lambda^{-1}. \tag{5.82}$$

4. Calculate error estimate or innovation:

$$e(k) = y(k) - \underbrace{(\boldsymbol{\Phi}(k)^T\hat{\boldsymbol{\theta}}(k-1))}_{\hat{y}(k)}. \tag{5.83}$$

5. Determine the updated ARX model parameters:

$$\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \mathbf{P}(k)\boldsymbol{\Phi}(k)e(k). \tag{5.84}$$

Parameter $0 < \lambda < 1$ is the forgetting factor to account for the time varying weighting of the data. PX4 uses the sampling interval dt and system time constant τ in calculating the forgetting factor.

The initialization of $\mathbf{P}(0)$ is typically set with β being a very small value [125] where for $\lambda < 1$ the effect of $\mathbf{P}(0)$ reduces exponentially for a large k. However, in the case where the parameters have abrupt changes, $\mathbf{P}(k)$ is periodically set to $\beta\mathbf{I}$ with β being a large number [126]. From the PX4 code it is set to $\beta = 10^3$.

The covariance matrix $\mathbf{P}(k)$ is constructed from various components, with one of them being the gain vector $\mathbf{g}(k)$, which is expressed as

$$\mathbf{g}(k) = \frac{\mathbf{P}(k-1)\boldsymbol{\Phi}(k)}{\lambda + \boldsymbol{\Phi}(k)^T\mathbf{P}(k-1)\boldsymbol{\Phi}(k)}. \tag{5.85}$$

Remark 8. *The equations for the RLS algorithm show resemblance to the* Kalman Filter *although they focus on parameter estimation instead of state estimation.*

With the identified system model, the PID gains for the angular rate controllers are determined with a Generalized Minimum Variance Control (GMVC) law from

[127]. The GMVC algorithm is

$$\begin{cases} \rho & = \frac{dt}{\sigma} \\ \mu & = 0.25(1-\delta) + 0.51\delta \\ p_1 & = -2\mathrm{e}^{\frac{-\rho}{2\mu}} \cos\left(\frac{\sqrt{4\mu-1}}{2\mu}\rho\right) \\ p_2 & = \mathrm{e}^{\frac{\rho}{\mu}} \\ e_1 & = -a_1 + p_1 + 1 \\ f_0 & = -a_1 e_1 - a_2 e_1 + a_2 \\ f_2 & = a_2 e_1 \\ \nu & = \gamma + (e_1 + 1)(b_0 + b_1 + b_2) \end{cases} \tag{5.86}$$

where dt is the sampling interval in seconds, σ is the desired closed-loop rise time in seconds set by the user, δ is the sampling index (between 0 and 2) with 0 being critical and 1 is Butterworth. Parameter γ is known as the 'detuning' coefficient, this is related to gain k_c only, where you can increase to detune the controller. Finally, the angular rate PID gains can be set as

$$\begin{cases} k_c & = \frac{1}{\nu}(f_1 + 2f_2) \\ k_i & = \frac{1}{T_i} = -\frac{f_1+2f_2}{f_0+f_1+f_2} dt \\ k_d & = T_d = -\frac{f_2}{f_1+2f_2} dt. \end{cases} \tag{5.87}$$

The proportional gain for the attitude controllers k_q is determined with the following empirical rule

$$k_q k_c 1.047 = 1, \tag{5.88}$$

meaning an error of 60° or $1.047 rad$ should produce the maximum control output. The value of k_q is constrained to $k_q \in [2, 6.5]$. The complete auto-tuning system identification diagram is presented in Fig. 5.8.

This auto-tuning process was applied to the COEX Clover in experiment to determine the controller gains. The RLS time varying parameter estimation for the ARX model can be seen in Fig. 5.9. Resulting models for each dynamic are presented in Table 5.3.

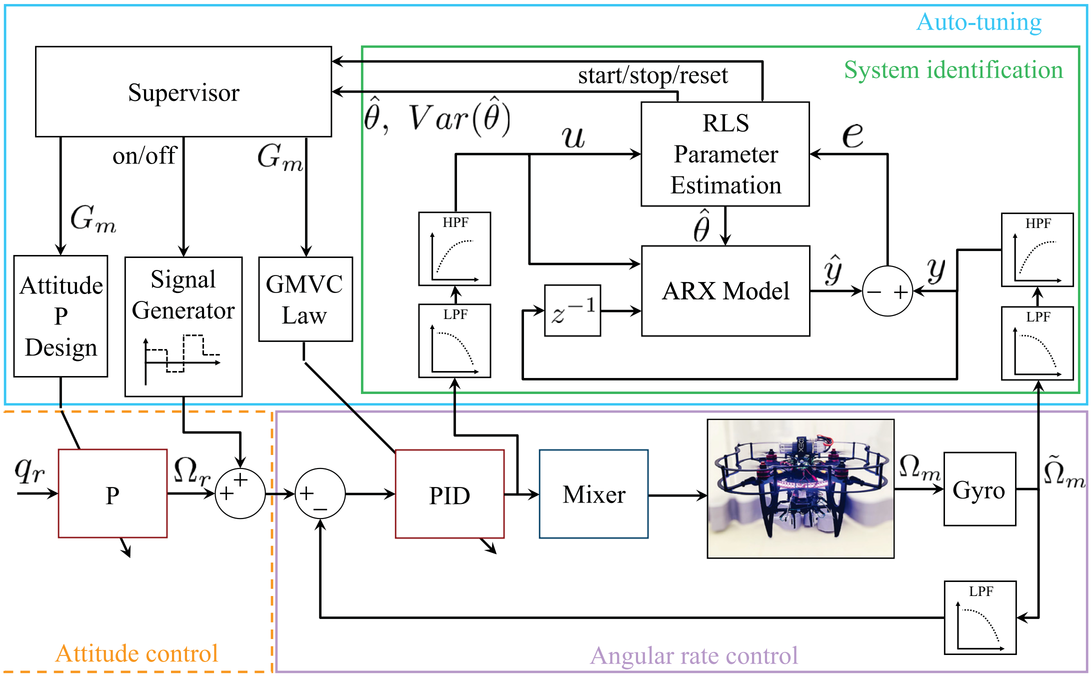


Figure 5.8: Complete systems identification and auto-tuning process for the angular rate PID controllers and P attitude controllers.

Table 5.3: System identified ARX model for each angular rate dynamic for the COEX Clover.

Dynamic	$\hat{a}_1$	$\hat{a}_2$	$\hat{b}_0$	$\hat{b}_1$	$\hat{b}_2$	dt
Roll Rate	-1.68	0.676	0.301	0.447	0.258	0.0093
Pitch Rate	-1.64	0.639	0.292	0.479	0.265	0.0093
Yaw Rate	-1.57	1.05	0.928	0.479	-1.09	0.0093

Remark 9. *The models identified in Table 5.3 can be simulated in MATLAB/Simulink to test various responses and offline gain tuning.*

The PID gain variation for angular rates controllers and P gain variation for the angle controllers is illustrated in Fig. 5.10. The filtered input u and filtered output y of the system is plotted in Fig. 5.11(a) with the innovation $y - \hat{y}$ plotted in Fig. 5.11(b). The system model estimate predicts the system angular rate very well from the small innovation.

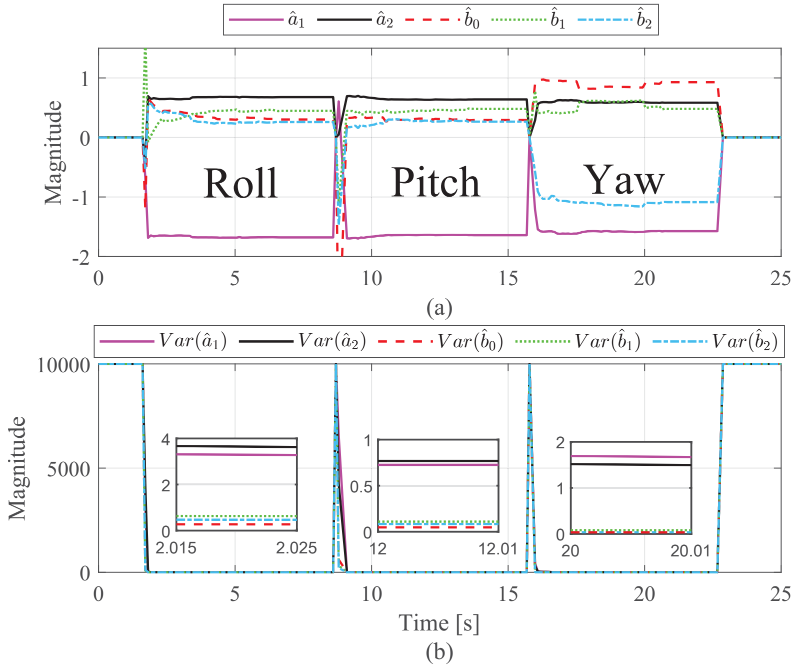


Figure 5.9: System identified ARX model parameter profiles: (a) Time varying ARX parameters through each roll, pitch, and yaw system identification phase; (b) Variance of each estimated time varying ARX parameter during each phase.

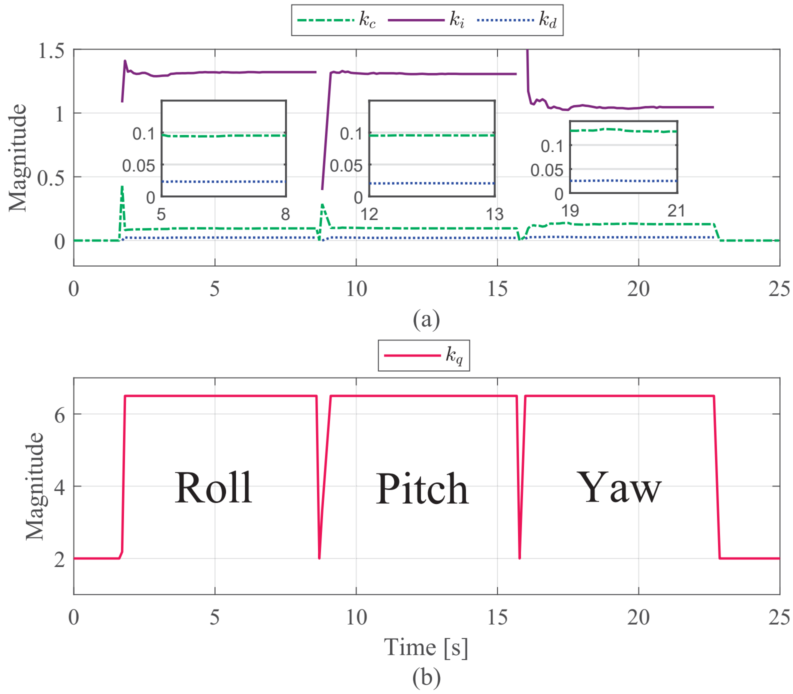


Figure 5.10: Time varying controller gain profiles: (a) Time varying angular rate PID controller gains during each roll, pitch, and yaw auto-tuning phase; (b) Time varying attitude P controller gains during each roll, pitch, and yaw auto-tuning phase.

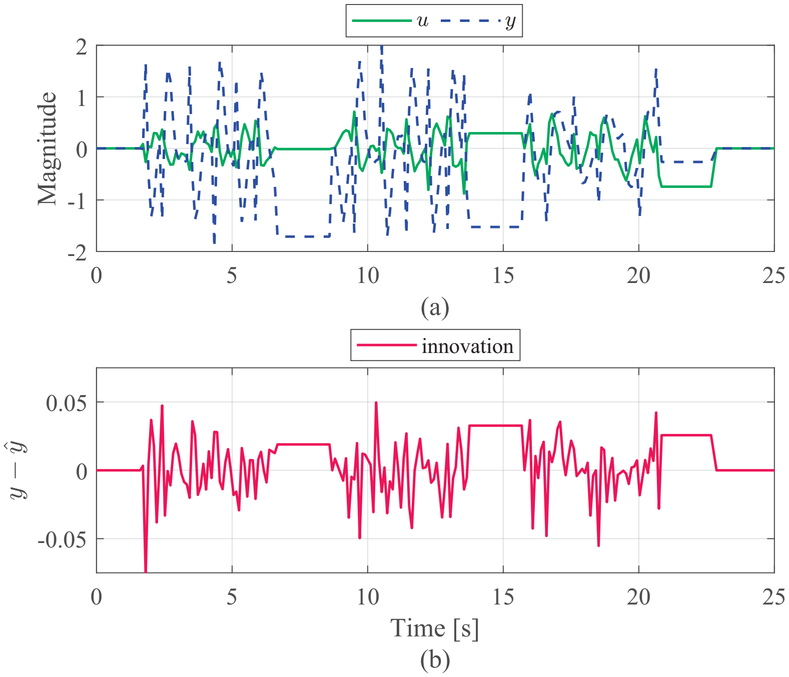


Figure 5.11: System input and output information profiles: (a) Filtered input and output data of the Clover system used for system identification; b) Innovation or difference between estimated model output and actual Clover output from the Gyro.

5.6 Summary

This chapter explored the development of a model-based control system, consisting of a super-twisting sliding mode controller based on an adaptive higher order sliding mode observer. The adaptive law is introduced and its integration with the HOSMO is validated through rigorous Lyapunov stability analysis, demonstrating the observer's finite time stability properties in the presence of external perturbations and measurement noises. The theoretical analysis yields gain conditions that emphasize the benefits of the observer and adaptive law in practical applications. To address the chattering phenomenon, where the switching frequency of the *sign* function cannot be realized in physical systems, final control modifications are discussed, including the implementation of a smoothing function.

The PX4 control system is then presented, outlining the cascaded PID controllers and the integration of the STSMC-AHOSMO as a replacement for the position control module within the PX4 environment. Additionally, the integration of the gripper mechanism into the UAV system necessitates the re-tuning of the attitude and angular rate controllers. The chapter provides a comprehensive overview of the indirect adaptive auto-tuning algorithm, followed by the presentation of auto-tuning results for the COEX Clover.

Chapter 6

Simulation and Experimental Results

6.1 Performance Metrics

This section discusses the various performance metrics used to compare results in both simulation and hardware testing.

For the trajectory tracking results, performance comparison is quantified with the MAE, which computes the average magnitude of errors during trajectory tracking, giving equal weight to all deviations. This metric is calculated as follows

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |x^d - x_1|. \tag{6.1}$$

Two metrics are used to evaluate control effort performance between the manually tuned control system STSMC-HOSMO and adaptive one STSMC-AHOSMO, particularly its magnitude and chattering analysis. The first is the Root Mean Square (RMS) given by

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n} a_r^2}, \tag{6.2}$$

which simply gives insights into the magnitude or strength of the control signal. The final metric used is a chattering index [128] formulated as

$$\Lambda_k = \frac{1}{n}\sum_{i=1}^{n} |a_r - \bar{a}_r|, \tag{6.3}$$

where Λ_k is a chattering index over n data points. It is essentially the MAE of the control input a_r and $\bar{a}_r$ which is a low-pass filtered control signal (see Fig. 6.1).

$$\bar{a}_r = \frac{1}{s+\omega_c} a_r, \tag{6.4}$$

where ω_c is the cutoff frequency.

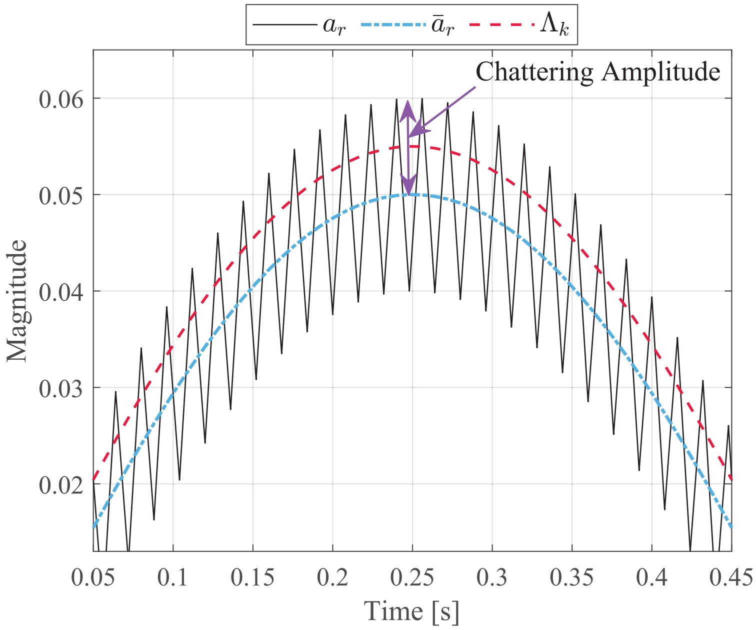


Figure 6.1: Chattering of a_r.

Remark 10. *Equation (6.4) represents a very simple first order and ideal low pass filter which can be set with the 'butter' function in MATLAB. However, this work used MATLAB's 'lowpass' filter function where the the order is determined by the internal design algorithm of the function, which optimizes the filter response according to the provided parameters. A passband of 0.1 Hz was selected and found to provide sufficient results.*

6.2 Robust Trajectory Tracking under Varying Disturbances

In this section, extensive simulation tests are conducted to evaluate the performance of the proposed adaptive observer-based controller in addressing the reference tracking control problem under diverse disturbance scenarios. The testing process includes initial evaluations within the MATLAB/Simulink environment, as outlined in Section 4.3.1, to verify the controllers' ability to stabilize the quadrotor vehicle under specific disturbances within a targeted condition framework. Comparisons are made against classical PID controllers and the manually set algorithm, referred to as STSMC-HOSMO, to validate the effectiveness of the proposed approach.

Thereafter, integrating it into the PX4 control system environment was accomplished within the PX4-powered Clover drone Gazebo simulator, as described in Section 4.3.2. To further evaluate its performance, the controller was tested within the simulation environment, incorporating a custom disturbance plugin.

Nonetheless, it is important to highlight that the trajectory tracking algorithms developed in simulation exhibited the expected behavior when implemented on hardware. This successful hardware implementation is demonstrated in the final video presentation of my CopterHack 2023 project, which can be accessed via the following link: https://www.youtube.com/watch?v=jOovjoOaBpQ&t=5s&ab_channel=SeanSmith.

6.2.1 MATLAB/Simulink Simulations

The focus of the following simulations is analyze controller performance when commanding the position dynamics (3.28)-(3.30) to track a complex trajectory under external aerodynamic perturbations.

The trajectory to track is a circular helix defined in the following

$$\begin{cases} x^d &= a\cos t, \\ y^d &= a\sin t, \\ z^d &= bt. \end{cases} \tag{6.5}$$

The helix has a radius of $a = 8m$, a slope of a/b ($b = 2m$), and t represents the time variation where the desired trajectory (x^d, y^d, z^d) traces a right-handed helix with a pitch of $2\pi b$ about the z-axis.

The Dryden wind gust model from Section 3.3.2 was applied as a time varying external disturbance for $t \in [0, 18]s$ and Gaussian noise was introduced for $t \in [10, 30]s$. The wind model variables can be found in Table 6.1, where the same disturbance was applied to each position dynamic. The tracking results are plotted in Fig. 6.2 with

Table 6.1: Dryden wind gust model parameters for each position dynamic.

Dynamic	n	a_i [m/s^2]	Ω_i [rad/s]	ψ_i [rad]
North	4	[2.3, 3.6, 3, 2.5]	[1, 1.5, 1, 3]	[1, 3, -1, -1]
East	4	[2.3, 3.6, 3, 2.5]	[1, 1.5, 1, 3]	[1, 3, -1, -1]
Down	4	[2.3, 3.6, 3, 2.5]	[1, 1.5, 1, 3]	[1, 3, -1, -1]

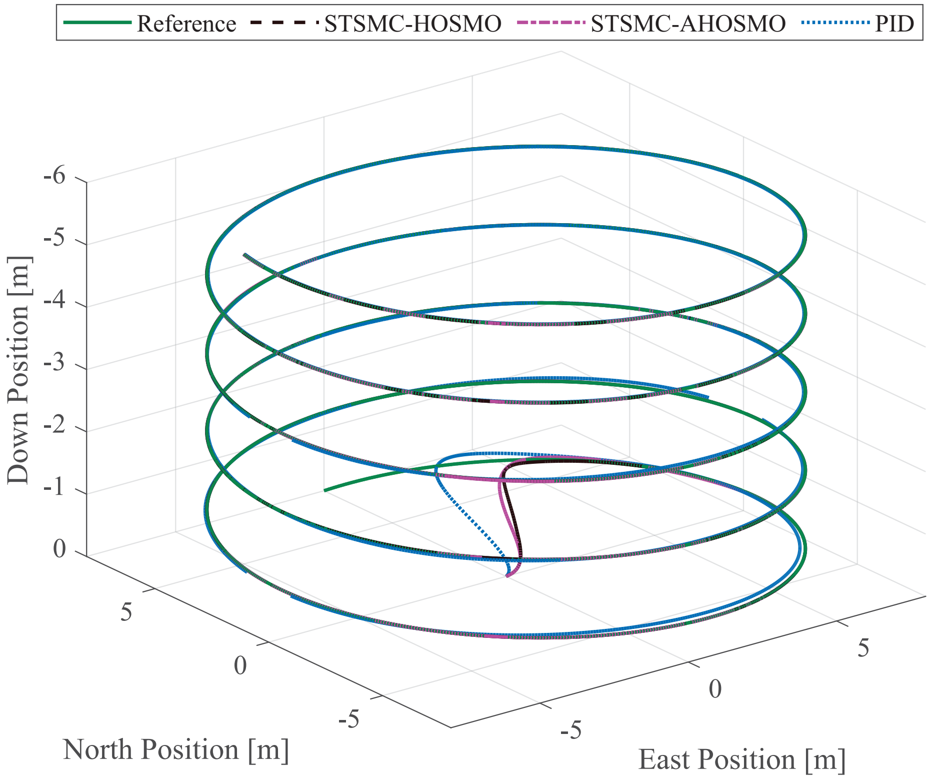
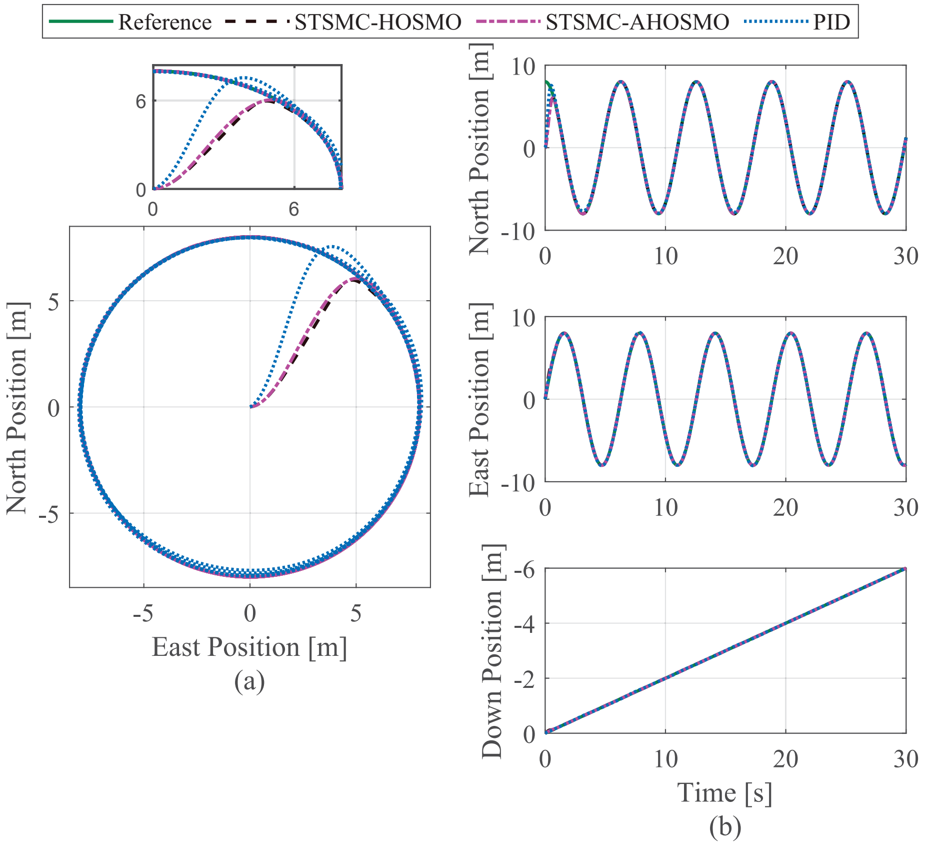


Figure 6.2: 3D circular helix tracking profile with STSMC-AHOSMO, STSMC-HOSMO, and PID control under Dryden wind gust model $t \in [0, 18]s$ and noisy position measurements $t \in [10, 30]s$.

a 2-Dimensional view in Fig. 6.3 and the performance index is quantified in Table 6.2. The overall tracking performance of each control method was reasonably good under the applied disturbance, as depicted in the position tracking plots (see Fig. 6.4(d)). However, the top-down view (Fig. 6.3(a)) clearly shows that the PID control struggles to effectively dampen out the wind gust model. In contrast, the proposed STSMC-AHOSMO demonstrates improved tracking performance by estimating and mitigating the disturbance, as shown in Fig. 6.4(d). This control method achieved a significant reduction in MAE, namely 25% for north tracking, 74% for east tracking, and an impressive 84% for down tracking.

Based on the data presented in Table 6.2, it can be inferred that the tracking performance of both the manually tuned algorithm, STSMC-HOSMO, and the adaptive system algorithm, STSMC-AHOSMO, are comparable when dealing with the influence of wind and noise. However, the primary objective of the adaptive law is to ensure consistent tracking performance across both high and low disturbance



(a)

(b)

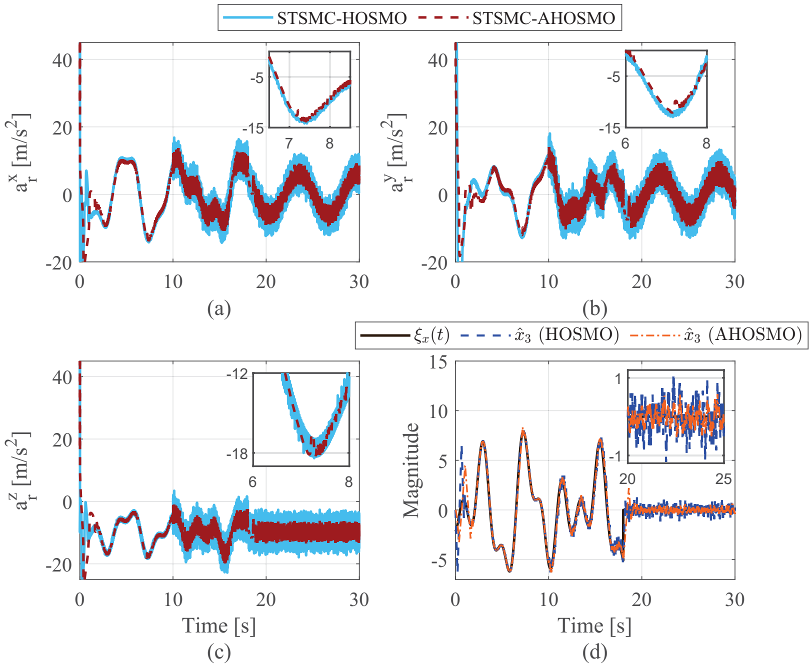
Figure 6.3: 2D simulation response profiles of circular helix: (a) Top-down position tracking of circular helix; (b) Individual dynamic responses.

zones, which has been successfully achieved. Additionally, the adaptive system aims to minimize control effort under bi-directional disturbances, resulting in lower gains, reduced chattering, and decreased sensitivity to measurement noises associated with high gain observers.

From Fig. 6.5(b), it can be observed that the adaptive gains converge to values significantly lower than the manually set gains ($L = 20$). As a result, the disturbance

Table 6.2: Tracking performance for each controller in each reference dynamic.

MAE [cm]	North Position	East Position	Down Position
PID	12.57	4.10	3.55
STSMC-HOSMO	8.93	1.14	0.65
STSMC-AHOSMO	9.36	1.05	0.58



(a)

(b)

(c)

(d)

Figure 6.4: Simulation response profiles from manual and adaptive control algorithms: (a) x-dynamic control input; (b) y-dynamic control input; (c) z-dynamic control input; (d) Disturbance estimation $\tilde{x}_3$ on z-dynamics ξ_z.

estimation component highlighted in Fig. 6.4(d) exhibits reduced chattering amplitudes. This improvement is further complemented by lower state estimation gains, leading to noteworthy reductions in chattering amplitudes in the resulting control efforts, as illustrated in Figs. 6.4(a), (b), and (c), and quantified in Table 6.3.

During the time interval $t \in [0, 10]s$, the presence of wind disturbances causes minor chattering. However, after 10 seconds, the introduction of measurement noise leads to a significant amplification of chattering, as depicted in Fig. 6.4.

The RMS values presented in Table 6.3 provide a measure of the magnitude of the control effort. It is evident that the manually tuned control algorithm, STSMC-HOSMO, consistently applies higher control effort throughout each phase of the simulation. However, despite the increased control effort, the tracking performance does not exhibit significant improvement when compared to the adaptive version, STSMC-AHOSMO (refer to Table 6.2).

This situation creates a scenario where the system operates at high gain levels despite relatively low disturbance levels, resulting in amplified chattering and increased sensitivity to measurement noises, as depicted in Fig. 6.4. This observation reinforces

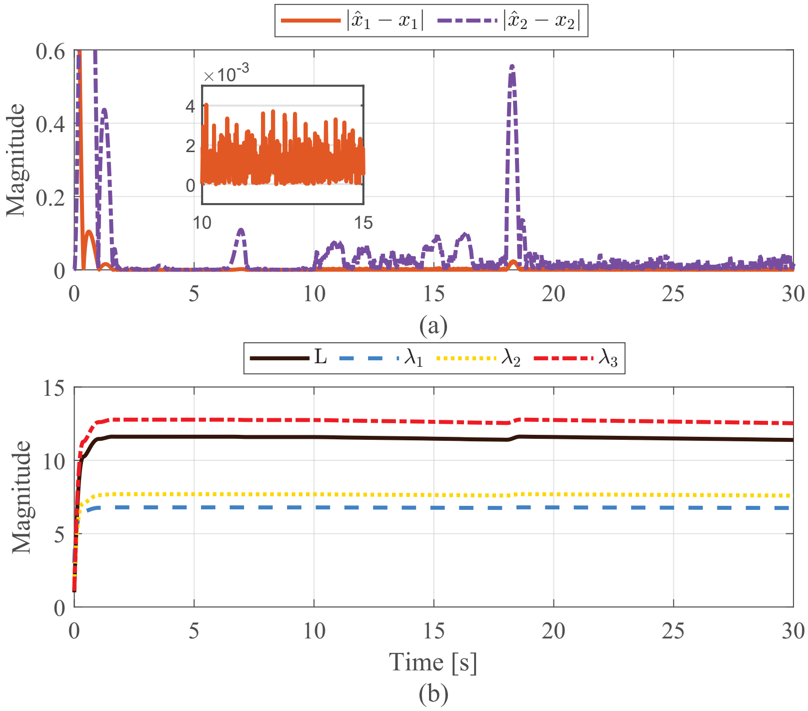


Figure 6.5: Simulation response profiles for STSMC-AHOSMO on the x-dynamics: (a) Position estimation error $\tilde{x}_1$ and velocity estimation error $\tilde{x}_2$ with AHOSMO; (b) Time variation of adaptive gains.

the effectiveness of the adaptive gain design.

It can be observed that despite the acting perturbations (Fig. 6.4(d)), the estimation error (Fig. 6.5(a)) remains in the neighborhood of the origin for almost all time. This interval is defined by Λ_0 and is considered the zero of the system for $t \in [10, 30]s$ in the presence of measurement noise, and the interval is zero for $t \in [0, 10]s$ when the system is absent of any noises. This outcome validates [90, Theorem 5] i.e. an exact robust differentiator.

In general, the differentiation accuracy inevitably deteriorates rapidly with the growth of the differentiation order [67]. This is illustrated with the velocity estimation $|\hat{x}_2 - x_2|$ in Fig. 6.5(a). Despite efforts to mitigate the impact of noisy signals through treatment and filtering, the performance improvement is limited and relies mainly on reducing the sampling step. However, it is worth noting that this observer exhibits a high level of robustness against parametric uncertainty when compared to an EKF. Moreover, when combined with a low noise signal feedback that undergoes filtering within PX4 before being utilized in the high-rate firmware, this observer demonstrates excellent performance in the given environment. The subsequent sections, including Gazebo simulations and hardware application, further illustrate and support these positive outcomes.

Lastly, although the proposed adaptive law successfully reduces the chattery signal (as depicted in Fig. 6.4), it is important to note that the chattering is not completely eliminated. In these simulations, the presence of the *sign* function persists, in contrast to the smoothing function described in Section 5.4.5. In MATLAB simulations, when the estimation error (Fig. 6.5) converges to the neighborhood $|\tilde{x}_1| < \Lambda_0$, it does so rapidly due to the simplicity of the mathematical dynamics and the effectiveness of the discontinuous *sign* function in conjunction with the proposed algorithm. Consequently, the adaptive gains exhibit a slower dynamic reduction, contingent upon the estimation error $|\tilde{x}_1|$.

However, it should be recognized that in real-world scenarios or more complex systems with higher uncertainty, the adaptive gains would dynamically reduce at a more appropriate rate with a well set K_a gain. Therefore, in the subsequent sections where the STSMC-AHOSMO is applied to the Gazebo simulator and COEX Clover hardware in the PX4 environment, the smoothing function is utilized, and the dynamic reduction of the adaptation law becomes more prominent.

Table 6.3: Root mean square and chattering index of control input for a: $t \in [0, 10]s$, b: $t \in [10, 18]s$, c: $t \in [18, 30]s$ for [a, b, c] to evaluate control performance.

RMS [m/s²]	a_r^x	a_r^y	a_r^z
STSMC-HOSMO	[26.18, 8.11, 6.27]	[17.81, 6.58, 6.59]	[8.14, 5.32, 4.31]
STSMC-AHOSMO	[25.5, 6.79, 4.71]	[7.36, 5.06, 4.95]	[8.84, 3.87, 2.20]
$\boldsymbol{\Lambda}_k$ **[m/s²]**	a_r^x	a_r^y	a_r^z
STSMC-HOSMO	[1.59, 3.17, 3.28]	[0.68, 3.54, 3.64]	[1.06, 3.75, 3.87]
STSMC-AHOSMO	[1.53, 1.64, 1.90]	[0.38, 1.84, 2.13]	[0.61, 1.66, 1.89]

6.2.2 Clover Gazebo Simulations

For the complex trajectory tracking simulation in this environment, the COEX Clover was targeted to track a lemniscate of Bernoulli. This trajectory can be represented by the following parametric equations

$$\begin{cases} x &= \frac{rcos(a)}{1+sin^2(a)} \\ y &= \frac{rsin(a)cos(a)}{1+sin^2(a)}, \end{cases} \tag{6.6}$$

where a is the parametric variable or the angle to sweep through for this tracking and $r = 3m$ is the radius of the lemniscate. While (6.6) provides the position of the trajectory, feedforward components such as velocity are needed for the STSMC and will also improve trajectory tracking with the PID controllers. With (6.6) being a smooth trajectory, the velocity and acceleration are available everywhere, with the derivative defined as

$$\frac{dx}{dt} = \frac{dx}{da} \cdot \frac{\Delta a}{\Delta t}, \quad \frac{dy}{dt} = \frac{dy}{da} \cdot \frac{\Delta a}{\Delta t},$$

where the product rule of a linearly increasing angle $\frac{\Delta a}{\Delta t}$ (constant angular rate) is used to get the velocity from the change in position with angle, this value is set within the high level command module. The final velocity form is given

$$\begin{cases} \frac{dx}{dt} &= -\frac{2rsin(a)(cos(2a)+5)}{(cos(2a)-3)^2} \frac{\Delta a}{\Delta t} \\ \frac{dy}{dt} &= \frac{2r(3cos(2a)-1)}{(cos(2a)-3)^2} \frac{\Delta a}{\Delta t}. \end{cases} \tag{6.7}$$

Similarly, the acceleration feedforward is determined with the following

$$\begin{cases} \frac{d^2x}{dt^2} &= \frac{rcos(a)(44cos(2a)+cos(4a)-21)}{(cos(2a)-3)^3} (\frac{\Delta a}{\Delta t})^2 \\ \frac{d^2y}{dt^2} &= \frac{4rsin(2a)(3cos(2a)+7)}{(cos(2a)-3)^3} (\frac{\Delta a}{\Delta t})^2. \end{cases} \tag{6.8}$$

Remark 11. *Any smooth analog path can be discretized at a high enough update rate to have PX4 fly what looks to be a completely smooth path. Setpoints must be given to PX4 at a rate of at least 2Hz or the Clover will go into fail-safe mode, this sets the motors into safety mode where the Clover will most likely glide into a fall. Preferably sending setpoints at 30 Hz or greater is desired for smooth trajectories. Clovers simple offboard sends setpoints at 30 Hz and publishes using MAVROS.*

Similar to the MATLAB/Simulink simulations, a variable force disturbance in the form of a Dryden wind gust model, or a sum of *sin* functions, is applied to the center of mass of the Clover in the inertial frame of reference during the complex portion of the trajectory tracking. The custom plugin was used as discussed in Section 4.3.2. The model parameters are listed in Table 6.4. The gains for the HOSMO were set

Table 6.4: Dryden wind gust model parameters for each position dynamic in Gazebo.

Dynamic	n	a_i [m/s^2]	Ω_i [rad/s]	ψ_i [rad]
North	1	[0.8]	$[\frac{2\pi}{9}]$	[0]
East	1	[1.1]	$[\frac{2\pi}{7}]$	[0]
Down	1	[1.1]	$[\frac{2\pi}{8}]$	[0]

conservatively in the presence of time varying disturbances and can be found in Table B.5.

The translational tracking is plotted in Fig. 6.6. It is clear the PID struggles to damp out and reject the varying disturbances specifically in the horizontal plane. The robust control algorithm, particularly the STSMC-AHOSMO reduces the MAE by 44.4% and 74% in the North and East directions respectively (see Table. 6.5). A main reason is the AHOSMO is able to estimate the disturbance influence online on each dynamic presented in Figs. 6.7(b), (d), and (f). This estimated disturbance

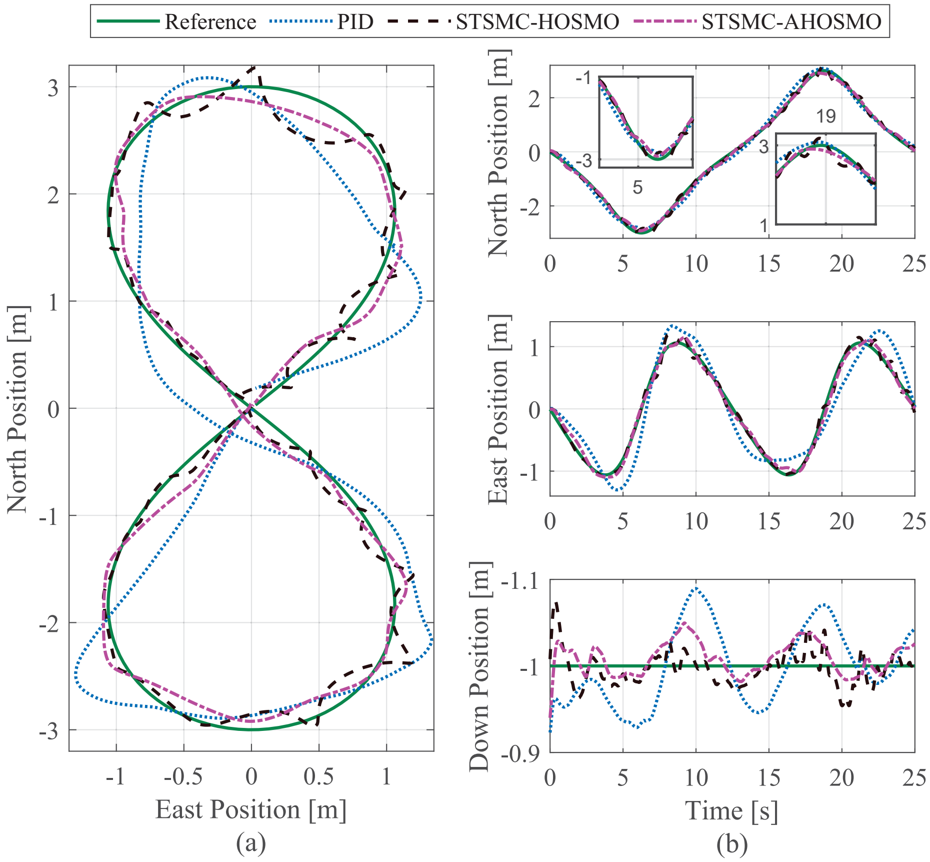


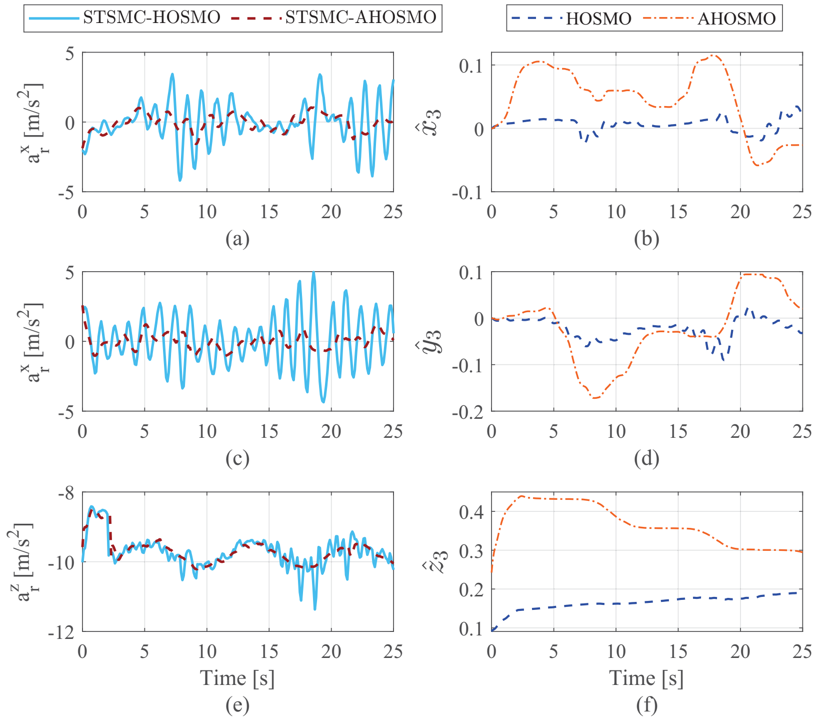
Figure 6.6: Clover Gazebo tracking profiles: (a) Top down view of lemniscate tracking; (b) Individual dynamic trajectory tracking.

Table 6.5: Tracking performance for each controller in each reference dynamic for Gazebo simulation.

MAE [cm]	North Position	East Position	Down Position
PID	11.9	25.1	4.15
STSMC-HOSMO	6.89	6.41	1.86
STSMC-AHOSMO	6.61	6.52	1.93

influence is incorporated into the control input to effectively counteract these forces acting on the quadrotor. The persistent steady-state error observed with the PID control arises from integral overload, wherein the gains are tuned for undisturbed flight scenarios.

From Table 6.5, it is clear that the manually tuned algorithm STSMC-HOSMO and adaptive STSMC-AHOSMO provide identical tracking and performance on each dynamic during the trajectory tracking under the varying force disturbances. Although, similar to the MATLAB/Simulink simulations, the adaptive law empowers the STSMC-AHOSMO to achieve this efficiency without relying on conservatively set



(a) (b)

(c) (d)

(e) (f)

Figure 6.7: Clover Gazebo control characteristic profiles: (a,c,e) Individual dynamic control input; (b,d,f) Disturbance estimation on each individual position dynamic.

gains (see Table B.5). This efficiency is quantified in Table 6.6, where according to the RMS the adaptive algorithm requires a lower magnitude of control input with a reduction of 53.5%, 70.6%, and 15.2% on the x, y, and z dynamics respectively. Likewise, there is a significant improvement on the chattering amplitude of the control input (see Fig. 6.7) which is confirmed with the reduction in chattering index seen in Table 6.6.

The simulation exhibits low noise levels, as the sensor plugins incorporate noise models with a standard deviation of $\sigma = 0.001m$. The estimation error (Fig. 6.8) converges on an interval on the order of 10^{-3} for the altitude dynamics remaining in the vicinity of zero defined by Λ_0. The low estimation error is a result of the growing adaptive gains in Fig. 6.8(d). The L_z gain grows to compensates disturbances before dynamically reducing to minimize the control effort for the system. Gains L_x and L_y converge on smaller values with constant transitions between reductions to minimize

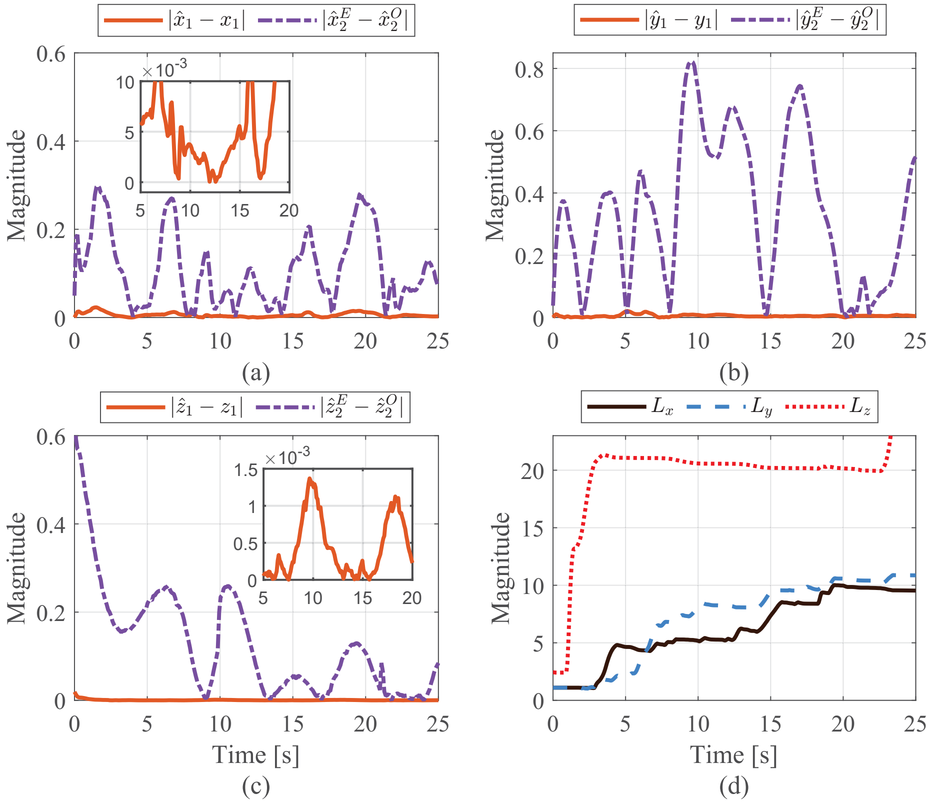


Figure 6.8: Clover Gazebo response profiles: (a,b,c) Estimation error for each AHOSMO on position dynamics; (d) Time variation of adaptive gains for each AHOSMO.

control effort and jumps to ensure the gain conditions (5.22) are satisfied under the influence of perturbations (Table 6.4).

The velocity estimation difference between the onboard EKF and AHOSMO (see Fig. 6.8) varies due to the measurement noise and mainly the perturbations applied to the system. It has been shown in [129] that the HOSMO outperforms the EKF with state estimation in the presence of disturbances and in ideal conditions, and while the HOSMO is more sensitive to noise, it produces a more reliable behaviour even in the presence of noises and disturbances. The lower order SMO was also shown to outperform the EKF under these conditions however it does not provide a reliable perturbation estimation like the HOSMO. Therefore, it benefits the the control system to use velocity estimations from the AHOSMO in these conditions, which may also improve the velocity PID control performance if they were paired.

Table 6.6: Root mean square and chattering index of control input to evaluate control performance.

RMS[m/s^2]	a_r^x	a_r^y	a_r^z
STSMC-HOSMO	1.44	1.77	0.46
STSMC-AHOSMO	0.67	0.52	0.39
$\boldsymbol{\Lambda}_k$[m/s^2]	a_r^x	a_r^y	a_r^z
STSMC-HOSMO	0.91	1.38	0.16
STSMC-AHOSMO	0.18	0.22	0.044

6.3 Disturbance Handling for Low Altitude UAV Pick-and-Place Tasks

Similar to the trajectory tracking section, this section involves thorough simulation testing to assess the performance of the proposed observer-based controller in the presence of diverse disturbances. However, the specific control problem addressed here differs, as the focus is on altitude stabilization during real-world UAV pick-and-place missions under various disturbance scenarios.

To begin, initial tests are conducted in MATLAB/Simulink to verify the controller's capability to handle altitude tracking in the face of load variations and ground effects, particularly during the low altitude load transportation. The performance of the proposed controller is compared to that of PID control, which is commonly employed in industry and open-source control systems.

Prior to conducting the experimental tests, pre-experimental evaluations were carried out in the PX4-powered Clover drone Gazebo simulator to verify the functionality and compatibility of the control algorithms within the PX4 environment. To emulate object grasping, a constant force plugin was employed in this simulation environment.

The final phase of the evaluation involved experimental testing, serving as the ultimate assessment of the thesis work in real-time conditions. This phase encompassed a multi-phase autonomous pick-and-place mission that involved low altitude grasping while considering ground effects and varying loads. A video showcasing the experiment can be accessed at *https://youtu.be/QxSlLZeTRd8*.

6.3.1 MATLAB/Simulink Simulation

For this simulation, the altitude dynamics (3.30) was commanded to track varying altitudes under the discussed disturbances to verify the robustness of the proposed control method with the tracking results plotted in Fig. 6.9.

It is clear the PID controllers struggle to damp and eliminate the time varying

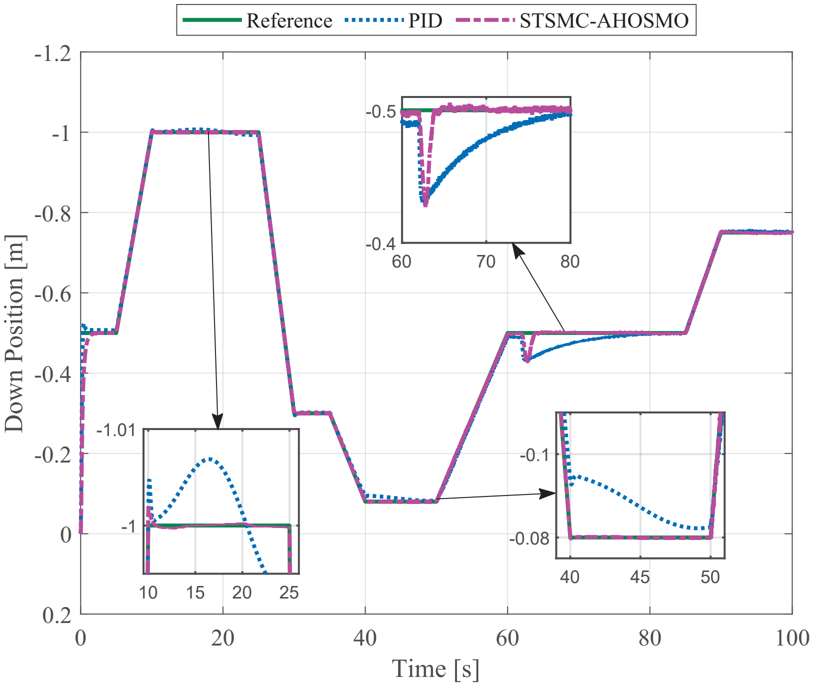


Figure 6.9: Simulation results on the altitude dynamic tracking for PID and STSMC-AHOSMO control under Dryden wind gust model $t \in [0, 100]s$, significant ground-effect influence $t \in [40, 50]s$, noisy altitude measurements $t \in [50, 100]s$, and an additional mass of 150 g introduced at $t = 60s$.

perturbations from $t \in [10, 25]s$ with gains tuned for an undisturbed system. The AHOSMO is able to estimate them and feed this into the STSMC for disturbance rejection allowing for a more robust control method and reducing MAE by 76% (0.47 cm ⇒ 0.11 cm) for $t \in [0, 25]s$. This estimation is illustrated in Fig. 6.10(a).

During $t \in [40, 50]s$, the ground effect becomes significant at a low altitude of 0.08 m equating to a ratio of $z_r/r = 1.27$. The STSMC-AHOSMO compensates for the additional unmodelled force on the quadcopter (upward shift in Fig. 6.10(a)) while maintaining finite time error convergence (Fig. 6.10(c)) helping the quadcopter reach the desired minimum height. A MAE reduction of 94% is established on the interval $t \in [25, 50]s$.

When an additional mass of 150 g was introduced to the system at $t = 60s$ to replicate in-flight object grasping, it causes an initial tracking error (Fig. 6.10(b)), however the AHOSMO is able to quickly adjust (shift down in Fig. 6.10(a)) to compensate this. The MAE was reduced by 65% (1.14 cm ⇒ 0.4 cm) for $t \in [50, 100]s$. Generally, if parametric uncertainties exist in a vehicle system in terms of a total mass, the uncertainty is compensated by introducing an integral (I) controller in the PID

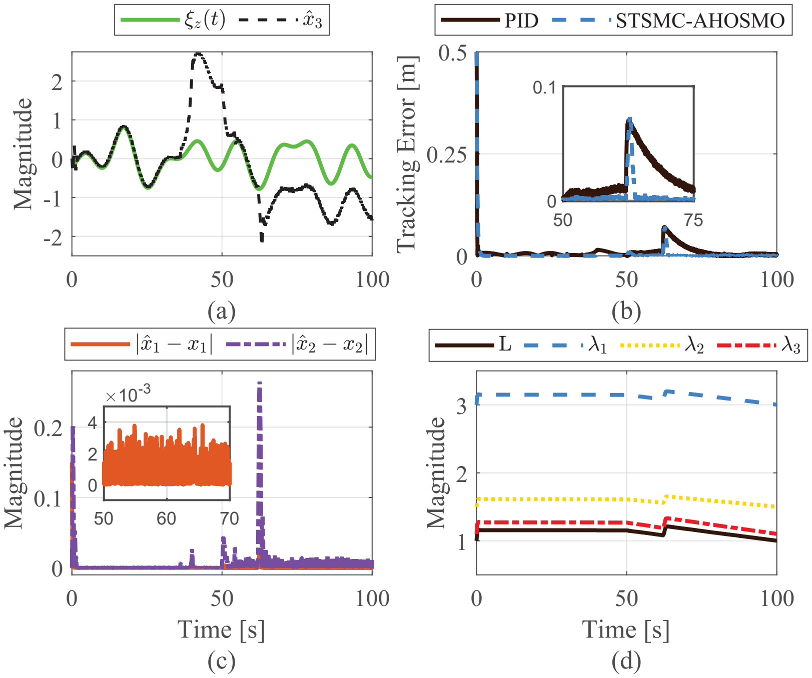


Figure 6.10: Simulation response profiles: (a) Disturbance estimation $\tilde{x}_3$ from AHOSMO; (b) Altitude tracking error; (c) Position estimation error $\tilde{x}_1$ and velocity estimation error $\tilde{x}_2$ with AHOSMO; (d) Time variation of adaptive gains.

controller [39]. Overloading this term may lead to instability (steady state error or complete loss of control) highlighting the limitations of PID control.

It can be observed that despite the acting perturbations (Fig. 6.10(a)) the estimation error (Fig. 6.10(c)) remains in the neighborhood of the origin for almost all time. This neighborhood is considered as the zero of the system and is defined by the threshold Λ_0. For $t \in [0, 50]s$ the system is absent of measurement noise therefore the error converges to zero in a finite time.

The adaptive parameters shown in Fig. 6.10(d) increase until the estimation error reaches the origin in the absence of noise leading to constant gains. At $t = 50s$ they dynamically reduce where $|\tilde{x}_1| < \Lambda_0$ in the presence of Gaussian measurement noise. This adjustment minimizes the control effort and optimizes control performance in the low disturbance zone. The standard deviation was set to $\sigma = 0.001m$ therefore most estimation error is within 3σ providing a good estimate for Λ_0. At $t = 60s$, the sliding mode is lost because of the growth of the disturbance amplitude from the additional payload (see Fig. 6.10(a)). Likewise, the gains (Fig. 6.10(d)) grow until the observation error reaches $|\tilde{x}_1| < \Lambda_0$ where the fine tuning phase begins again.

While low gains were sufficient (see Fig. 6.10(d)) in this simulation, practical systems have an increased level of uncertainty and disturbance levels and this is illustrated in the Gazebo and experimental results where the adaptive parameters converge on larger values.

6.3.2 COEX Clover Gazebo Simulation

The COEX Clover Gazebo simulator was used to test the proposed control system in the PX4 environment. A 'simple offboard' trajectory generation module was used to provide position coupled with feedforward velocity, and acceleration setpoints through MAVROS. The altitude was determined by the Clover laser range-finder plugin with other localization from ArUco marker vision-based navigation and data was fused with the PX4 EKF, including sensor noise ($\sigma = 0.001m$) and bias. The altitude tracking results can be seen in Fig. 6.11.

The STSMC-AHOSMO provides a faster convergence rate in the transient stages of the Clovers tracking where it reacts on the delayed takeoff quicker from the motor

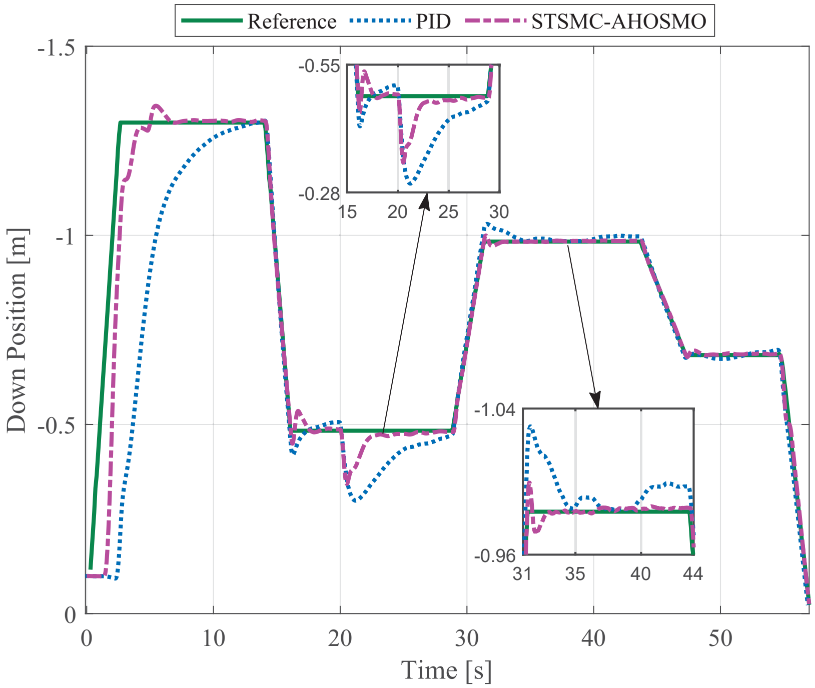


Figure 6.11: Clover Gazebo simulation: altitude tracking for PID and STSMC-AHOSMO with a force of 4.3N introduced at $t = 19s$.

arming, this reduced MAE 65% (32.1 cm ⇒ 11.1 cm) from $t \in [0, 14]s$. This is not always ideal, as it will cause a jerk in the takeoff depending on the commanded velocity, therefore a lower velocity should be set in the takeoff for this reason. The robustness of the control method is tested by introducing a downward force of 4.3N at $t = 19s$ on the Clover to replicate grasping an object mid-flight. The STSMC-AHOSMO is able to quickly adjust and stabilize with help from the observer illustrated in Fig. 6.12(a), which reduced the MAE significantly by 59% (6.52 cm ⇒ 2.62 cm) for $t \in [14, 29]s$. An increased steady-state error with PID is evident from $t \in [32, 44]s$ suggesting integral instability from the additional load where STSMC-AHOSMO reduces MAE by 37% (1.24 cm ⇒ 0.78 cm) for $t \in [28, 57]s$.

In Fig. 6.12(c), the normalized hover thrust determined from the PX4 zero order EKF module is shown. The sudden increase at $t = 19s$ is from the additional payload. When this value changes, the acceleration adjustment is added to the integral portion of the velocity PID controller to absorb the effects (see Fig 6.12 (d)). This hover thrust variance is better handled by the STSMC-AHOSMO when loads become significant.

With a low noise level, the estimation error converges to values on the order of

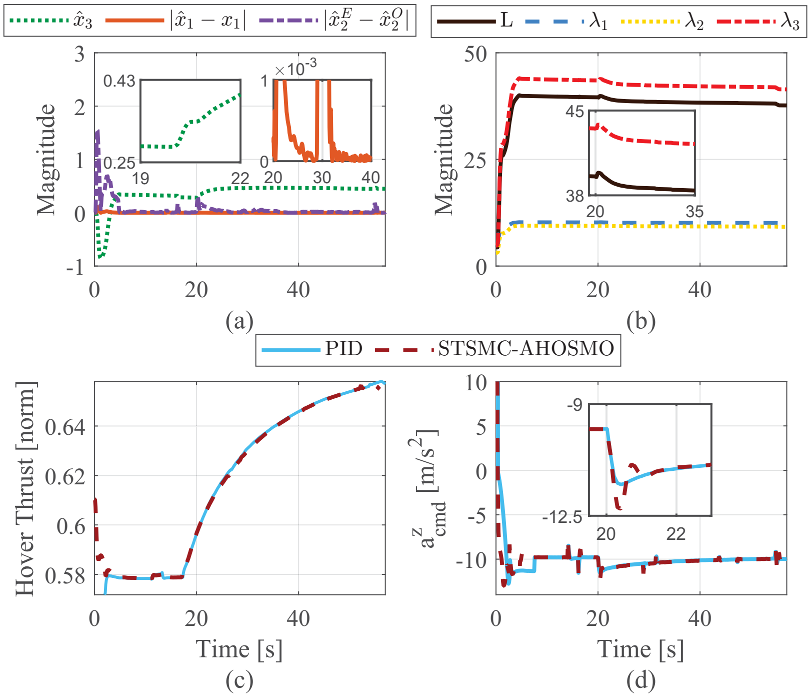


Figure 6.12: Clover Gazebo simulation response profiled: (a) Disturbance estimation $\hat{x}_3$, position estimation error $\tilde{x}_1$, and velocity variation between the EKF and AHOSMO $|\hat{x}_2^E - \hat{x}_2^O|$; (b) Time variation of adaptive gains; (c) Hover thrust estimation; (d) Control input.

10^{-5} with jumps around 10^{-3} remaining in the vicinity of zero defined by Λ_0. This is a result of growing adaptive gains until the conditions in (5.22) are satisfied. At $t = 19s$ the gains grow to compensate the additional load (see Fig. 6.12(b)) where $|\tilde{x}_1| < \Lambda_0$ is reached, and then are fine tuned again to minimize the control effort.

6.3.3 Practical Application of STSMC-AHOSMO for Altitude Control

This section presents experimental results through a multi-phase experiment used to replicate the simulation tests with comparable characteristics, highlighting the effectiveness of the proposed control algorithms. The attitude controllers were tuned using the adaptive auto-tune control module provided by PX4, assuming sufficient tracking of the quaternion setpoints provided.

The multi-phase experiment and laboratory environment can be seen in Fig. 6.13. Four Flex 13 OptiTrack cameras are used to cover the fight area with an ArUco marker labelling the start position and drop zone. The cameras are connected to a computer over USB that runs OptiTrack Motive software. A custom UDP connection is established between the NatNet IP multicasting server in Motive and Raspberry Pi to stream pose feedback and remap it to the PX4 using MAVROS. Each computer in the setup interacts using a time synchronized local network.

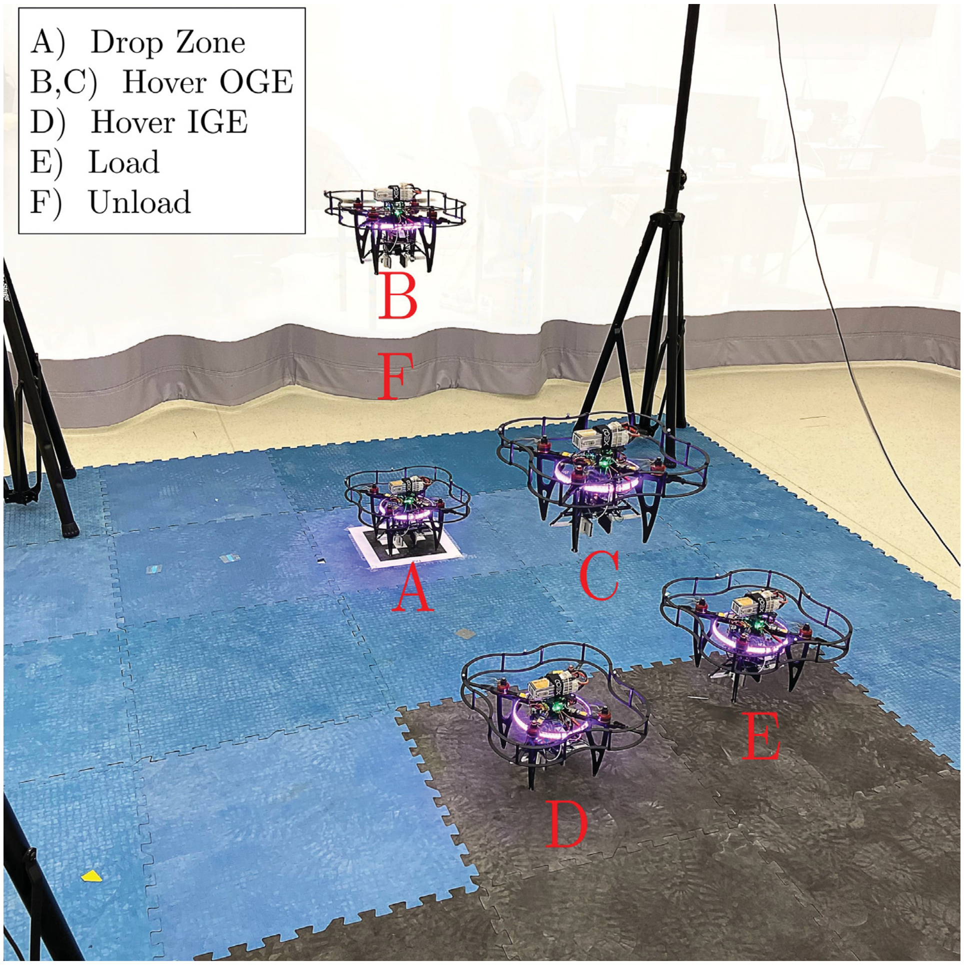


Figure 6.13: Multi-phase Clover experiment. Phase1: [A,B,C][0,27]s, Phase2: [C,D][27,43]s, Phase3: [D,E][43,60]s, and Phase4: [E,C,F,A][60,80]s.

The altitude tracking results for the multi-phase experiment can be seen in Fig. 6.14 with performance quantified in Table 6.7. *Phase 1* focuses on the takeoff and tracking under minimal disturbances, *Phase 2* focuses on the flight stability IGE, *Phase 3* focuses on the flight stability during the autonomous grasping, and *Phase 4* shows the tracking when returning to the drop zone under an increased payload.

Table 6.7: Tracking performance for each controller in each phase.

MAE [cm]	Phase 1	Phase 2	Phase 3	Phase 4
PID	2.75	1.8	3.2	3.0
STSMC-AHOSMO	1.0	0.51	0.80	0.84

Phase 1

The Clover takes off to an altitude of 1.1m above the drop zone and tracks to an altitude of 0.8m above the loading station. The STSMC-AHOSMO provides a tighter tracking and reacts to the delayed takeoff from motor arming compared to PID reducing MAE by 64% (see Table 6.7). It can be observed that the estimation error (Fig. 6.15(a)) converges to a proximity of zero defined by Λ_0 as a result of adapting gains (Fig. 6.15(b)) in the presence of perturbations. When $|\tilde{x}_1| < \Lambda_0$, the gains dynamically reduce to minimize the control effort (Fig. 6.15(d)) in OGE regions. It should be noted the AHOSMO and EKF converge on a similar velocity estimation with $|\hat{x}_2^E - \hat{x}_2^O|$ converging towards zero (see Fig. 6.15(a)). However, an advantage of the AHOSMO over an EKF is that it can be made considerably more robust to parametric uncertainties [104].

Phase 2

The Clover descends to a marker altitude of 0.24 m. The ground effect becomes significant corresponding to $z_r/r = 3$ where the required hover thrust drops at $t = 30s$ (see Fig. 6.15(c)) leading to an increased adaption rate where $|\tilde{x}_1| \approx \Lambda_0$ (see Fig. 6.15(b)). The estimation error remains in the interval of zero defined by Λ_0 and the STSMC-AHOSMO provides a 71.7% MAE reduction (see Table 6.7) compared to that of PID by compensating the additional force.

Phase 3

The Clover ascends to an altitude of 0.35 m for the grasping task which reduces ground effect and increases the thrust to hover slightly (see Fig. 6.15(c)). A 100 g box is handed to the gripper to grasp at $t = 52s$. Upon grasping, the added load drops the Clover about 8 cm with PID control and 4 cm with STSMC-AHOSMO showing

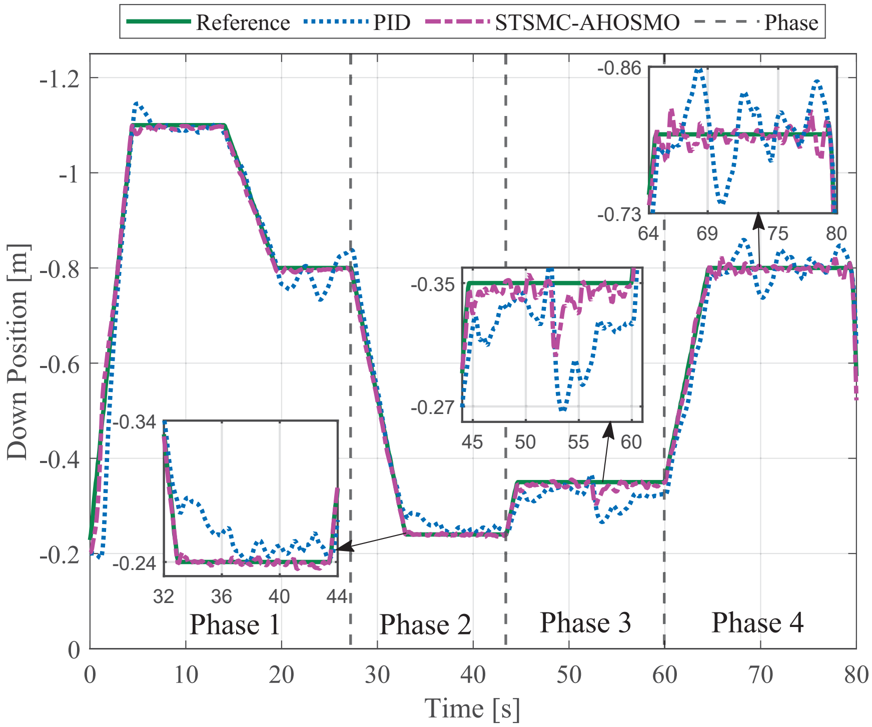
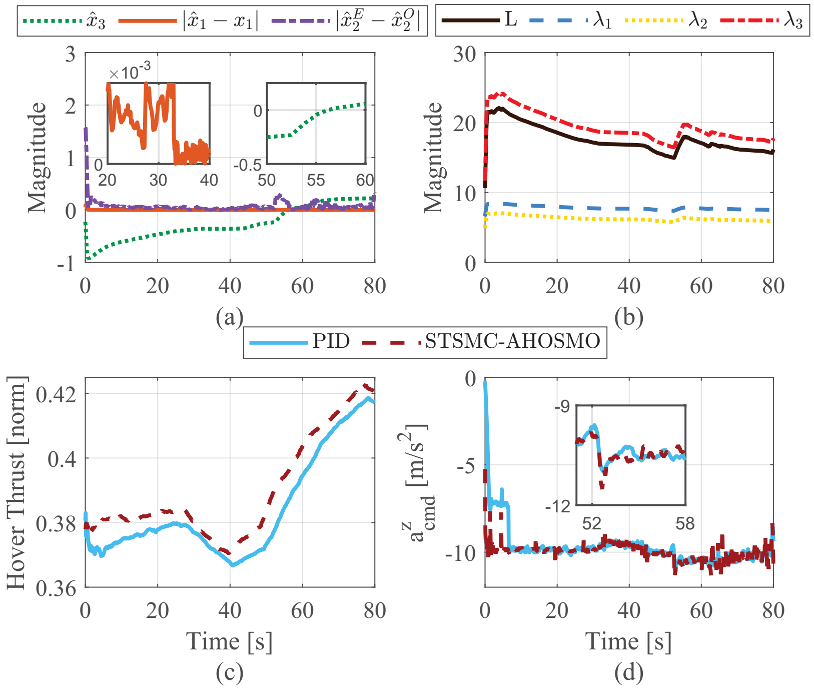


Figure 6.14: Multi-phase Clover experiment altitude tracking with a mass introduced at $t = 52s$.

improved robustness. The STSMC-AHOSMO provides a 75% MAE reduction (see Table 6.7) compared to PID by compensating the load. The jump in hover thrust is from the additional load where the sliding mode is lost at this time resulting in a gain increase (Fig. 6.15(b)) to satisfy the conditions in (5.22) and bring estimation error within the bound defined by Λ_0.

Phase 4

The Clover ascends to 0.8 m and is tasked to return the target object back to the drop zone. The STSMC-AHOSMO improves altitude tracking by reducing the MAE by 72% (see Table 6.7) compared to the errors by PID method. Despite the additional load and perturbations, the estimation error is kept within Λ_0 and the gains dynamically reduce (Fig. 6.15(b)) where $|\tilde{x}_1| < \Lambda_0$ to minimize control effort. The hover thrust increase is combined with the integral (I) term (notice shift down at $t = 52s$ in Fig. 6.15(d)), reducing steady state stability and hover thrust variance is better handled by STSMC-AHOSMO. The control signal amplitude increases (Fig. 6.15(d))



(a) (b) (c) (d)

Figure 6.15: Experimental response profiles: (a) Disturbance estimation $\hat{x}_3$, position estimation error $\tilde{x}_1$, and velocity variation between the EKF and AHOSMO $|\hat{x}_2^E - \hat{x}_2^O|$; (b) Time variation of adaptive gains; (c) Hover thrust estimation; (d) Control input.

with the magnitude of perturbation to maintain a low estimation and tracking error.

All of the characteristics from the MATLAB and Gazebo simulations were replicated in this multi-phase experiment verifying the effectiveness of the control methods.

6.3.4 Limitations and Discussion

From the observation, limitations with this development include the initial assumption that the grasped target load is a point mass having negligible effects on the mass moment of inertia. Also, it was assumed that ground effect produces a uniform force on the quadrotor by effecting the total thrust, as opposed to considering IGE on each rotor [50] where additional disturbance torques are produced when the multirotor is close to the ground [45]. During the experiment, both controllers managed to keep the Clover in the air however, many test runs involved the Clover struggling to maintain its attitude leaving it swaying back and forth within the horizontal plane of the motion capture volume. This behavior was particularly noticeable when the gripper grasped the target object closer to the front or the back rather than the middle, thereby influencing the systems mass moment of inertia.

It should be noted that all baseline PID controllers were tuned for flight in an undisturbed environment including using the adaptive auto-tune module to tune the attitude controllers. Therefore, even with baseline PID attitude controllers, the proposed robust altitude algorithm is able to significantly improve altitude performance, although the attitude could still be compromised. Compensation methods such as improved controllers based on estimated inertial parameters [130], compensating varying inertial parameters [41], or applying the adaptive robust controller from this work on the pitch and roll dynamics offer solutions to these limitations.

6.4 Summary

This chapter is dedicated to the testing and verification of the designed STSMC-AHOSMO in two practical control problems. These problems are:

1. Complex trajectory tracking under time varying disturbances in simulations.
2. Low altitude indoor aerial pick-and-place missions under varying loads and ground effect in real-time implementation.

Both of these cases involve the challenge of tracking trajectory setpoints in the presence of real-world disturbances. In problem (1), the STSMC-AHOSMO was simulated using MATLAB/Simulink, and it outperformed PID control in terms of trajectory tracking. The adaptive gain of the controller reduced control effort and chattering compared to the manually tuned STSMC-HOSMO. The tests were then replicated using the Clover Gazebo simulator and PX4 environment, yielding comparable results to the MATLAB/Simulink simulations.

Moving on to problem (2), the STSMC-AHOSMO exhibited significant improvement over PID control when dealing with varying system loads and ground effect. PID control had limited tolerance for large constant disturbances or abrupt changes, whereas the STSMC-AHOSMO allowed the Clover drone to adapt quickly under such conditions. These characteristics were observed in both simulation environments.

The firmware used to run the SITL simulator was flashed to the COEX Clover drone for a multi-phase practical mission involving low altitude flight and target object retrieval. The performance observed in simulation was successfully replicated during the experimental phase. However, it was noted that the Clover drone was unable to maintain a low altitude, unlike the STSMC-AHOSMO, which was able to operate effectively in the presence of ground effect perturbations. The sudden change in system mass resulting from object grasping caused destabilization for both control methods, but the STSMC-AHOSMO demonstrated better handling of this situation. Although altitude was maintained using the developed method, attitude tracking was compromised under ground effect and with heavier payloads. This outcome highlights the need for robust and adaptive attitude control solutions in future work, as the initial assumption of a point mass was deemed unreasonable for heavier and larger objects.

Chapter 7

Conclusions

The chapter summarizes the work in this thesis, the main contributions, and areas that can be extended to further research and development in the field of robotics.

7.1 Summary of Contributions

In this thesis, a novel position control module for a quadcopter under various real world disturbances is proposed. These disturbances are complex in nature and have a significant influence on the vehicles ability to follow trajectories while performing mission and safety critical tasks.

A mathematical model of a quadrotor was presented and a ground effect model was empirically determined for the COEX Clover to be tested in the MATLAB/Simulink simulation environment for controller design and verification. Likewise, a hardware system was developed including a rigid gripper mechanism for indoor aerial pick-and-place tasks to test varying payload disturbances. This setup was integrated into a motion capture system while high-level communication and trajectory generation modules allowed for fully autonomous mission testing under a targeted condition framework. A PX4 powered Gazebo simulator was utilized to verify the proposed algorithms before experimental testing.

The control strategy was presented to estimate and compensate the unmodelled dynamics acting on the system to ensure stable flight. Despite not incorporating any disturbance models, the method proved to handle the bounded unknown disturbances with significant robustness. A novel adaptive law was developed allowing the observer to estimate and compensate bi-directional disturbances while minimizing the control effort and noise sensitivity that is common in high gain observers. This method is unique in its application by not only handling ground effect but also varying loads in aerial pick-and-place tasks. Detailed Lyapunov stability analysis is used to prove the finite time convergence of the estimation error and asymptotic tracking of the desired trajectories with the robust control method. The integration of this control method into the PX4 control structure is highlighted along with gain selection, chattering attenuation, and gain auto-tuning of the existing PID controllers.

Extensive simulations and experiments were conducted to verify the effectiveness of the proposed control solution, including a complex trajectory tracking task and a multi-phase low altitude pick-and-place mission. The proposed method exhibited significant tracking performance improvements upon classical PID controllers common in standard multirotor control systems. It is accomplished through disturbance estimation and rejection while being robust against parametric uncertainties and system perturbations. The adaptive law improved the performance of the control system by minimizing the gains and lowering the effects of measurement noises and the chattering phenomena.

7.2 Future Work

This section provides possible improvements and extension in the future work.

7.2.1 Autonomous Flight

The experimental outcomes presented in this thesis are limited to flight within an indoor motion capture system volume. Initially, the COEX Clover platform utilized computer vision-based localization employing ArUco markers. However, the field of view of the downward-facing camera was obstructed by the gripper mechanism, necessitating its replacement. Many real world scenarios do not have a precise motion capture system to localize both the drone and target object, therefore replacement of this system setup is an important area of research and development.

Flight experiments outside of the motion capture system require an alternative means of state estimation, particularly one that runs completely onboard the drone. Re-configuring the Clover and using vision based methods is a possibility, other considerations are non-vision based methods including particle filter-based indoor navigation or advanced ultrasonic acoustic signal methods. Another issue with removing the motion capture system is locating the target object, where in this case an onboard vision based algorithm could be developed to precisely locate the target object relative to the gripper mechanism. With the being said, matching the precision provided by the motion capture system is not likely, and adjusting the gripper design to handle the position uncertainty is necessary.

7.2.2 Gripper Design

The success of autonomous grasping using a rigid gripper mechanism is highly dependent on the stability of the UAV, given its limited DOF. Consequently, it is crucial to prioritize compliance and account for limited positional accuracy in the design. In this thesis, the 4-bar rigid gripper mechanism is limited to stationary grasping under ideal conditions. Future work should focus on design improvements with the following considerations:

- Integrating soft robotics into the design to allow for flexible grasping of different geometries while minimizing contact forces with low and highspeed grasping.
- Consider passive mechanical compliance through springs and force absorption over robot compliance using sensors and actuators.
- Ensure a large/adjustable work envelope that is robust to positional errors of the UAV.

The above-mentioned characteristics will alleviate the pressure put on the designed control system.

7.2.3 System Modelling

While a ground effect model was identified for the COEX Clover in Section 3.3.2 there is still an extensive amount of area for contribution with the Clover. A listed of possibilities are provided:

- Identify the physical characteristics of the Clover, such as component dimensions, system mass, and the system mass moment of inertia through experiment.
- Identify a motor model through experiment.
- Integrate more sophisticated aerodynamic modelling with relation to the Clovers velocity relative to the forces and moments acting on the vehicle.

- Use the identified modelling characteristics to replace the URDF physics in the PX4 powered Gazebo simulator.

This work will enhance simulation testing before the experimental evaluation.

7.2.4 System Control

In Section 6.3.4, the limitations of the proposed control application and control solution were examined. Specifically, simplifications were used to restrict disturbances to forces primarily affecting the position dynamics, with particular emphasis on altitude dynamics, which encompassed factors such as point mass varying loads and the uniform force ground effect.

The scope of the control problem in this thesis can be extended by applying the proposed STSMC-AHOSMO to the attitude dynamics and replacing the existing classical PID controllers. This extension presents an opportunity for a more complete solution when handling the system perturbations caused by varying payloads and ground effect. Existing literature [62] has successfully done this with the non-adaptive version. The HOSMO has proven to estimate unknown system disturbance torques within the attitude dynamics, where the AHOSMO from this thesis would allow for an improved practical application in these challenging flight conditions.

7.2.5 Trajectory Generation

The trajectory generation framework implemented in this thesis, albeit simplistic, served as a foundation for the autonomous pick-and-place mission. The initial approach relied on the linear waypoint trajectory generation module provided by COEX, which offered position setpoints. However, modifications were made to incorporate additional feedforward terms such as velocity and acceleration to enhance trajectory tracking. Although, this was limited to low speed applications because of the discontinuity in the velocity leading to trajectory overshoot. The complex trajectory generation was accomplished using pre-set mathematical functions such as a Lemniscate of Bernoulli, where higher order derivatives are available.

Future work could focus on

- Implementing minimum snap trajectory generation methods.

- Consider more advanced methods such as an optimal control solver with Bayesian optimization to optimize a cost function between trajectory segments.

It is import to ensure smooth trajectories for feedforward control applications.

Bibliography

[1] Glenn Taylor, "Retail Tech: Stripe Raises $6.5 Billion, Ikea Deploys 100 Drones." Jun. 2023, [Online]. Available: https://sourcingjournal.com/topics/technology/retail-tech-ikea-verity-stripe-locus-robotics-radial-geek-extensiv-tradebeyond-tecsys-volumental-hilos-424170/.

[2] Simona Shemer, "Walmart Pilots On-Demand Drone Deliveries With Israel's Flytrex." Jun. 2023, [Online]. Available: https://nocamels.com/2020/09/drone-delivery-flytrex-walmart-pilot-service-north-carolina/.

[3] Senay Boztas, "Electric air taxis being developed for Paris Olympics in 2024." Jun. 2023, [Online]. Available: https://www.theguardian.com/world/2023/mar/21/sb-paris-taxis.

[4] Irrigation Management Drones, "Irrigation Management Drones Assist With Crop Development." Jun. 2023, [Online]. Available: https://specdrones.us/agriculture/irrigation-management-drones/.

[5] COEX, "COEX Clover." Jun. 2023, [Online]. Available: https://clover.coex.tech/en/.

[6] COEX, "COEX Pix." Jun. 2023, [Online]. Available: https://clover.coex.tech/en/coex_pix.html.

[7] Raspberry Pi, "Raspberry Pi 4." Jun. 2023, [Online]. Available: https://www.zdnet.com/product/raspberry-pi-4-model-b/.

[8] Raspberry Pi, "Raspberry Pi Camera OV5647 Sensor Module." Jun. 2023, [Online]. Available: https://www.amazon.ca/.

[9] COEX, "ArUco marker detection." Jun. 2023, [Online]. Available: https://clover.coex.tech/en/aruco_marker.html.

[10] OptiTrack Shop, "Flexible 13." Jun. 2023, [Online]. Available: https://www.optitrack-shop.de/flex-serie/kameras/125/flex-13.

[11] Beamques, "VL53L1X Time-of-flight Chip Laser Ranging For Flight Distance Measurement Sensor." Jun. 2023, [Online]. Available: https://www.beamqus.com/product/mcu-531-vl53l1x-time-of-flight-chip-laser-ranging-for-flight-distance-measurement-sensor-400cm-distance-measurement-2/.

[12] Catherine Thorbecke, “FAA clears way for commercial drone deliveries with new rules.” Jun. 2023, [Online]. Available: https://www.wbal.com/article/493538/113/faa-clears-way-for-commercial-drone-deliveries-with-new-rules.

[13] Alex Hern, “Amazon claims first successful Prime Air drone delivery.” Jun. 2023, [Online]. Available: https://www.theguardian.com/technology/2016/dec/14/amazon-claims-first-successful-prime-air-drone-delivery.

[14] Kris Holt, “DoorDash is piloting drone deliveries with Wing in Australia.” Jun. 2023, [Online]. Available: https://www.engadget.com/doordash-wing-drone-delivery-alphabet-australia-171015755.html.

[15] Willie Vogt, “Guardian Agriculture gets FAA nod to fly in U.S..” Jun. 2023, [Online]. Available: https://www.farmprogress.com/technology/guardian-agriculture-gets-faa-nod-to-fly-in-u-s-.

[16] A. Fellan, C. Schellenberger, M. Zimmermann, and H. D. Schotten, “Enabling communication technologies for automated unmanned vehicles in industry 4.0,” in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 171–176, 2018.

[17] Jessica Reed, “Zipline Partners With Government of Rwanda for Autonomous Drone Delivery Services.” Jun. 2023, [Online]. Available: https://www.aviationtoday.com/2022/12/15/zipline-partners-government-rwanda-autonomous-drone-delivery-services/.

[18] Helena Treeck, “Manned First Flight Writes Aviation History.” Jun. 2023, [Online]. Available: https://www.volocopter.com/newsroom/manned-first-flight-writes-aviation-history/.

[19] A. Mishra and S. Behra, “Design of an aerial manipulation system with robotic claw for search and rescue operations,” in *2020 IEEE Bombay Section Signature Conference (IBSSC)*, pp. 224–230, 2020.

[20] H. Abaunza González, P. Castillo Garcia, A. Victorino, and R. Lozano, “Dual quaternion modeling and control of a quad-rotor aerial manipulator,” *Journal of Intelligent & Robotic Systems*, vol. 88, pp. 1–17, 12 2017.

[21] S. Sadr, S. Ali, S. Moosavian, and P. Zarafshan, “Dynamics modeling and control of a quadrotor with swing load,” *Journal of Robotics*, vol. 2014, 11 2014.

[22] H. B. Khamseh, F. Janabi-Sharifi, and A. Abdessameud, “Aerial manipulation-a literature survey,” *Robotics and Autonomous Systems*, vol. 107, pp. 221–235, 2018.

[23] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, “Practical aerial grasping of unstructured objects,” in *2011 IEEE Conference on Technologies for Practical Robot Applications*, pp. 99–104, 2011.

[24] L. Y. Lee, O. A. Syadiqeen, C. P. Tan, and S. G. Nurzaman, “Closed-structure compliant gripper with morphologically optimized multi-material fingertips for aerial grasping,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 887–894, 2021.

[25] D. Kim and P. Y. Oh, “Toward lab automation drones for micro-plate delivery in high throughput systems,” in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 279–284, 2018.

[26] U. A. Fiaz, M. Abdelkader, and J. S. Shamma, “An intelligent gripper design for autonomous aerial transport with passive magnetic grasping and dual-impulsive release,” in *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 1027–1032, 2018.

[27] M. Lieret, J. Lukas, M. Nikol, and J. Franke, “A lightweight, low-cost and self-diagnosing mechatronic jaw gripper for the aerial picking with unmanned aerial vehicles,” *Procedia Manufacturing*, vol. 51, pp. 424–430, 2020.

[28] D. Rus and M. T. Tolley, “Design, fabrication and control of soft robots,” *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.

[29] J. Fishman, S. Ubellacker, N. Hughes, and L. Carlone, “Dynamic grasping with a ”soft” drone: From theory to practice,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4214–4221, 2021.

[30] Y. Yang, K. Jin, H. Zhu, G. Song, H. Lu, and L. Kang, “A 3d-printed fin ray effect inspired soft robotic gripper with force feedback,” *Micromachines*, vol. 12, no. 10, p. 1141, 2021.

[31] M. Ng, F. Vanegas, K. Morton, J. Sandino, and F. Gonzalez, “Design and flight test of an aerial manipulator for applications in gps-denied environments,” in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 20–29, 2022.

[32] R. Cano, C. Pérez, F. Pruano, A. Ollero, and G. Heredia, “Mechanical design of a 6-dof aerial manipulator for assembling bar structures using uavs,” in *2nd RED-UAS 2013 workshop on research, education and development of unmanned aerial systems*, vol. 218, 2013.

[33] M. Xu, T. Senoo, and T. Takaki, “An underactuated parallel-link gripper for a multicopter capable of plane perching,” *ROBOMECH Journal*, vol. 9, no. 1, pp. 1–12, 2022.

[34] M. E. Mumcuoğlu, I. Yüksel, and E. Altuğ, “Design of an automatic item pick-up system for unmanned aerial vehicles.,” *Celal Bayar University Journal of Science*, vol. 16, no. 1, 2020.

[35] D. Kim and P. Y. Oh, “Lab automation drones for mobile manipulation in high throughput systems,” in *2018 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–5, 2018.

[36] T. W. Danko, K. P. Chaney, and P. Y. Oh, “A parallel manipulator for mobile manipulating uavs,” in *2015 IEEE international conference on technologies for practical robot applications (TePRA)*, pp. 1–6, IEEE, 2015.

[37] V. Ghadiok, J. Goldin, and W. Ren, “Autonomous indoor aerial gripping using a quadrotor,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4645–4651, 2011.

[38] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, “Grasping from the air: Hovering capture and load stability,” in *2011 IEEE International Conference on Robotics and Automation*, pp. 2491–2498, 2011.

[39] B.-C. Min, J.-H. Hong, and E. T. Matson, “Adaptive robust control (arc) for an altitude control of a quadrotor type uav carrying an unknown payloads,” in *2011 11th International Conference on Control, Automation and Systems*, pp. 1147–1151, 2011.

[40] B. J. Emran, J. Dias, L. Seneviratne, and G. Cai, “Robust adaptive control design for quadcopter payload add and drop applications,” in *2015 34th Chinese Control Conference (CCC)*, pp. 3252–3257, 2015.

[41] M. Vahdanipour and M. Khodabandeh, “Adaptive fractional order sliding mode control for a quadrotor with a varying load,” *Aerospace Science and Technology*, vol. 86, pp. 737–747, 2019.

[42] D. C. Robinson, H. Chung, and K. Ryan, “Computational investigation of micro rotorcraft near-wall hovering aerodynamics,” in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1055–1063, 2014.

[43] J. Fishman, S. Ubellacker, N. Hughes, and L. Carlone, “Dynamic grasping with a ”soft” drone: From theory to practice,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4214–4221, 2021.

[44] X. Kan, J. Thomas, H. Teng, H. G. Tanner, V. Kumar, and K. Karydis, “Analysis of ground effect for small-scale uavs in forward flight,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3860–3867, 2019.

[45] P. Sanchez-Cuevas, G. Heredia, and A. Ollero, “Characterization of the aerodynamic ground effect and its influence in multirotor control,” *International Journal of Aerospace Engineering*, vol. 2017, pp. 1–17, 08 2017.

[46] M. Orsag, C. Korpela, S. Bogdan, and P. Oh, “Dexterous aerial robots—mobile manipulation using unmanned aerial systems,” *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1453–1466, 2017.

[47] S. Smith, S. Buchanan, and Y.-J. Pan, “A linkage-based gripper design with optimized data transmission for aerial pick-and-place tasks,” in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, (Seattle, USA), June-July 2023.

[48] A. R. Council, I. Cheeseman, and W. Bennett, *The Effect of the Ground on a Helicopter Rotor in Forward Flight.* ARC techncial report, H.M. Stationery Office, 1957.

[49] L. Danjun, Z. Yan, S. Zongying, and L. Geng, “Autonomous landing of quadrotor based on ground effect modelling,” in *2015 34th Chinese Control Conference (CCC)*, pp. 5647–5652, 2015.

[50] X. He, G. Kou, M. Calaf, and K. K. Leang, “In-ground-effect modeling and nonlinear-disturbance observer for multirotor unmanned aerial vehicle control,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 7, 2019.

[51] A. Cesetti, E. Frontoni, A. Mancini, P. Zingaretti, and S. Longhi, “A vision-based guidance system for uav navigation and safe landing using natural landmarks,” *Journal of Intelligent and Robotic Systems*, vol. 57, pp. 233–257, 01 2010.

[52] T. H. Anh, N. T. Binh, and J. W. Song, “In-ground-effect model based adaptive altitude control of rotorcraft unmanned aerial vehicles,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 794–801, 2022.

[53] N. Xuan Mung and N. P. Nguyen, “Adaptive altitude flight control of quadcopter under ground effect and time-varying load: theory and experiments,” *Journal of Vibration and Control*, pp. 1–11, 10 2021.

[54] K. Nonaka and H. Sugizaki, “Integral sliding mode altitude control for a small model helicopter with ground effect compensation,” in *Proceedings of the 2011 American Control Conference*, pp. 202–207, 2011.

[55] J. Bartholomew, A. Calway, and W. Mayol-Cuevas, “Learning to predict obstacle aerodynamics from depth images for micro air vehicles,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4967–4973, 2014.

[56] J. Bartholomew, A. Calway, and W. Mayol-Cuevas, “Improving mav control by predicting aerodynamic effects of obstacles,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4826–4833, 2015.

[57] W. Luo, H. Eschmann, and P. Eberhard, "Gaussian process regression-augmented nonlinear model predictive control for quadrotor object grasping," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 11–19, 2022.

[58] A. Matus-Vargas, G. Rodriguez-Gomez, and J. Martinez-Carranza, "Ground effect on rotorcraft unmanned aerial vehicles: A review," *Intelligent Service Robotics*, vol. 14, no. 1, pp. 99–118, 2021.

[59] P. Wei, S. N. Chan, S. Lee, and Z. Kong, "Mitigating ground effect on mini quadcopters with model reference adaptive control," *International Journal of Intelligent Robotics and Applications*, vol. 3, pp. 283–297, 2019.

[60] H. Du, Z. Pu, J. Yi, and H. Qian, "Advanced quadrotor takeoff control based on incremental nonlinear dynamic inversion and integral extended state observer," in *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, pp. 1881–1886, 2016.

[61] S. Smith and Y.-J. Pan, "Observer-based super-twisting sliding mode control of quadcopters," in *Proceedings of the 2022 CSME International Congress of Canadian Mechanical Engineering*, vol. 5, 2022.

[62] V. K. Tripathi, A. K. Kamath, L. Behera, N. K. Verma, and S. Nahavandi, "Finite-time super twisting sliding mode controller based on higher-order sliding mode observer for real-time trajectory tracking of a quadrotor," *IET Control Theory & Applications*, vol. 14, no. 16, pp. 2359–2371, 2020.

[63] I. Gonzalez-Hernandez, F. M. Palacios, S. S. Cruz, E. S. E. Quesada, and R. L. Leal, "Real-time altitude control for a quadrotor helicopter using a super-twisting controller based on high-order sliding mode observer," *International Journal of Advanced Robotic Systems*, vol. 14, no. 1, p. 1729881416687113, 2017.

[64] J.-J. E. Slotine, W. Li, *et al.*, *Applied nonlinear control*, vol. 199. Prentice hall Englewood Cliffs, NJ, 1991.

[65] J. A. Moreno, "Lyapunov approach for analysis and design of second order sliding mode algorithms," *Sliding modes after the first decade of the 21st century: State of the art*, pp. 113–149, 2012.

[66] R. R. Nair and L. Behera, "Robust adaptive gain higher order sliding mode observer based control-constrained nonlinear model predictive control for spacecraft formation flying," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 367–381, 2018.

[67] A. Levant, "Robust exact differentiation via sliding mode technique," *automatica*, vol. 34, no. 3, pp. 379–384, 1998.

[68] J. A. Moreno, “Lyapunov function for levant’s second order differentiator,” in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 6448–6453, 2012.

[69] A. Poznyak and K. (Firm), *Advanced Mathematical Tools for Automatic Control Engineers: Deterministic techniques*. No. v. 1 in Advanced Mathematical Tools for Automatic Control Engineers: Deterministic Techniques, Elsevier, 2008.

[70] J. Acosta, A. Dòria-Cerezo, and E. Fossas, “Stabilisation of state-and-input constrained nonlinear systems via diffeomorphisms: A sontag’s formula approach with an actual application,” *International Journal of Robust and Nonlinear Control*, vol. 28, no. 13, pp. 4032–4044, 2018.

[71] A. Bacciotti and L. Rosier, *Liapunov functions and stability in control theory*. Springer Science & Business Media, 2005.

[72] F. A. Ortiz-Ricardez, T. Sánchez, and J. A. Moreno, “Smooth lyapunov function and gain design for a second order differentiator,” in *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 5402–5407, 2015.

[73] T. Sánchez and J. A. Moreno, “A constructive lyapunov function design method for a class of homogeneous systems,” in *53rd IEEE Conference on Decision and Control*, pp. 5500–5505, 2014.

[74] A. Chalanga, S. Kamal, L. M. Fridman, B. Bandyopadhyay, and J. A. Moreno, “Implementation of super-twisting control: Super-twisting and higher order sliding-mode observer-based approaches,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 6, pp. 3677–3685, 2016.

[75] T. Luukkonen, “Modelling and control of quadcopter,” *Independent research project in applied mathematics, Espoo*, vol. 22, no. 22, 2011.

[76] Q. Lu, B. Ren, and S. Parameswaran, “Uncertainty and disturbance estimator-based global trajectory tracking control for a quadrotor,” *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 3, pp. 1519–1530, 2020.

[77] A. Erasmus, “Stabilization of a rotary wing unmanned aerial vehicle with an unknown suspended payload,” m.s. thesis, Stellenbosch University, 2020.

[78] E. Tal and S. Karaman, “Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness,” *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 1203–1218, 2021.

[79] S. Waslander and C. Wang, “Wind disturbance estimation and rejection for quadrotor position control,” in *AIAA Infotech@ Aerospace conference and AIAA unmanned... Unlimited conference*, p. 1983, 2009.

[80] S. Waslander and C. Wang, “Wind disturbance estimation and rejection for quadrotor position control,” in *AIAA Infotech@ Aerospace conference and AIAA unmanned... Unlimited conference*, p. 1983, 2009.

[81] W. Johnson, “Helicopter theory, ser,” 1994.

[82] P. G. Ioppo, *The design, modelling and control of an autonomous tethered multirotor UAV*. PhD thesis, Stellenbosch: Stellenbosch University, 2017.

[83] ArduCam, “Arducam MINI 5MP OV5647 Wide Angle Camera Module for Raspberry Pi.” Jun. 2023, [Online]. Available: https://www.arducam.com/product/arducam-mini-ov5647-wide-angle-b0370/.

[84] D. Vongsathorn, G. Johnson, S. Buchanan, and C. Bleau, “Design and control of uav grippers and transportation mechanisms,” design project, Dalhousie University, 2022.

[85] H. Lee and Y. Choi, “Stackable 4-bar mechanisms and their robotic applications,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2792–2797, 2010.

[86] D. Saha, S. Sanfui, R. Kabiraj, and S. Das, “Design and implementation of a 4-bar linkage gripper,” *IOSR Journal of Mechanical and Civil Engineering*, vol. 11, pp. 61–66, 09 2014.

[87] OpenCV, “OpenCV Library.” Jun. 2023, [Online]. Available: https://opencv.org/.

[88] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, “Practical aerial grasping of unstructured objects,” in *2011 IEEE Conference on Technologies for Practical Robot Applications*, pp. 99–104, 2011.

[89] X. Kan, J. Thomas, H. Teng, H. G. Tanner, V. Kumar, and K. Karydis, “Analysis of ground effect for small-scale uavs in forward flight,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3860–3867, 2019.

[90] A. Levant, “Higher-order sliding modes, differentiation and output-feedback control,” *International journal of Control*, vol. 76, no. 9-10, pp. 924–941, 2003.

[91] A. Levant, “Homogeneity approach to high-order sliding mode design,” *Automatica*, vol. 41, no. 5, pp. 823–830, 2005.

[92] D. Y. Negrete-Chávez and J. A. Moreno, “Second-order sliding mode output feedback controller with adaptation,” *International Journal of Adaptive Control and Signal Processing*, vol. 30, no. 8-10, pp. 1523–1543, 2016.

[93] J. Liu, S. Laghrouche, M. Harmouche, and M. Wack, “Adaptive-gain second-order sliding mode observer design for switching power converters,” *Control Engineering Practice*, vol. 30, pp. 124–131, 2014.

[94] P. Ioannou and A. Datta, “Robust adaptive control: a unified approach,” *Proceedings of the IEEE*, vol. 79, no. 12, pp. 1736–1768, 1991.

[95] M. Cocetti, S. Tarbouriech, and L. Zaccarian, “High-gain dead-zone observers for linear and nonlinear plants,” *IEEE Control Systems Letters*, vol. 3, no. 2, pp. 356–361, 2019.

[96] D. Astolfi, L. Marconi, L. Praly, and A. Teel, “Sensitivity to high-frequency measurement noise of nonlinear high-gain observers,” *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 862–866, 2016.

[97] G. Palli, S. Strano, and M. Terzo, “A novel adaptive-gain technique for high-order sliding-mode observers with application to electro-hydraulic systems,” *Mechanical Systems and Signal Processing*, vol. 144, p. 106875, 2020.

[98] A. Levant, “Sliding order and sliding accuracy in sliding mode control,” *International journal of control*, vol. 58, no. 6, pp. 1247–1263, 1993.

[99] V. Utkin, “Discussion aspects of high-order sliding mode control,” *IEEE Transactions on Automatic Control*, vol. 61, no. 3, pp. 829–833, 2015.

[100] J. Davila, L. Fridman, and A. Levant, “Second-order sliding-mode observer for mechanical systems,” *IEEE Transactions on Automatic Control*, vol. 50, no. 11, pp. 1785–1789, 2005.

[101] F. Plestan, Y. Shtessel, V. Bregeault, and A. Poznyak, “New methodologies for adaptive sliding mode control,” *International journal of control*, vol. 83, no. 9, pp. 1907–1919, 2010.

[102] J. A. Moreno, “A lyapunov approach to output feedback control using second-order sliding modes,” *IMA Journal of Mathematical Control and Information*, vol. 29, no. 3, pp. 291–308, 2012.

[103] M. T. Angulo, L. Fridman, and A. Levant, “Robust exact finite-time output based control using high-order sliding modes,” *International Journal of Systems Science*, vol. 42, no. 11, pp. 1847–1857, 2011.

[104] J.-J. E. Slotine, J. K. Hedrick, and E. A. Misawa, “On sliding observers for nonlinear systems,” in *1986 American Control Conference*, pp. 1794–1800, 1986.

[105] C. Lin, S. Sun, P. Walker, and N. Zhang, “Accelerated adaptive second order super-twisting sliding mode observer,” *IEEE Access*, vol. 7, pp. 25232–25238, 2019.

[106] J. A. de Loera and F. Santos, “An effective version of pólya’s theorem on positive definite forms,” *Journal of Pure and Applied Algebra*, vol. 108, no. 3, pp. 231–240, 1996.

[107] C. N. Delzell, “Impossibility of extending pólya’s theorem to “forms” with arbitrary real exponents,” *Journal of Pure and Applied Algebra*, vol. 212, no. 12, pp. 2612–2622, 2008.

[108] J. A. Moreno and M. Osorio, “Strict lyapunov functions for the super-twisting algorithm,” *IEEE Transactions on Automatic Control*, vol. 57, no. 4, pp. 1035–1040, 2012.

[109] J. A. Moreno and M. Osorio, “A lyapunov approach to second-order sliding mode controllers and observers,” in *2008 47th IEEE Conference on Decision and Control*, pp. 2856–2861, 2008.

[110] A. Bacciotti and L. Rosier, *Liapunov functions and stability in control theory.* Springer Science & Business Media, 2005.

[111] H. K. Khalil, “Nonlinear systems third edition,” *Patience Hall*, vol. 115, 2002.

[112] J. A. Moreno and M. Osorio, “Strict lyapunov functions for the super-twisting algorithm,” *IEEE Transactions on Automatic Control*, vol. 57, no. 4, pp. 1035–1040, 2012.

[113] M. J. Mirzaei, M. A. Hamida, and F. Plestan, “Super-twisting control of offshore wind turbine in region iii with self-tuning adaptive gains,” in *2022 16th International Workshop on Variable Structure Systems (VSS)*, pp. 172–177, 2022.

[114] H. A. B. Anuar, F. Plestan, A. Chriette, and O. Kermorgant, “Super-twisting sliding mode control with adaptive gain of quadrotor with rigid manipulator,” in *2022 16th International Workshop on Variable Structure Systems (VSS)*, pp. 53–58, 2022.

[115] C.-H. Lin and F.-Y. Hsiao, “Proportional-integral sliding mode control with an application in the balance control of a two-wheel vehicle system,” *Applied Sciences*, vol. 10, no. 15, p. 5087, 2020.

[116] L. Tan, J. Gao, Y. Luo, and L. Zhang, “Super-twisting sliding mode control with defined boundary layer for chattering reduction of permanent magnet linear synchronous motor,” *Journal of Mechanical Science and Technology*, vol. 35, pp. 1829–1840, 2021.

[117] A. Levant, “Principles of 2-sliding mode design,” *automatica*, vol. 43, no. 4, pp. 576–586, 2007.

[118] Y. Wang, Y. Feng, X. Zhang, J. Liang, and X. Cheng, “New reaching law control for permanent magnet synchronous motor with extended disturbance observer,” *IEEE Access*, vol. 7, pp. 186296–186307, 2019.

[119] M. Faessler, A. Franchi, and D. Scaramuzza, “Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 620–626, 2018.

[120] E. Tal and S. Karaman, “Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness,” *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 1203–1218, 2021.

[121] M. Faessler, A. Franchi, and D. Scaramuzza, “Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories,” *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, 11 2017.

[122] D. Brescianini, M. Hehn, and R. D’Andrea, “Nonlinear quadrocopter attitude control: Technical report,” tech. rep., ETH Zurich, 2013.

[123] I. D. Landau, R. Lozano, M. M’Saad, and A. Karimi, *Adaptive control: algorithms, analysis and applications.* Springer Science & Business Media, 2011.

[124] D. Bonvin and A. Karimi, “Identification de systemes dynamiques,” *Lausanne, October*, 2006.

[125] B. Farhang-Boroujeny, *Adaptive filters: theory and applications.* John Wiley & Sons, 2013.

[126] J. Coelho, J. B. Cunha, and P. d. M. Oliveira, “Recursive parameter estimation algorithms,” in *Controlo 2004-Sixth Portuguese Conference on Automatic Control*, 2004.

[127] T. Yamamoto, K. Fujii, and M. Kaneda, “Design and implementation of a self-tuning pid controller,” *IFAC Proceedings Volumes*, vol. 31, no. 22, pp. 59–64, 1998.

[128] M. M. Abdelhameed, “Enhancement of sliding mode controller by fuzzy logic with application to robotic manipulators,” *Mechatronics*, vol. 15, no. 4, pp. 439–458, 2005.

[129] G. Palli, S. Strano, and M. Terzo, “Sliding-mode observers for state and disturbance estimation in electro-hydraulic systems,” *Control Engineering Practice*, vol. 74, pp. 58–70, 2018.

[130] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, “Design, modeling, estimation and control for aerial grasping and manipulation,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2668–2673, 2011.

Appendix A

Modelling Calculations

A.1 Ground Effect Model Identification

From Section 3.3.2, the ground effect model (3.21) can be characterized by the ratio of input thrust to output thrust (or accelerations), expressed as

$$\frac{a^{out}_{PX4}}{a^{in}_{PX4}} \approx \frac{1}{1-\rho\left(\frac{r}{4z_r}\right)^2}. \tag{A.1}$$

Ideally, a^{out}_{PX4} is equal to a^{in}_{PX4} outside of ground effect. In general, the overall thrust generated by the rotors is equal to the quadrotor's weight and its vertical acceleration $a^{out}_{PX4} = g + a_z$. However, measuring this accurately without sophisticated experimental setups or detailed motor models can be challenging. In [49], it is assumed the vertical acceleration is zero during hover, letting the thrust be constant where $a^{out}_{PX4} \approx g$.

The coefficient ρ in the ground effect model was identified through multiple hovering experiments at various altitudes, and the corresponding results are presented in Fig. 3.3. The input acceleration, denoted as a^{in}_{PX4}, was determined using the following equation

$$a^{in}_{PX4} = \delta_{T_{PX4}} \frac{g}{f^{HT}_{PX4}}, \tag{A.2}$$

where $\delta_{T_{PX4}}$ is constrained to the direction defined by $\vec{b}_{z_r}$, which is ideally vertical during hover ($a^r_x = a^r_y = 0$). The output acceleration, denoted as a^{out}_{PX4}, was determined as follows

$$a^{out}_{PX4} = g + \tilde{a}_z. \tag{A.3}$$

The filtered acceleration $\tilde{a}_z$ was calculated from the hovering phase using the PX4 uLogs and was expected to be close to zero ($a^{out}_{PX4} \approx g$). The ratio of $G(z)$ was determined for each altitude, and utilizing this data, function (3.21) was fitted using MATLAB's nonlinear least squares curve fitting.

Appendix B

Quadrotor Control System Gains

This Appendix provides detailed information about the COEX Clovers control system gains, applicable to both the PX4-powered Gazebo simulator and the hardware setup used in experimental studies.

For the hardware application, a set of base gains is utilized through an airframe module provided by COEX, specifically designed for the onboard PX4 firmware. To optimize the gains, the adaptive auto-tune module discussed in Section 5.5.3 is employed, in combination with adjustments to the PID gains of the position control module. The STSMC gains are also included, consisting of the super-twisting algorithm gains, the sliding manifold gain, and the configured HOSMO gains being adaptive or manually set.

In the case of the Gazebo simulator, the gains defined by COEX for the PX4 firmware settings are utilized. These gains have been found to perform effectively within the simulator environment.

B.1 COEX Clover Controller Gains

The angular rate controller gains are given in Table B.1.

Table B.1: The angular rate controller gains.

Gazebo			
Controller	**P Gain**	**I Gain**	**D Gain**
Roll Rate	0.150	0.20	0.0030
Pitch Rate	0.150	0.20	0.0030
Yaw Rate	0.20	0.10	0
Hardware			
Roll Rate	0.098706	0.13149	0.0023715
Pitch Rate	0.094438	0.121586	0.00175668
Yaw Rate	0.110602	0.12078	0.00220357

The angle controller gains are given in Table B.2.

Table B.2: The angle controller gains.

Gazebo	
Controller	**P Gain**
Roll Angle	6.50
Pitch Angle	6.50
Yaw Angle	2.80
Hardware	
Roll Angle	6.50
Pitch Angle	6.50
Yaw Angle	6.50

The velocity controller gains are given in Table B.3.

Table B.3: The velocity controller gains.

Gazebo			
Controller	**P Gain**	**I Gain**	**D Gain**
Longitudinal Velocity	1.80	0.40	0.20
Lateral Velocity	1.80	0.40	0.20
Heave Velocity	4.0	2.0	0
Hardware			
Longitudinal Velocity	2.4	0.4	0.26
Lateral Velocity	2.4	0.4	0.26
Heave Velocity	4.8	2.0	0

The position controller gains are given in Table B.4.

Table B.4: The position controller gains.

Gazebo	
Controller	**P Gain**
North Position	0.95
East Position	0.95
Down Position	1.0
Hardware	
North Position	1.72
East Position	1.72
Down Position	1.25

B.1.1 Robust Controller and Observer

To assist in selecting appropriate gains, Section 5.4.4 provides useful relationships, specifically $k_1 = 1.5\sqrt{\Delta}$ and $k_2 = 1.1\Delta$. These relationships can be considered when adjusting the gains, taking into account the specific characteristics of the system and the anticipated disturbance levels. For the Gazebo simulations, the Dryden wind gust model in Table 6.4 provided a starting point for gain setting, in hardware they were iteratively tuned for performance.

The STSMC-HOSMO gains are given in Table B.5 with the AHOSMO represented by the Adaptive Terms.

Table B.5: Robust controller and observer gains.

Gazebo			
Super-Twisting Sliding Mode Controller	$\mathbf{c_1}$ **Gain**	$\mathbf{k_1}$ **Gain**	$\mathbf{k_2}$ **Gain**
Longitudinal Dynamics	3	5.2	6.2
Lateral Dynamics	3	5.2	6.2
Heave Dynamics	3	5.5	6.5
Higher Order Sliding Mode Observer	$\boldsymbol{\lambda}_1$	$\boldsymbol{\lambda}_2$	$\boldsymbol{\lambda}_3$
Longitudinal Dynamics	15	18	30
Lateral Dynamics	15	18	30
Heave Dynamics	15	18	32
Adaptive Terms	$\boldsymbol{\Lambda}_0$	$\mathbf{k_a}$ **Gain**	$\boldsymbol{\mu}$
Longitudinal Dynamics	0.003	1.5	0.1
Lateral Dynamics	0.003	1.5	0.1
Heave Dynamics	0.003	1.7	0.1
Hardware			
Super-Twisting Sliding Mode Controller	$\mathbf{c_1}$ **Gain**	$\mathbf{k_1}$ **Gain**	$\mathbf{k_2}$ **Gain**
Heave Dynamics	6	9.5	15
Adaptive Terms	$\boldsymbol{\Lambda}_0$	$\mathbf{k_a}$ **Gain**	$\boldsymbol{\mu}$
Heave Dynamics	0.002	3.0	0.1

Appendix C

Author's Publication List

S. Smith and Y. -J Pan, "Observer-based super-twisting sliding mode control of quadcopters," in *Proceedings of the 2022 CSME International Congress of Canadian Mechanical Engineering*, vol. 5, 2022.

S. Smith, S. Buchanan, and Y.-J. Pan, "A linkage-based gripper design with optimized data transmission for aerial pick-and-place tasks," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Seattle, USA, 2023.

S. Smith and Y. -J Pan, "Adaptive observer-based super-twisting sliding mode control for low altitude quadcopter grasping," in *IEEE/ASME Transactions on Mechatronics*, 2023. (Submitted for review, Manuscript ID: TMECH-05-2023-15546).

Appendix D

COEX Clover International Aerial Robotics Project

I actively participated in the CopterHack 2023 international project competition, simultaneously working on an external project alongside my thesis program. This event brought together 17 teams from across the globe, showcasing cutting-edge open-source aerial robotics projects that had been meticulously developed over a span of 7 months. Ultimately, the competition reached a final of 7 teams, representing countries like Russia, India, Belarus, Brazil, and Canada, where we presented our innovative projects. These projects were built upon the foundation of the COEX Clover platform, incorporating various essential components such as PX4, OpenCV, ROS, MAVLink, and MAVROS.

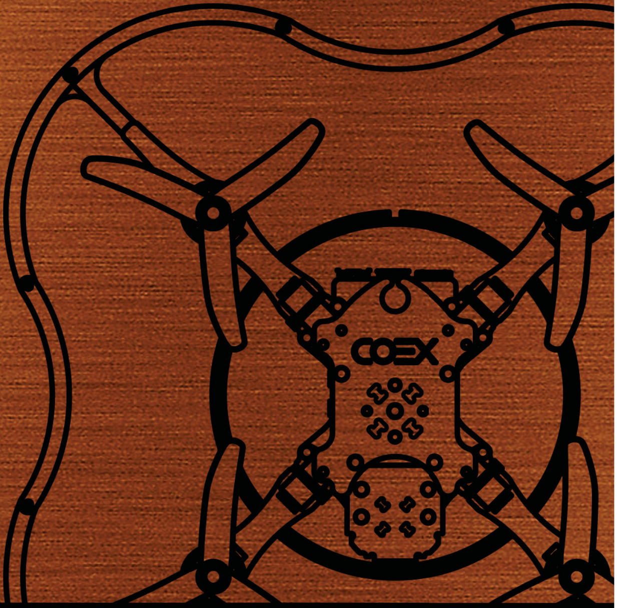
I am proud to share that my project, focused on integrating the COEX Clover platform with the Motion Capture System, achieved a 3rd place in the competition. The core of my work revolved around creating an educational document that delved into various aspects of this integration. Notably, the document covered a wide range of topics, including PX4 auto-tuning, sensor fusion, data transmission, experimental setups, complex trajectory tracking, Software-in-the-Loop simulations, Hardware-in-the-Loop simulations, low-level control, feedforward control, control theory, and high-level trajectory generation.

The full details of the project can be found in the following Gitbook educational documentation: https://0406hockey.gitbook.io/mocap-clover/ and the final project video can be found here https://youtu.be/j0ovjo0aBpQ.

3 PLACE

COEX

COPTER HACK '23



DIPLOMA

AWARDED TO

Sean Smith

FOR THE 3rd PLACE IN THE INTERNATIONAL PROJECTS COMPETITION COPTERHACK 2023

Oleg Ponfilenok
CEO of COEX