



**Landing of a Quadrotor UAV on a Moving Platform
using Visual Servoing Techniques**

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF

DOCTOR OF PHILOSOPHY

BY

YOGESH KUMAR

(PHD18110)

ELECTRONICS AND COMMUNICATION ENGINEERING
INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

NEW DELHI- 110020

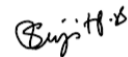
August 5, 2025

THESIS CERTIFICATE

This is to certify that the thesis titled **Landing of a Quadrotor UAV on a Moving Platform using Visual Servoing Techniques**, submitted by **Yogesh Kumar**, to the Indraprastha Institute of Information Technology (IIIT), Delhi, for the award of the degree of **Doctor of Philosophy**, is a bonafide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.



Sayan Basu Roy
Thesis Supervisor
Assistant Professor
Dept. of Electronics and Communication
IIIT Delhi, 110020



P.B. Sujit
Thesis Co-Supervisor
Professor
Dept. of Electrical Engineering &
Computer Science
IISER Bhopal, 462066

Place: New Delhi

Date: August 5, 2025

ACKNOWLEDGEMENTS

This thesis wouldn't have been possible without many individuals' direct and indirect support. First and foremost, I would like to express my deepest gratitude to my advisors, Dr. Sayan Basu Roy and Dr. P.B. Sujit, for their mentorship, insightful discussions, and constant encouragement throughout my PhD journey. Beyond their technical guidance, Dr. Sujit has imparted essential academic skills, including academic writing, staying abreast of the latest literature, and identifying practical research gaps, all of which have profoundly influenced the direction of my work. I owe much of the mathematical and analytical skills I have developed to Dr. Roy. His thoughtful approach to discussing complex theoretical concepts and calm demeanor in navigating challenging problems have profoundly influenced my growth as a researcher. I aspire to emulate his ability to dissect intricate details with clarity as a future mentor. I am also immensely grateful to my yearly review committee members, Dr. Shobha Sundar Ram and Dr. Sanat K. Biswas, for their constructive feedback, thoughtful suggestions, and valuable guidance in refining and improving this thesis. Their insights have been crucial in shaping the final outcome of my research. I would like to express my gratitude to Dr. Girish Chowdhary at UIUC for hosting me as a visiting researcher. During my time there, I had the opportunity to explore the emerging field of neuro-adaptive controllers, which played a pivotal role in extending his work and making a significant contribution to this thesis. Additionally, I am deeply thankful to all the professors at IIIT Delhi, whose teachings have played a crucial role in shaping my academic and research journey.

The IIIT Delhi truly felt like home, thanks to the camaraderie and support of my friends and colleagues, including Bushra Ansari, Tushar Garg, Amith Manoharan, Pranjali Bhatia, Rajan Girsu, Kshitiz Srivastava, and Shivangi Agarwal. I would also like to extend my gratitude to my friends from IISER Bhopal, Kartikeya Singh, Saksham Madan, Kasi Vishwanath, Rajat Agarwal, Prithvi Poddar, Swadhin Agarwal, Manav Mishra, Vinita Sao, and Aditya Sinha for their encouragement and support throughout my journey. I could not have made it this far without our tea-fueled (and sometimes more-than-tea) discussions, shared laughter, free-food lines, and even impromptu lab

cricket matches. I am equally thankful to the many other friends-too numerous to name individually-whose presence, conversations, and quiet encouragement have been invaluable along the way.

I owe a profound debt of gratitude to my parents for their blessings, unconditional love, and belief in me, which have been the foundation of my strength. Despite knowing little about the nuances of the PhD journey, they supported me wholeheartedly through this uncertain path. Finally, I am deeply thankful to my wife, Ms. Kamlesh Bajiya, for her unwavering support, patience, and encouragement through every high and low. Her understanding and sacrifices have been my constant source of motivation.

Lastly, I extend my heartfelt gratitude to TCS Research for awarding me the TCS Research Scholar Scholarship. This generous support has been instrumental in enabling me to focus on my research and academic pursuits, allowing me to dedicate the time and resources needed to make meaningful progress in my work.

Yogesh

ABSTRACT

Unmanned aerial vehicles (UAVs), particularly quadrotors, have become indispensable in both military and civilian applications, offering autonomous capabilities that enhance operational efficiency. Among the many challenges faced during autonomous UAV missions, landing is one of the most critical tasks, requiring precise localization of the landing spot and smooth, controlled trajectories for safe and accurate landings. The quadrotor landing problem has significant implications for real-world applications, such as delivery services, search and rescue operations, and military missions, where efficient and precise target tracking and landing are of critical importance. This research aims to address the problem of robust and precise landing of a quadrotor UAV onto a moving planar target. Given the superior tracking capabilities of vision-based control techniques, this work primarily focuses on developing dynamically feasible Image-Based Visual Servoing (IBVS) schemes for quadrotors to land on arbitrarily moving targets.

The thesis is structured around two main sub-problems: (1) the development of a dynamically feasible visual servoing velocity controller that enables the quadrotor to accurately track and land on a planar target with arbitrary linear and angular motion, and (2) the design of a trajectory tracking control system that allows the quadrotor to robustly follow the velocity reference derived from the IBVS controller while accounting for external disturbances, such as the ground effect. This research presents several key contributions, including developing a generalized IBVS method for tracking targets with arbitrary translational motion with realistic target dynamics. This method is further extended into a feasible kinematic IBVS technique for quadrotors, with experimental validation demonstrating its effectiveness. In addition, the thesis discusses the design of a trajectory tracking controller for the quadrotor, considering partial knowledge of desired trajectories (with higher derivatives unknown), which serves as an initial step toward solving the second sub-problem. Further, the thesis focuses on integrating a vision-based dynamic controller for landing on a moving target under state and control input-dependent uncertainties. Simulations considering realistic environments validate

the effectiveness of the proposed controllers, showcasing their ability to achieve precise landing performance despite considering dynamic targets.

Contents

ACKNOWLEDGEMENTS	i
ABSTRACT	iii
LIST OF FIGURES	xii
NOTATION	xiii
1 Introduction	1
1.1 Literature Review	2
1.2 Objectives	4
1.3 Organisation	6
1.4 Contributions of This Thesis	13
2 Classical IBVS Framework for Target Tracking with Arbitrary Translational Motion	14
2.1 Problem Formulation	14
2.1.1 Image Feature Dynamics	15
2.1.2 Control Objective and Error Formulation	15
2.2 Target Velocity Estimator	16
2.3 Control Design	18
2.3.1 Control Law	18
2.3.2 Stability Analysis	19
2.4 Results	24
2.4.1 Target Moving with Constant Velocity	25
2.4.2 Target Moving with Time-Varying Velocity	26
2.5 Conclusion	26
3 Dynamically Feasible IBVS for Quadrotor UAVs Tracking Arbitrary Moving Planar Targets	27
3.1 Problem Formulation	28

3.1.1	Virtual Image Plane	28
3.1.2	Image Features & Dynamics	30
3.1.3	Control Objective	31
3.1.4	Quadrotor Kinematic Control	32
3.2	Control Design	33
3.2.1	Translational Velocity Estimator	34
3.2.2	Angular Velocity Estimator	35
3.2.3	Control Law	37
3.2.4	Stability and Convergence Proofs	37
3.3	Results	41
3.3.1	Discrete Implementation	41
3.3.2	SITL Simulations	42
3.3.3	Outdoor Experiments	45
3.3.4	Comparison with Super-twisting Sliding Mode Control	46
3.4	Conclusion	51
4	Adaptive IBVS-based Planar Non-Holonomic Target Tracking for Quadrotors	52
4.1	Problem Formulation	53
4.1.1	Image Feature Dynamics	54
4.1.2	Error Formulation	54
4.1.3	Target Velocity Parameterization	55
4.1.4	Control Objective	56
4.2	Control Design	57
4.2.1	Control Law- Translational Kinematics	57
4.2.2	Control Law- Rotational Kinematics	61
4.2.3	Stability and Convergence - Translation Kinematics	62
4.2.4	Stability and Convergence - Rotational Kinematics	64
4.3	Results	65
4.3.1	Simulation Environments	66
4.3.2	MATLAB Environement	66
4.3.3	SITL Environment	66
4.3.4	Discretization	68

4.3.5	MATLAB Simulations-Method I	69
4.3.6	MATLAB Simulation-Method II	69
4.3.7	SITL Simulations-Method I	73
4.3.8	SITL Simulations-Method II	75
4.4	Conclusion	75
5	Partially Known Trajectory Tracking Control for Quadrotors	80
5.1	Problem Formulation and Preliminaries	81
5.1.1	Notations	81
5.1.2	Multi-rotor Dynamics	81
5.1.3	Control Objective	82
5.2	Control Design	84
5.2.1	Position Tracking Controller	84
5.2.2	Estimate of $\ddot{x}_d(t)$	85
5.2.3	Auxiliary Desired Attitude Construction	87
5.2.4	Effect of large attitude errors on $f(t)$	88
5.2.5	Attitude Tracking Controller	89
5.3	Theoretical Analysis	90
5.3.1	Position Tracking Error System	90
5.3.2	Attitude Tracking Error System	92
5.3.3	Stability Analysis	92
5.4	Results	106
5.4.1	Example 1	107
5.4.2	Example 2	109
5.5	Conclusion	109
6	Asynchronous DNN-based Control for Systems with State and Control Input Dependent Uncertainties	111
6.1	Problem Formulation	111
6.1.1	DNN Representation of Uncertainty	112
6.2	Control Design	114
6.2.1	Non-Affine Control Law	115
6.2.2	Outer Layer Weight Update Law	116

6.2.3	Basis Function Training	117
6.2.4	Lyapunov Analysis	118
6.3	Results	119
6.3.1	Control Input Synthesis	120
6.3.2	Simulation Results	121
6.3.3	Experimental Results	123
6.4	Conclusion	129
7	Vision-Based Dynamic Control for Quadrotor Tracking and Landing on Moving Targets	130
7.1	Problem Formulation	130
7.1.1	Quadrotor Dynamics	130
7.1.2	IBVS Preliminaries for Quadrotor	132
7.1.3	Landing as an IBVS problem	133
7.1.4	Translational Dynamics Revisited	134
7.2	Control Design	136
7.2.1	Translational Motion Controller	137
7.2.2	Attitude Tracking Controller	142
7.2.3	Stability Analysis	144
7.3	Results	146
7.3.1	Ground Effect Model:	147
7.3.2	Artificial State and Input-Dependent Uncertainties:	149
7.4	Conclusion	151
8	Conclusion and Opportunities for Further Research	152
8.1	Future Research Directions	153
8.1.1	Incorporating Field of View Constraints	153
8.1.2	Marker Free IBVS	154
8.1.3	Efficient Differentiators for Noisy Signals	154
8.1.4	Hybrid Visual Servoing for High and Low-Speed Target Tracking	154
8.1.5	Real-World Testing of the Proposed IBVS-based Landing Controller	155

LIST OF PAPERS BASED ON THESIS

Journals

1. Kumar, Y., Roy, S.B. and Sujit, PB, 2024. Geometric tracking control of a multi-rotor UAV for partially known trajectories. *International Journal of Robust and Nonlinear Control*, 34(14), pp.9572-9595.
2. Kumar, Y., Shamsi, B.P., Roy, S.B. and Sujit, PB, 2024. Tracking a planar target using image-based visual servoing technique. *IEEE Transactions on Intelligent Vehicles*, 9(3), pp.4362-4372.
3. Kumar, Y., Roy, S.B., Chowdhary, G. and Sujit, PB, 2025. Asynchronous DNN-based control for systems with state and control input dependent uncertainties. Submitted to *Aerospace Science and Technology*.
4. Kumar, Y., Roy, S.B. and Sujit, PB, 2025. Neuro-adaptive IBVS for quadrotor landing on moving targets with uncertainties. In Preparation.

Conferences

5. Kumar, Y., Roy, S.B., and Sujit, PB. Adaptive ibvs based planar non-holonomic target tracking for quadrotors. In 2024 International Conference on Unmanned Aircraft Systems (ICUAS) (pp. 201-208). IEEE, 2024.
6. Kumar, Y., Roy, S.B., and Sujit, PB. Dynamic image-based visual servoing for quadrotor to track a planar target with unknown motion. In 2022 IEEE 61st Conference on Decision and Control (CDC) (pp. 1398-1403). IEEE, 2022.
7. Kumar, Y., Roy, S.B., and Sujit, PB. Velocity estimator augmented image-based visual servoing for moving targets with time-varying 3-d motion. In 2021 60th IEEE Conference on Decision and Control (CDC) (pp. 5187-5192). IEEE, 2021.
8. Kumar, Y., Roy, S.B., and Sujit, PB. Barrier lyapunov function based trajectory tracking controller for autonomous vehicles with guaranteed safety bounds. In 2020 International Conference on Unmanned Aircraft Systems (ICUAS) (pp. 722-728). IEEE, 2020.
9. Kumar, Y., Roy, S.B., and Sujit, PB. Exponentially convergent adaptive image-based visual servoing of quadrotor for tracking planar dynamic target," In 2025 American Control Conference (ACC) (Accepted). IEEE, 2025.

List of Figures

1.1	Illustration of vision-based landing problem.	4
1.2	Research Objective	5
2.1	IBVS control architecture	19
2.2	Simulation results for constant target velocity	25
2.3	Simulation results for time-varying target velocity	26
3.1	Coordinate frames illustration. A coordinate frame $\{A\}$ is represent by axes $\hat{x}_A$, $\hat{y}_A$ and $\hat{z}_A$	29
3.2	Virtual features illustration for a target with four visual points features (i.e., $N = 4$) represented by $\circ$ in the virtual plane. The point (x_g, y_g) represents the centroid of the target in the virtual plane. a in the a_n represents the sum of the squared distances (i.e. $\sum_{k=1}^4 l_k^2$) of each point feature from the centroid. The scalar feature $\alpha(t)$ is calculated using the centered image moments of order two as given in equation (3.4).	31
3.3	Control architecture	34
3.4	SITL environment with ground rover and quadrotor.	43
3.5	SITL results for linear and constant target motion	44
3.6	SITL results for time-varying target motion	44
3.7	Outdoor setup	45
3.8	Outdoor experiment results. (a) Feature tracking error (e_s, e_α) (b) Target motion estimates ($\hat{v}_{p_x}$ and $\hat{v}_{p_y}$ are the estimates of first two elements of target velocity vector $v_p(t)$)	46
3.9	Comparative MIL simulation results.	49
3.10	SITL results with ST controller	50
3.11	SITL simulation control input. (a) Control input with proposed controller (b)Control input with ST controller	50
4.1	Control architecture and the simulation environment flow chart for Method-I	67
4.2	Control architecture and the simulation environment flow chart for Method-II	67
4.3	MATLAB simulation results considering translational kinematics controller as given in Method I.	70

4.4	MATLAB simulation results considering translational kinematics controller as given in Method II.	71
4.5	Comparative MATLAB simulation results with the controller given in Method I and Method II	72
4.6	GAZEBO simulation results for Scenario I	74
4.7	GAZEBO simulation results for Scenario II	76
4.8	SITL simulation results with the Method II controller	77
4.9	Comparative SITL simulation results with the controller given in Method I and II	78
5.1	Coordinate frames illustration. The x, y, z axes of a coordinate frame $\{A\}$ are denoted by x_A, y_A and z_A , respectively.	83
5.2	Control architecture	84
5.3	Filters illustration	86
5.4	Simulation results for Example 1. [18] in there error norm plot represents plot with controller in Lee <i>et al.</i> (2010)	108
5.5	Simulation results for Example 2. [18] in there error norm plot represents plot with controller in Lee <i>et al.</i> (2010).	110
6.1	Illustration of control architecture.	114
6.2	Illustration of switching of Basis function weights over t_j and uncertainty inference for any time $t \in [t_{j-1}, t_j)$	115
6.3	Simulation results using Method 1	122
6.4	Simulation results using Method 2	122
6.5	Simulation results using Method 3	123
6.6	Tracking error norms with different basis function learning settings.	124
6.7	Experimental setup	126
6.8	Example 1 results	127
6.9	Example 2 results	128
7.1	Coordinate frames illustration. The x, y, z axes of a coordinate frame $\{A\}$ are denoted by $\hat{x}_A, \hat{y}_A$ and $\hat{z}_A$, respectively.	131
7.2	Control architecture.	137
7.3	Simulation results demonstrating trajectory tracking (top and side views), feature tracking error, and velocity estimation error with the proposed controller.	148
7.4	Simulation results without DNN-based uncertainty mitigation.	148

7.5	Simulation results for artificial uncertainty with the proposed controller.	150
7.6	Simulation results without DNN-based uncertainty mitigation, showing trajectory plots (top and side views) and feature tracking error. . . .	150

NOTATION

$R_{\zeta}(\Phi) \in \text{SO}(3)$	Rotation matrix representing the rotation of angle Φ about the ζ -axis
$R_x^y \in \text{SO}(3)$	Rotation matrix from frame $\{x\}$ to $\{y\}$
$\text{SO}(3)$	Special orthogonal group of $\{R \in \mathbb{R}^{3 \times 3} R^T R = R R^T = I, \det(R) = 1\}$
$\mathfrak{so}(3)$	Group of skew symmetric matrices $\{S \in \mathbb{R}^{3 \times 3} S = -S^T\}$
$\phi, \theta, \text{ and } \psi$	Roll, pitch, and yaw angles
$S_k(.) : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$	Skew-symmetric map
$\mathbb{R}$	Real Space
$\mathbb{R}^n$	Real Vector Space of dimension n
$\mathbb{R}_+$	Non-negative Real Space
$\mathbb{R}^{n \times n}$	Real Matrix Space of dimension $n \times n$
I	Identity Matrix with appropriate dimension
$A > 0 (< 0)$	Positive (Negative) Definite Matrix
$\lambda_{\min}(A)$	Minimum eigen-value of the Matrix A
$\lambda_{\max}(A)$	Maximum eigen-value of the Matrix A
$\det(A)$	Determinant of matrix A

Chapter 1

Introduction

Unmanned Aerial Vehicles (UAVs) are becoming an essential tool for both military and civilian applications. Among the various types of UAVs, quadrotors (rotary type of UAV) are particularly popular due to their versatile capabilities and suitability for autonomous operations. Quadrotors find applications in diverse fields such as search and rescue, object transportation, inspection, surveillance, and mapping. Their compact size, maneuverability, and ability to operate autonomously make them an ideal choice for these tasks Doitsidis *et al.* (2012); Gupte *et al.* (2012). However, the utility of quadrotors is significantly constrained by their limited flight endurance, which poses challenges in applications requiring prolonged, continuous flight. In applications such as package delivery, search and rescue, and others, where continuous flight is required for a long duration, the UAVs alone cannot accomplish the task entirely by themselves Falanga *et al.* (2017); Paris *et al.* (2019). Allowing these vehicles to autonomously land on predefined targets for battery charging/swapping or delivery of supplies can greatly enhance their usefulness. This capability would reduce or eliminate the need for human intervention, allowing for extended operation times and greater autonomy.

Collaboration between quadrotors and ground-based vehicles offers a promising solution for increasing operational efficiency and extending range. Researchers have explored the potential of truck-UAV delivery systems, where a moving truck serves as a mobile base for UAVs. In such systems, UAVs can deliver parcels independently and return to the truck—whether stationary or in motion—for recharging, battery replacement, or to pick up new deliveries Ham (2018); Ferrandez *et al.* (2016). This concept, employed in delivery services such as Amazon's Prime Air, eliminates the need for UAVs to return to fixed distribution centers after each delivery. Instead, the truck and UAV can work in tandem, optimizing the entire delivery process. Beyond commercial applications, coordination between moving vehicles and quadrotors has significant implications in military operations. Such cooperation is often critical for mission success, enabling enhanced flexibility, range, and efficiency in demanding environments.

Autonomous landing is one of the most critical maneuvers in UAV missions, as it requires precise localization of the landing spot and smooth trajectory planning to ensure a safe and controlled touchdown Gautam *et al.* (2014). The complexity further increases when a UAV must land on a moving target, where both spatial and temporal precision are paramount for successful and efficient landing. Over the years, numerous approaches have been proposed to address challenges in autonomous landing and target tracking Bhargavapuri *et al.* (2019); Gautam *et al.* (2014); Falanga *et al.* (2017); Báa *et al.* (2019). Despite these advancements, the authors of Báa *et al.* (2019) observed that, "only a few of these works have demonstrated deployment in real-world conditions, and none of them have presented a reliable performance that enables repeated landing on a fast-moving helipad in a demanding outdoor environment". This conclusion was drawn based on evaluations conducted by the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) 2017 scientific board, organized by Khalifa University of Science in Abu Dhabi. In the MBZIRC competition, the framework proposed in Báa *et al.* (2019) showcased the most reliable results within the controlled competition setting. However, the authors acknowledged that their approach was specifically tailored to the competition's requirements. A more generalized approach to target motion estimation is essential for real-world applications, where target motion is often arbitrary and time-varying. Autonomous landing on a moving vehicle remains a promising and open research direction with significant potential for further exploration and innovation.

1.1 Literature Review

Targets for UAV landing can be broadly classified into cooperative and non-cooperative categories. Cooperative targets are equipped with systems capable of actively transmitting their pose, which can act as a reference trajectory for the quadrotor. However, even in cooperative cases, the transmitted pose information may be corrupted, especially if it relies on GPS data. GPS signals are prone to errors caused by multipath effects, signal blockage, or environmental interference, which can degrade the accuracy and reliability of the pose information. On the other hand, non-cooperative targets do not share any information, requiring the UAV to rely entirely on its onboard sensors for tracking and localization. For such cases, augmented sensors like cameras are particularly effec-

tive, as they provide rich visual data that can be used to estimate the target’s position and orientation in real time. Vision-based target tracking is therefore preferred in both cooperative and non-cooperative scenarios due to its higher precision and adaptability across a wide range of environments Bhargavapuri *et al.* (2019); Falanga *et al.* (2017); Paris *et al.* (2019); Báa *et al.* (2019).

Visual Servoing (VS) is a class of control techniques that uses visual information for robot motion control. In scenarios that require high precision, such as tracking and landing on a moving target using quadrotors, visual servoing techniques have shown promise as visual measurement provides good tracking performance. The two main visual servoing approaches are (1) position-based visual servoing (PBVS), which performs position reconstruction based on visual information for control laws, and (2) image-based visual servoing (IBVS), where control calculations are performed directly in the image plane Corke (2013). Achieving the necessary precision in position reconstruction parameters such as target model depth, intrinsic parameters, and 3D geometry can be challenging in position-based visual servoing (PBVS). To address this, the authors in Lee *et al.* (2012) proposed an image-based visual servoing (IBVS) approach, which performs control calculations primarily in the 2D image space rather than relying on a 3D body or inertial reference frame. This eliminates the need for 3D position reconstruction, simplifying the control process. Despite its advantages, IBVS techniques face challenges when dealing with unknown target motion, which introduces uncertainties into the IBVS error dynamics. In the literature on IBVS for underactuated systems, these uncertainties are often addressed through robust conservative components in the controller Lin *et al.* (2022); Jabbari Asl (2019) or by employing ad-hoc feedback compensation methods. Such methods include optical flow techniques Serra *et al.* (2016); Huang *et al.* (2021); Herissé *et al.* (2012), Extended Kalman Filters Li *et al.* (2021b); Báa *et al.* (2019), and others, albeit with restrictive assumptions on target motion dynamics Abdessameud and Janabi-Sharifi (2013, 2015).

Another critical phenomenon affecting precise landing is the proximity (or ground) effect. During the final phase of landing, when the quadrotor is in close proximity to the landing surface, an additional thrust is generated, augmenting the total thrust vector and leading to uncertainties in the landing time Danjun *et al.* (2015); Bernard *et al.* (2017). This proximity (or ground) effect occurs due to a reduction in the induced drag of the aircraft, which increases its lift-to-drag ratio. Danjun et al. Danjun *et al.* (2015) devel-

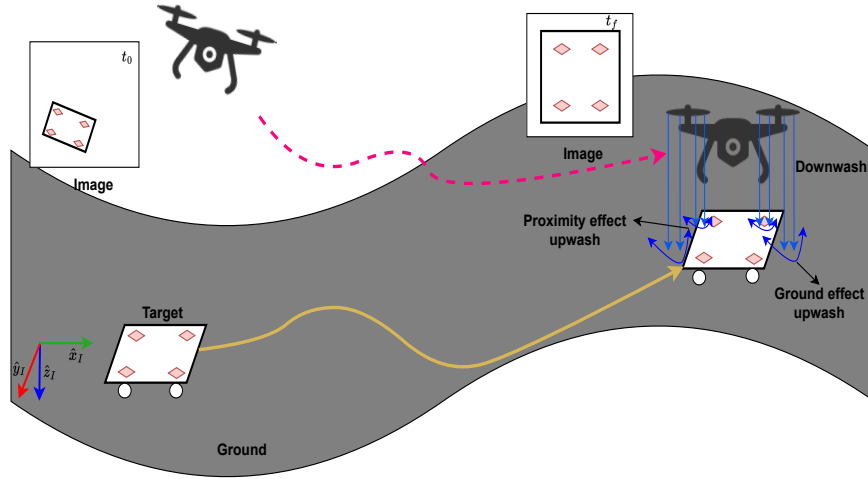


Figure 1.1: Illustration of vision-based landing problem.

oped a controller based on a ground effect model for static landing scenarios. However, landing on a moving target introduces more complexities than static target landings. In such cases, the quadrotor performs translational motions in the X and Y planes, causing the thrust vector to deviate from alignment with the body Z-axis. Consequently, the ground effect model proposed in Danjun *et al.* (2015); Bernard *et al.* (2017) may not be suitable for dynamic landing applications. To address this limitation, the authors in Pedro Sanchez-Cuevas and Ollero (2017) proposed a model that accounts for the partial ground effect on the entire quadrotor, rather than evaluating the effect on a single rotor and extrapolating it to the others. While this approach is a step forward, a universal proximity effect model that accurately captures the physical interaction between the quadrotor's rotors and the landing surface under dynamic conditions remains unavailable in the literature. Developing such a model is crucial for improving the reliability and precision of UAV landings on moving targets.

1.2 Objectives

Considering the above-identified challenges in the state of the art, this thesis aims to address the problem of robust and precise landing of a quadrotor UAV onto an arbitrarily moving planar target. Leveraging the superior tracking accuracy of vision-based control techniques, the research aims to design dynamically feasible image-based visual servoing (IBVS) schemes tailored for this purpose. Additionally, the thesis aims to

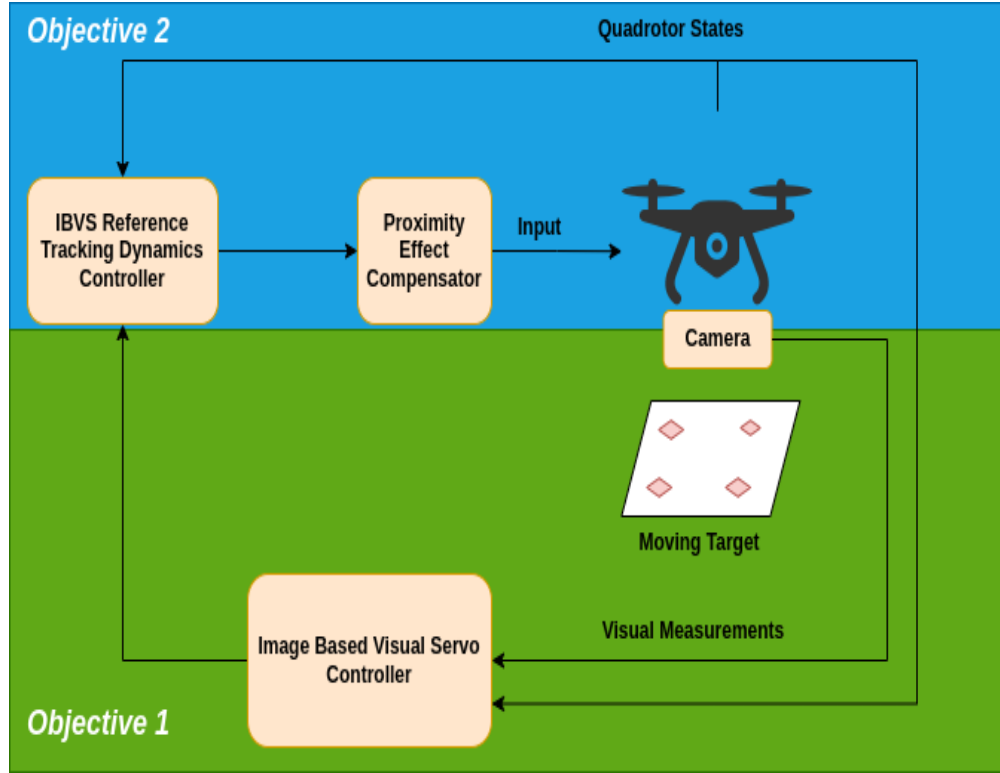


Figure 1.2: Research Objective

develop controllers with formal stability guarantees, emphasizing reproducibility and repeatability to enhance their practical applicability and reliability. An illustration of the problem statement is given in Figure 1.1. To achieve this goal, the research focuses on solving the following two interconnected sub-problems, as illustrated in Figure 1.2:

Problem 1 (Outer Loop)

- Develop a dynamically feasible visual servoing velocity controller for a quadrotor.
- The controller should enable precise tracking and landing on a moving planar target with unknown linear and angular motion.

Problem 2 (Inner Loop)

- Design a trajectory tracking controller for quadrotor dynamics.
- The controller must ensure the quadrotor can accurately follow the velocity references generated by the visual servo controller.
- Additionally, it should exhibit robustness to external disturbances, such as proximity (ground) effects.

The combination of these two solutions will provide a comprehensive framework to address the challenges of autonomous landing on dynamically moving targets.

1.3 Organisation

The thesis is organized into eight chapters, each written in a self-consistent manner to ensure ease of understanding without requiring heavy cross-referencing between chapters. This structure is designed to enhance readability and guide the reader through the research. The works presented in Chapters 2, 3, and 4 discuss the solutions to Objective Problem 1, while Chapters 5 and 6 provide solutions for Objective Problem 2. Finally, Chapter 7 integrates these contributions to present a robust vision-based framework for moving target tracking and landing for the quadrotor. A brief summary of the content covered in each chapter, along with motivation, is outlined as follows:

Chapter 1

This chapter introduces the motivation behind the research, focusing on the challenges of autonomous landing for quadrotors on moving targets. It highlights the limitations in existing methods, the significance of vision-based tracking, and the impact of the proximity effect. The chapter also outlines the research objectives, which include developing a robust and dynamically feasible landing system through visual servoing and trajectory tracking controllers for quadrotors.

Chapter 2

Chapter 2 marks the initial step toward addressing Objective Problem 1 defined in section 1.2. This chapter lays the foundation for understanding the intricacies of IBVS and its applicability to dynamic target tracking, serving as a critical step in achieving the objective.

Classical IBVS controllers are designed under the assumption of a stationary target, making them unsuitable for scenarios involving moving targets. In such cases, the unknown target motion introduces additional tracking errors that must be addressed.

Early approaches, such as tracking error integration Chaumette *et al.* (1991) and non-smooth differentiator-based estimates Chaumette and Santos (1993), provide solutions for constant target velocities but lack robustness for time-varying motions. Filtering methods Bensalah and Chaumette (1995); Hashimoto and Kimura (1995) and Structure and Motion (SaMfM) techniques Dani *et al.* (2010, 2011) have been proposed, but they impose restrictive assumptions, such as straight-line or planar motion and persistence of excitation conditions. Concurrent learning methods Bell *et al.* (2018) offer global convergence but remain untested in practical IBVS scenarios. Recent works have attempted to address these gaps, such as single-point feature tracking Shahriari *et al.* (2013), but fail to estimate target velocity along the optical axis. Unscented Kalman Filter-based approaches Li *et al.* (2019, 2021b) tackle issues like intermittent updates but rely on accurate target motion modeling, which is often impractical. While some studies Herissé *et al.* (2012); Abdessameud and Janabi-Sharifi (2013) explore IBVS for specific target motions, there is limited research on IBVS for arbitrarily bounded target motion in 3D space, leaving a significant gap in addressing dynamic target tracking comprehensively.

This chapter proposes a generalized IBVS method to track a target with arbitrary translational motion. A target velocity estimation law is designed using dynamic surface-like filters and incorporated into the control law (6-DOF velocity reference) to compensate for the uncertainties due to the unknown target motion.

Chapter 3

Chapter 3 advances the development of IBVS control by focusing on its application to quadrotors. Unlike traditional platforms, quadrotors are underactuated systems capable of tracking only 4-DOF references—three for translational motion and one for heading. This poses significant challenges, especially when dealing with targets exhibiting angular motion, as the coupled Jacobian in the point feature-based formulation from Chapter 2 complicates the control design. This chapter addresses these challenges by adapting the velocity reference generated in Chapter 2 to ensure feasibility and compatibility with quadrotor dynamics, laying the groundwork for robust and practical implementation of IBVS on underactuated systems.

The challenge of dealing with the coupled Jacobian in classical IBVS literature has

been addressed by employing feature sets based on spherical image projection and perspective image moments, as seen in Bourquardez *et al.* (2009). However, a comparative study in Ozawa and Chaumette (2013) pointed out that spherical feature-based methods lack proper stabilization during vertical motion, whereas image moment-based methods perform better when the image plane is parallel to the target plane. Motivated by this requirement, Jabbari Asl *et al.* (2014) introduced the concept of a virtual image plane invariant to the pitch and roll motion of the quadrotor. While these approaches have made significant strides, they primarily focus on static targets.

Similar to the classical IBVS literature, extending IBVS for moving targets introduces additional challenges due to the uncertainties in the feature error dynamics. In the IBVS literature for underactuated systems, these challenges are often addressed by introducing a robust conservative component in the controller Jabbari Asl (2019); Lin *et al.* (2022), which increases the overall control effort or by deriving feedback compensation for target velocity using ad hoc techniques. These techniques include optical flow Serra *et al.* (2016); Huang *et al.* (2021); Herissé *et al.* (2012), which is computationally demanding, extended Kalman filtering Li *et al.* (2021b), which requires a target motion model limiting its versatility Abdessameud and Janabi-Sharifi (2013, 2015), and nonlinear observer-based target velocity estimators that are limited to constant target velocities Li *et al.* (2021a). Furthermore, no theoretical results are provided to support the estimation performance and its impact on the IBVS controller for these ad hoc methods.

This chapter addresses these complexities by designing a control strategy suitable for a quadrotor's underactuated system, ensuring practical feasibility while maintaining robustness for dynamic target tracking.

Chapter 4

Building upon the kinematic IBVS controller developed in Chapter 3, this chapter extends the framework of virtual image plane-based features, which eliminate explicit dependence on feature depth. Introducing the desired feature depth concept as a substitute for precise depth information, the approach in Chapter 3 simplified control design. However, the simultaneous tracking of moving targets with unknown desired depth re-

mains a significant challenge and has been largely overlooked in existing literature Jabbari Asl (2019).

We present two control design approaches for the problem at hand. The first part formulates and solves a bi-linear parametric uncertainty estimation problem, greatly expanding the feasible range of target motions for effective feedback compensation-based solutions. Although the first approach successfully tackles the challenge of tracking moving targets with unknown desired depth, it does not provide formal guarantees regarding the convergence of parameter estimation errors. This raises a critical question: Can parameter convergence be formally ensured, and how does it impact the performance of feature tracking? The second part of the chapter is dedicated to addressing these limitations. By rigorously analyzing the closed-loop error dynamics, the proposed methods establish conditions under which parameter estimation errors converge exponentially.

Chapter 5

Chapter 5 marks the progression of this thesis towards addressing dynamic control challenges for quadrotors, specifically in the context of integrating the reference velocity commands generated by the IBVS controller (outer loop) into the quadrotor's inner loop dynamics. This work is pivotal in bridging the gap between the kinematic-level controller developed in earlier chapters and the dynamic behavior of quadrotors, which are inherently underactuated systems with highly nonlinear and coupled dynamics.

The commonly used multi-rotor UAVs, of which quadrotors are a prominent subclass, demand specialized trajectory tracking controllers due to their inherent underactuation and complex dynamics Mahony *et al.* (2012). While advances in hardware and software have brought theoretical designs closer to real-world applications, new challenges arise from evolving use cases. A notable challenge is tracking a desired trajectory when its complete details are unavailable. Such scenarios often occur in non-cooperative target tracking or coordination tasks with other vehicles, where acquiring comprehensive target motion data—beyond pose measurements—is impractical due to hardware limitations or security constraints Thomas *et al.* (2017); Chen *et al.* (2021); Tyagi *et al.* (2021).

Non-cooperative tracking problems often leverage visual sensors, employing methods such as position-based visual servoing (PBVS) Chen *et al.* (2016); Thomas *et al.* (2017); Gomez-Balderas *et al.* (2013) or image-based visual servoing (IBVS) Li *et al.* (2021b); Serra *et al.* (2016); Jabbari Asl (2019); Abdessameud and Janabi-Sharifi (2015). However, compensating for unknown target dynamics remains a significant hurdle. Robust controllers or feedback compensation techniques, such as optical flow Serra *et al.* (2016); Abdessameud and Janabi-Sharifi (2015); Herissé *et al.* (2012) or Kalman filtering Thomas *et al.* (2017); Sanchez-Lopez *et al.* (2014); Li *et al.* (2021b), are often employed but come with restrictive assumptions about the target motion. Our work compiled in Chapters 3, and 4 tackled these challenges at the kinematic level by proposing an IBVS framework for quadrotors that mitigates the reliance on fully known target trajectories.

Nevertheless, dynamic control introduces additional complexities, as inner-loop attitude and translational controllers often demand knowledge of higher derivatives of the target dynamics. Addressing this requirement typically involves high-gain controllers Jabbari Asl (2019) or command filter-based methods Farrell *et al.* (2009); Lin *et al.* (2022). Recent advances, such as those in Boss *et al.* (2020), have employed high-gain controllers using Euler angle formulations to estimate and manage higher-order dynamics. However, the singularity issues inherent in Euler angles Lee *et al.* (2010) limit their applicability for nontrivial trajectories.

This chapter introduces a novel geometric control framework for dynamic trajectory tracking, overcoming the limitations of Euler angle-based methods and ensuring robust and stable performance. By leveraging a geometric control formulation, the proposed control design guarantees smooth, singularity-free operation for quadrotors. While this chapter focuses on developing a generalized controller for dynamic trajectory tracking, the subsequent chapters will build upon these developments to address the objectives outlined in 1.2, specifically achieving robust and precise landing of a quadrotor UAV onto an arbitrarily moving planar target.

Chapter 6

Chapter 6 addresses a critical challenge in quadrotor systems—the presence of uncertainties during dynamic landing, particularly those introduced by the ground (or proximity) effect. This phenomenon, which arises when a quadrotor is near a landing surface, results in state- and control input-dependent uncertainties. Specifically, the ground effect depends on the quadrotor’s height (state) and thrust generated by its propellers (control input). The unstructured nature of this uncertainty makes it difficult to model using conventional basis functions, necessitating the use of more flexible tools such as deep neural networks (DNNs).

Designing adaptive controllers for uncertain systems has been a long-standing research focus, addressing challenges such as unmodeled dynamics, environmental interactions, and parameter variations Tao (2014). Recent advancements have highlighted the superior expressiveness of DNNs for approximating complex nonlinear functions Rolnick and Tegmark (2017), spurring interest in DNN-based control architectures. These approaches can be broadly categorized as offline or online. Offline methods Shi *et al.* (2019); Nubert *et al.* (2020) rely on extensive data for open-loop approximations but face limitations in generalization and stability. In contrast, online methods employ Lyapunov-based stability principles for real-time weight updates, enabling robust performance in dynamic scenarios Lewis *et al.* (1996); Joshi and Chowdhary (2019); Joshi *et al.* (2021); Patil *et al.* (2021); Mishra *et al.* (2025).

While existing online DNN-based controllers effectively address state-dependent uncertainties Joshi *et al.* (2021); Patil *et al.* (2021, 2025), challenges arise when uncertainties depend on state and control inputs, as seen in the ground effect. This chapter delves into designing a DNN-based controller to handle such coupled uncertainties. Building on Joshi and Chowdhary (2019), the proposed approach employs a composite adaptation law Slotine and Li (1989); Patre *et al.* (2010) to refine the DNN’s outer-layer weights, achieving enhanced tracking accuracy during both transient and steady-state phases. By leveraging the expressiveness of DNNs, the solution provides a robust framework to address unstructured uncertainties like the ground effect, paving the way for precise and reliable quadrotor landings under dynamic conditions.

Chapter 7

Chapter 7 addresses the challenge of enabling quadrotors to dynamically track and land on moving targets with arbitrary and unknown motion while accounting for uncertainties in the quadrotor's dynamics. Building upon the developments in earlier chapters, we propose a vision-based dynamic control framework that integrates target motion estimation, uncertainty adaptation, and precise landing strategies.

In Chapter 3, we established the foundation by designing an image-based visual servoing (IBVS) approach to generate dynamically feasible velocity references for quadrotor target tracking. This was further expanded in Chapter 4, where we enhanced IBVS laws to accommodate targets with uncertain desired depth. Chapter 5 introduced a general trajectory tracking controller to manage partially known trajectories, while Chapter 6 tackled uncertainties that are both state- and control input-dependent.

Leveraging these prior contributions, Chapter 7 develops a robust control strategy that unifies visual feedback with adaptive uncertainty mitigation to achieve reliable tracking and landing on moving targets. The approach incorporates dynamic visual references from IBVS while addressing uncertainties through an adaptive control law inspired by the methodologies discussed in Chapter 6. Stability and tracking convergence are rigorously analyzed using Lyapunov theory, ensuring the robustness of the proposed solution.

This chapter represents the final contribution of the thesis, showcasing a comprehensive solution to the problem of quadrotor tracking and landing on dynamically moving targets under uncertain conditions.

Chapter 8

The final chapter summarizes the key findings and the research contributions. It also reflects on the challenges encountered during the research and offers potential directions for further improvements and applications.

1.4 Contributions of This Thesis

- **Classical IBVS for dynamic target tracking:** This thesis presents a novel IBVS framework to handle arbitrarily moving targets with unknown and time-varying 3D translational motion.
- **Dynamic feasibility of IBVS laws for dynamic target tracking using quadrotor:** The underactuated nature of quadrotors limits their ability to track the full 6-DOF velocity references typically generated by classical IBVS controllers. This thesis proposes a modified IBVS framework that generates dynamically feasible velocity references, enabling robust tracking of moving targets.
- **Depth uncertainty mitigation in IBVS:** This thesis ensures the dynamic feasibility of IBVS control laws using virtual image plane-based features that eliminate the need for precise depth by substituting a desired value. It further explores target motions where this depth requirement can be relaxed, proposing adaptive IBVS controllers to handle unknown desired depth.
- **Partially known trajectory tracking for quadrotor:** The thesis also presents a trajectory-tracking controller for multi-rotor UAVs, addressing the unavailability of higher derivatives of the reference trajectory.
- **Handling control input and state-based uncertainties through DNN-based adaptive control:** To address state- and control input-dependent uncertainties like the ground effect during quadrotor landing, this thesis proposes a real-time DNN-based adaptive control framework.
- **Vision-based dynamic control for landing on moving targets:** As a final contribution, this thesis presents an IBVS-based control framework for quadrotor dynamics tasked with tracking and landing on a moving planar target with arbitrary and unknown motion, incorporating target motion estimation and a DNN-based neuro-adaptive framework for stable, accurate landing.

Chapter 2

Classical IBVS Framework for Target Tracking with Arbitrary Translational Motion

In this chapter, we address the challenging issue of estimating the time-varying target motion to improve the performance of IBVS control using only image plane coordinates of the feature points. A novel smooth and bounded target velocity estimation law is designed building on the philosophy of dynamic surface filter Swaroop *et al.* (2000). However, to avoid the peaking phenomenon due to the low value of dynamic surface filter gain, projection algorithm Khalil (1996) is suitably adopted in the framework. The estimator output is utilized as a compensator in the IBVS control law. An error system consists of tracking error, filter error, and target velocity estimation error is considered, and rigorous Lyapunov analysis is performed to capture the dynamical behavior of the strongly coupled error dynamics. It is shown that using the proposed velocity estimator and controller, the closed-loop error converges exponentially to an ultimate bound for the time-varying target velocity, whereas asymptotic stability can be achieved when the target velocity is constant. The performance of the proposed IBVS controller coupled with the target velocity estimator is verified through simulations using a realistic scenario considering both constant and time-varying target velocities. In contrast to existing literature, the proposed IBVS scheme does not impose any restrictive assumptions on target motion, i.e., it can track a moving target with any arbitrary bounded translational motion in a 3-D space.

2.1 Problem Formulation

The formulation of the IBVS control problem for moving targets is presented in this section.

2.1.1 Image Feature Dynamics

For a 3-D point $P = [x, y, z]^T$, expressed in the camera frame, the respective image plane point feature coordinates $s = [u, v]^T$ can be given as Corke (2013)

$$s = \begin{bmatrix} f \frac{x}{z\rho_w} + u_0 \\ f \frac{y}{z\rho_h} + v_0 \end{bmatrix}, \quad (2.1)$$

where ρ_h and ρ_w are the height and width of each pixel respectively, and f is the focal length of the camera. The principal point (u_0, v_0) is the point where the optical axis intersects the image plane. Consider that the point P is moving with the camera frame velocity $V_p = [v_p^T, \omega_p^T]^T = [V_{p_x}, V_{p_y}, V_{p_z}, 0, 0, 0]^T \in \mathbb{R}^6$, where v_p and ω_p represents the linear and angular velocities respectively. Then the motion of point feature s for a camera moving with spatial velocity $V_c = [v_c, \omega_c]^T = [v_{c_x}, v_{c_y}, v_{c_z}, \omega_{c_x}, \omega_{c_y}, \omega_{c_z}]^T \in \mathbb{R}^6$, where v_c and ω_c represents the linear and angular velocities, can be modeled as Shahriari *et al.* (2013)

$$\dot{s} = L_e(s, z)(V_c - V_p), \quad (2.2)$$

where

$$L_e(s, z) = \begin{bmatrix} -\frac{f'}{z} & 0 & \frac{\bar{u}}{z} & \frac{\bar{u}\bar{v}}{f'} & -\frac{f'^2 + \bar{u}^2}{f'} & \bar{v} \\ 0 & -\frac{f'}{z} & \frac{\bar{v}}{z} & \frac{f'^2 + \bar{v}^2}{f'} & -\frac{\bar{u}\bar{v}}{f'} & -\bar{u} \end{bmatrix}. \quad (2.3)$$

is the interaction matrix, where $\bar{u} = u - u_0$ and $\bar{v} = v - v_0$ are the pixel coordinates relative to the principal point, and z is the depth of the point P in the camera frame. For the typical case $f' = \frac{f}{\rho}$ as $\rho_w = \rho_h = \rho$.

2.1.2 Control Objective and Error Formulation

The control objective is to control the camera motion in Euclidean space such that the image feature s reaches the desired set-point s^* in the image plane. To control the 6-DOF motion of the camera, at least three feature points are required Corke (2013). In this chapter, we work with four feature points associated with four moving (considered to be attached on a target, hence the velocity $V_{p_i} = V_p$, $i = 1, 2, \dots, 4$ will be the same for pure translational motion) 3-D points. The overall dynamics for image feature vector

$s' = [s_1^T, s_2^T, s_3^T, s_4^T]^T \in \mathbb{R}^8$ can be thus defined as

$$\dot{s}' = L_e'(s', z')(V_c - V_p), \quad (2.4)$$

where $z' = [z_1, z_2, z_3, z_4]^T$, and

$$L_e'(s', z') = \begin{bmatrix} L_{e_1}^T & L_{e_2}^T & L_{e_3}^T & L_{e_4}^T \end{bmatrix}^T. \quad (2.5)$$

The respective error vector $e \in \mathbb{R}^8$ for the constant desired $s'^* = [s_1^{*T}, s_2^{*T}, s_3^{*T}, s_4^{*T}]^T \in \mathbb{R}^8$ can be formulated as

$$e = s' - s'^*. \quad (2.6)$$

The open loop error dynamics then can be defined by taking the time derivative of equation (2.6) as

$$\dot{e}(s', z', V_p) = L_e'(s', z')(V_c - V_p). \quad (2.7)$$

To design the control law V_c for the error dynamics given in (2.7), we require the pixel coordinates, depth, and velocity of the features points. But in the ideal IBVS setting, we only have access to pixel coordinates s' , the velocity V_p and the depth vector z' are needed to be supplied/estimated from some other means. The overall control objective is that the error e should converge to zero while compensating for the unknown uncertainties due to the unknown target motion.

Assumption 1. *The true depths of the features and accurate camera intrinsic parameters are assumed to be available.*

Assumption 2. *The target is moving with a velocity such that $\sup_{t \geq 0} \|V_p(t)\| = h_1$ and $\sup_{t \geq 0} \|\dot{V}_p(t)\| = h_2$, where $h_1, h_2 \in \mathbb{R}_+$.*

2.2 Target Velocity Estimator

In this section, we will present a novel estimator to estimate the unknown target velocity $V_p(t)$. The camera velocity is considered to be measurable along with the error signal

$e(t)$ as given in equation (2.6). Let's define two filters with design parameter $\gamma_1 > 0$ as

$$\gamma_1 \dot{e}_f + e_f = e, \quad e_f(0) = 0, \quad (2.8)$$

$$\gamma_1 \dot{h} + h = L_e' V_c, \quad h(0) = 0, \quad (2.9)$$

where $e_f \in \mathbb{R}^8$ and $h \in \mathbb{R}^8$ are the filtered variables of e and $L_e' V_c$ respectively. In order to capture the information about $\dot{e}$, consider a filter structure with filter variable $g \in \mathbb{R}^8$ as

$$\gamma_1 \dot{g} + g = \dot{e}, \quad g(0) = 0. \quad (2.10)$$

Integrating (2.10) for $g(t)$ will yield

$$g(t) = \frac{1}{\gamma_1} (e(t) - e_f(t) - \exp(-\frac{1}{\gamma_1} t) e(0)). \quad (2.11)$$

Lemma 1. Consider a filtered variable $S \in \mathbb{R}^8$ of unknown variable $L_e' V_p$ given as

$$\dot{S} = \frac{1}{\gamma_1} (L_e' V_p - S), \quad S(0) = 0. \quad (2.12)$$

Then using the error dynamics given in equation (2.7) and the filter equations (2.8), (2.9), (2.10) we can write

$$g = h - S. \quad (2.13)$$

Proof. Using equation (2.7) and (2.11) we can write

$$g(t) = \frac{1}{\gamma_1} \int_0^t \exp(-\frac{1}{\gamma_1} (t - \tau)) (L_e' (s', z', \tau) (V_c(\tau) - V_p(\tau))) d\tau. \quad (2.14)$$

Furthermore

$$\begin{aligned} g(t) &= \frac{1}{\gamma_1} \int_0^t \exp(-\frac{1}{\gamma_1} (t - \tau)) (L_e' (s', z', \tau) V_c(\tau)) d\tau \\ &\quad - \frac{1}{\gamma_1} \int_0^t \exp(-\frac{1}{\gamma_1} (t - \tau)) (L_e' (s', z', \tau) V_p(\tau)) d\tau. \end{aligned} \quad (2.15)$$

From equation (2.15), we obtain the relation given in (2.13). $\square$

The estimation of $L_e' V_p$ is then given by

$$S = h - g. \quad (2.16)$$

Based on (2.16), $L_e'^+ S$ can be used as an estimate of the target velocity V_p . However, this estimate can suffer from the peaking phenomenon due to the low value of γ_1 ; which is desirable for estimation efficiency in the long run. To overcome the issue, the following projection operator-based Khalil (1996) differential estimator with design parameter $\gamma_2 > 0$ is proposed for target velocity estimation with $\hat{V}_p(0) = 0$:

$$\dot{\hat{V}}_p = Proj(\hat{V}_p, \phi) = \begin{cases} \phi - \frac{\nabla f(\hat{V}_p)(\nabla f(\hat{V}_p))^T}{\|\nabla f(\hat{V}_p)\|^2} \phi f(\hat{V}_p), & \text{if } f(\hat{V}_p) > 0 \text{ \& } \phi^T \nabla f(\hat{V}_p) > 0 \\ \phi, & \text{otherwise} \end{cases}, \quad (2.17)$$

where

$$\phi = \frac{1}{\gamma_2} (L_e'^+ S - \hat{V}_p), \quad f(\hat{V}_p) = \frac{\|\hat{V}_p\|^2 - h_1^2}{\varepsilon_0 h_1^2}, \quad \nabla f(\hat{V}_p) = \frac{2\hat{V}_p}{\varepsilon_0 h_1^2},$$

where $\varepsilon_0 > 0$ is the tolerance and $L_e'^+$ is the pseudo-inverse of L_e' .

2.3 Control Design

2.3.1 Control Law

For the error dynamics given in (2.7), we design the implementable control law as

$$V_c = -\lambda L_e'^+ e + \hat{V}_p, \quad (2.18)$$

where $\lambda > 0$ is the controller gain. The estimate of target velocity $V_p(t)$ is fed to the controller using the update law given in (2.17). The overall control architecture is given in Fig. 2.1.

Remark 1. *If the interaction matrix L_e' is rank deficient i.e. $\text{rank}(L_e') < 6$ then $L_e'^+$ becomes singular (a rather classical problem in IBVS). Recent literature Pascual-Escudero et al. (2020) suggests that it can be avoided in a four-point IBVS setting by choosing four co-planar points which are not concyclic and no three of them are collinear.*

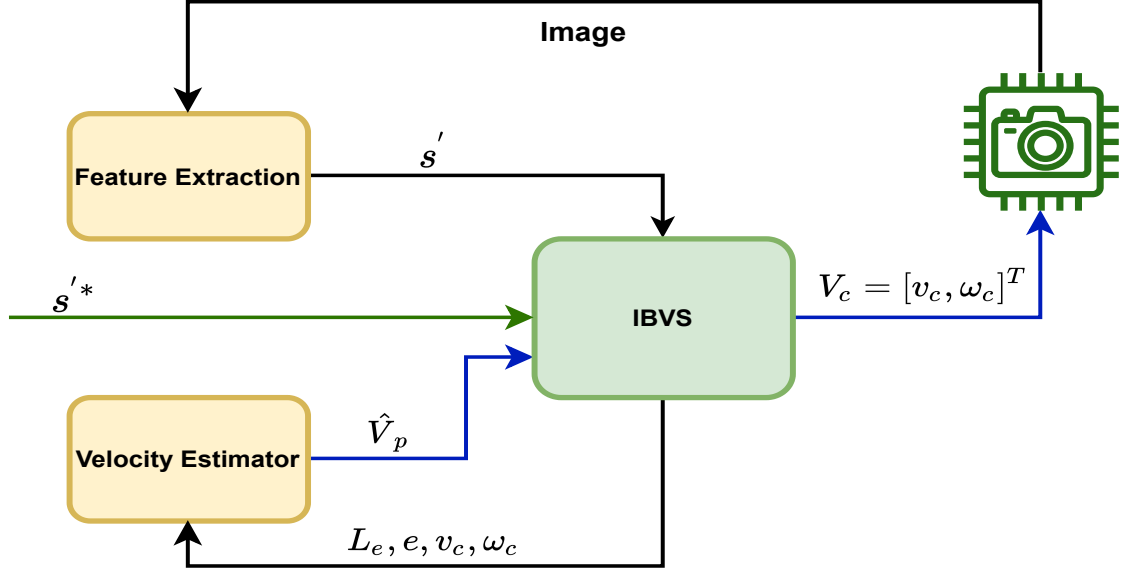


Figure 2.1: IBVS control architecture

2.3.2 Stability Analysis

We define the target velocity estimation error $e_{v_p}(t)$ and the filter error $e_v(t)$ as

$$e_{v_p} = V_p - \hat{V}_p, \quad (2.19)$$

$$e_v = L_e' V_p - S. \quad (2.20)$$

Using the tracking error system (2.7), control law from (2.18), velocity estimation law from (2.17) and equation (2.12), the overall closed loop error dynamics can be written as

$$\begin{bmatrix} \dot{e} \\ \dot{e}_v \\ \dot{e}_{v_p} \end{bmatrix} = \begin{bmatrix} -\lambda e - L_e' e_{v_p} \\ -\frac{1}{\gamma_1} e_v + \dot{L}_e' V_p + L_e' \dot{V}_p \\ \dot{V}_p - \text{Proj}(\hat{V}_p, \phi) \end{bmatrix}. \quad (2.21)$$

Lemma 2. Consider the error dynamics given in (2.21). There exist three functions $\rho_e()$, $\rho_{e^+}()$ and $\rho_1()$ such that

$$\|L_e'\| \leq \rho_e(\|e\|), \quad \|L_e'^+\| \leq \rho_{e^+}(\|e\|), \quad \|\dot{L}_e'\| \leq \rho_1(\|e\|)(\lambda \rho_{e^+}(\|e\|)\|e\| + \|e_{v_p}\|). \quad (2.22)$$

Proof. Consider the state vector $x_i = [x_{1_i}, x_{2_i}, x_{3_i}]^T = [\bar{u}_i, \bar{v}_i, \frac{1}{z_i}]^T$, for point feature $s_i, i =$

1, 2, ..., 4, then the dynamics for x_i can be written as Shahriari *et al.* (2013)

$$\begin{bmatrix} \dot{x}_{1_i} \\ \dot{x}_{2_i} \\ \dot{x}_{3_i} \end{bmatrix} = \begin{bmatrix} -f' x_{3_i} & 0 & x_{1_i} x_{3_i} & \frac{x_{1_i} x_{2_i}}{f'} & -\frac{f'^2 + x_{1_i}^2}{f'} & x_{2_i} \\ 0 & -f' x_{3_i} & x_{2_i} x_{3_i} & \frac{f'^2 + x_{2_i}^2}{f'} & -\frac{x_{1_i} x_{2_i}}{f'} & -x_{1_i} \\ 0 & 0 & x_{3_i}^2 & \frac{x_{2_i} x_{3_i}}{f'} & -\frac{x_{1_i} x_{3_i}}{f'} & 0 \end{bmatrix} (V_c - V_p). \quad (2.23)$$

Also, the respective interaction matrix L_{e_i} can be reformulated as

$$L_{e_i} = \begin{bmatrix} -f' x_{3_i} & 0 & x_{1_i} x_{3_i} & \frac{x_{1_i} x_{2_i}}{f'} & -\frac{f'^2 + x_{1_i}^2}{f'} & x_{2_i} \\ 0 & -f' x_{3_i} & x_{2_i} x_{3_i} & \frac{f'^2 + x_{2_i}^2}{f'} & -\frac{x_{1_i} x_{2_i}}{f'} & -x_{1_i} \end{bmatrix}. \quad (2.24)$$

The depth z_i can be lower bounded by the camera focal length f and hence $0 < x_{3_i} \leq \frac{1}{f}$.

To calculate the upper bound on L'_e , utilising the relation given in (2.5) we can write

$$\|L'_e\| \leq \|L_{e_1}\|_1 + \|L_{e_2}\|_1 + \|L_{e_3}\|_1 + \|L_{e_4}\|_1. \quad (2.25)$$

From the expression of L_{e_i} given in (2.24), we can say that $\|L'_e\| \leq \rho'_1(\|x\|)$, where $x = [s_1, \frac{1}{z_1}, s_2, \frac{1}{z_2}, s_3, \frac{1}{z_3}, s_4, \frac{1}{z_4}]^T$ and $\rho'_1(\cdot)$ is a class κ -function.

Since $s^{*'} is constant, then from equation (2.6) we can write$

$$\|L'_e\| \leq \rho_e(\|e\|), \quad (2.26)$$

where $\rho_e(\cdot)$ is a monotonically increasing function.

The time derivative of L_{e_i} can be given as

$$\dot{L}'_{e_i} = \begin{bmatrix} -f' \dot{x}_{3_i} & 0 & x_{1_i} \dot{x}_{3_i} + x_{3_i} \dot{x}_{1_i} & \frac{\dot{x}_{1_i} x_{2_i}}{f'} + \frac{\dot{x}_{2_i} x_{1_i}}{f'} & -\frac{f'^2 + 2x_{1_i} \dot{x}_{1_i}}{f'} & \dot{x}_{2_i} \\ 0 & -f' \dot{x}_{3_i} & x_{2_i} \dot{x}_{3_i} + x_{3_i} \dot{x}_{2_i} & \frac{f'^2 + 2x_{2_i} \dot{x}_{2_i}}{f'} & -\frac{\dot{x}_{1_i} x_{2_i}}{f'} - \frac{\dot{x}_{2_i} x_{1_i}}{f'} & -\dot{x}_{1_i} \end{bmatrix}. \quad (2.27)$$

From the dynamics given in (2.23) with control law (2.18) and the fact that $\|\dot{L}'_e\| \leq \sum_{i=1}^4 \|\dot{L}_{e_i}\|$, we can state

$$\|\dot{L}'_e\| \leq \rho_1(\|e\|) - \lambda L_e^+ e - e_{V_p} \quad (2.28)$$

where $\rho_1(\cdot)$ is another monotonically increasing function.

Moreover, utilising Remark 1 and equation (2.27) we can conclude that there exist an upper bound for $L_e'^+$ such that $L_e'^+ \leq \rho_{e++}(\|e\|)$, where $\rho_{e+}(\cdot)$ is a monotonically increasing function. Thus

$$\|\dot{L}_e'\| \leq \rho_1(\|e\|)(\lambda \rho_{e+}(\|e\|)\|e\| + \|e_{v_p}\|). \quad (2.29)$$

□

Theorem 1. *For system (2.4) with velocity estimator (2.17) and controller (2.18), the closed-loop error dynamics of $e_c(t) = [e^T(t), e_v^T(t), e_{v_p}^T(t)]^T$ given in (2.21) is uniformly ultimately bounded (UUB), provided the following gain conditions are satisfied.*

$$\lambda > 1 \quad (2.30)$$

$$\frac{1}{\gamma_1} > 1 + \frac{1}{2\gamma_2^2} + \frac{1}{2}(h_1\bar{\rho}_1\lambda\bar{\rho}_{e+})^2 \quad (2.31)$$

$$\frac{1}{\gamma_2} > \frac{1}{2} + \frac{1}{2}\bar{\rho}_{e+}^2 + \frac{1}{2}(h_1\bar{\rho}_1)^2 + \frac{1}{2}\bar{\rho}_e^2 \quad (2.32)$$

where the constants $\bar{\rho}_1$, $\bar{\rho}_{e+}$, $\bar{\rho}_e$ are subsequently defined.

Proof. Consider a Lyapunov candidate for the error system given in (2.21) as

$$V = \frac{1}{2}e^T e + \frac{1}{2}e_v^T e_v + \frac{1}{2}e_{v_p}^T e_{v_p}. \quad (2.33)$$

Differentiating V and using equation (2.21), we get

$$\dot{V} = -\lambda e^T e - e^T L_e' e_{v_p} - \frac{1}{\gamma_1} e_v^T e_v + e_v^T \left(L_e' \dot{V}_p + \dot{L}_e' V_p \right) + e_{v_p}^T \left(\dot{V}_p - Proj(\hat{V}_p, \phi) \right). \quad (2.34)$$

Substituting the expression for projection operator from (2.17) we get

$$\begin{aligned}
\dot{V} &= -\lambda e^T e - e^T L'_e e_{v_p} - \frac{1}{\gamma_1} e_v^T e_v + e_v^T \left(L'_e \dot{V}_p + \dot{L}'_e V_p \right) \\
&\quad + e_{v_p}^T \left(\dot{V}_p - \frac{1}{\gamma_2} \left(L_e'^+ S - \hat{V}_p \pm V_p \right) + \phi_1 \right), \\
&= -\lambda e^T e - e^T L'_e e_{v_p} - \frac{1}{\gamma_1} e_v^T e_v + e_v^T \left(L'_e \dot{V}_p + \dot{L}'_e V_p \right) \\
&\quad + e_{v_p}^T \left(\dot{V}_p - \frac{1}{\gamma_2} \left(e_{v_p} - L_e'^+ e_v \right) + \phi_1 \right),
\end{aligned}$$

where $\phi_1 = \frac{\nabla f(\hat{V}_p)(\nabla f(\hat{V}_p))^T}{\|\nabla f(\hat{V}_p)\|^2} \phi f(\hat{V}_p)$ and will appear only when $f(\hat{V}_p) > 0$ and $\phi^T \nabla f(\hat{V}_p) > 0$. Moreover, $e_{v_p}^T \phi_1 \leq 0$ (Please refer to the properties of projection in Khalil (1996)).

Therefore, the bounds on $\dot{V}$ can be given as

$$\begin{aligned}
\dot{V} &\leq -\lambda \|e\|^2 - \frac{1}{\gamma_1} \|e_v\|^2 - \frac{1}{\gamma_2} \|e_{v_p}\|^2 + \|e\| \|L'_e\| \|e_{v_p}\| + \|e_v\| \left(\|L'_e\| \|\dot{V}_p\| + \|\dot{L}'_e\| \|V_p\| \right) \\
&\quad + \|e_{v_p}\| \|\dot{V}_p\| + \frac{1}{\gamma_2} \|e_{v_p}\| \|L_e'^+\| \|e_v\|.
\end{aligned} \tag{2.35}$$

Furthermore, using Lemma 2 and Assumption 2 will yield

$$\begin{aligned}
\dot{V} &\leq -\lambda \|e\|^2 - \frac{1}{\gamma_1} \|e_v\|^2 - \frac{1}{\gamma_2} \|e_{v_p}\|^2 + \rho_e(\|e\|) \|e\| \|e_{v_p}\| + h_2 \|e_{v_p}\| \\
&\quad + \|e_v\| \left(h_2 \rho_e(\|e\|) + h_1 \rho_1(\|e\|) (\lambda \rho_{e^+}(\|e\|) \|e\| + \|e_{v_p}\|) \right) + \frac{1}{\gamma_2} \rho_{e^+}(\|e\|) \|e_{v_p}\| \|e_v\|.
\end{aligned} \tag{2.36}$$

Consider a compact set $C = \{(e, e_v, e_{v_p}) \mid \|e\| < \bar{e}, \|e_v\| < \bar{e}_v, \|e_{v_p}\| < \bar{e}_{v_p}\}$ and within C the following holds

$$\rho_e(\|e\|) \leq \rho_e(\bar{e}) = \bar{\rho}_e, \quad \rho_{e^+}(\|e\|) \leq \rho_{e^+}(\bar{e}) = \bar{\rho}_{e^+}, \quad \rho_1(\|e\|) \leq \rho_1(\bar{e}) = \bar{\rho}_1. \tag{2.37}$$

Then, we have

$$\begin{aligned}
\dot{V} &\leq -\lambda \|e\|^2 - \frac{1}{\gamma_1} \|e_v\|^2 - \frac{1}{\gamma_2} \|e_{v_p}\|^2 + \bar{\rho}_e \|e\| \|e_{v_p}\| + \|e_v\| (h_2 \bar{\rho}_e + h_1 \bar{\rho}_1 (\lambda \bar{\rho}_{e^+} \|e\| + \|e_{v_p}\|)) \\
&\quad + h_2 \|e_{v_p}\| + \frac{1}{\gamma_2} \bar{\rho}_{e^+} \|e_{v_p}\| \|e_v\|.
\end{aligned} \tag{2.38}$$

Using Cauchy inequality and solving equation (2.38) will yield

$$\begin{aligned}\dot{V} \leq & -\lambda \|e\|^2 - \frac{1}{\gamma_1} \|e_v\|^2 - \frac{1}{\gamma_2} \|e_{v_p}\|^2 + \frac{1}{2} \|e\|^2 + \frac{1}{2} \bar{\rho}_e^2 \|e_{v_p}\|^2 + \frac{1}{2} (h_2 \bar{\rho}_e)^2 + \frac{1}{2} \|e_v\|^2 \\ & + \frac{1}{2} \|e\|^2 + \frac{1}{2} (h_1 \bar{\rho}_1 \lambda \bar{\rho}_{e+})^2 \|e_v\|^2 + \frac{1}{2} \|e_v\|^2 + \frac{1}{2} (h_1 \bar{\rho}_1)^2 \|e_{v_p}\|^2 + \frac{1}{2} h_2^2 \\ & + \frac{1}{2} \|e_{v_p}\|^2 + \frac{1}{2\gamma_2^2} \|e_v\|^2 + \frac{1}{2} \bar{\rho}_{e+}^2 \|e_{v_p}\|^2.\end{aligned}\tag{2.39}$$

Further solving equation (2.39) will yield

$$\begin{aligned}\dot{V} \leq & -(\lambda - 1) \|e\|^2 - \left(\frac{1}{\gamma_1} - 1 - \frac{1}{2\gamma_2^2} - \frac{1}{2} (h_1 \bar{\rho}_1 \lambda \bar{\rho}_{e+})^2 \right) \|e_v\|^2 \\ & - \left(\frac{1}{\gamma_2} - \frac{1}{2} - \frac{1}{2} \bar{\rho}_{e+}^2 - \frac{1}{2} (h_1 \bar{\rho}_1)^2 - \frac{1}{2} \bar{\rho}_e^2 \right) \|e_{v_p}\|^2 + \frac{1}{2} (h_2 \bar{\rho}_e)^2 + \frac{1}{2} h_2^2.\end{aligned}\tag{2.40}$$

Furthermore we can write

$$\dot{V} \leq -\bar{\lambda}_1 \|e\|^2 - \bar{\lambda}_2 \|e_v\|^2 - \bar{\lambda}_3 \|e_{v_p}\|^2 + D,\tag{2.41}$$

where $D = \frac{1}{2} (h_2 \bar{\rho}_e)^2 + \frac{1}{2} h_2^2$ and $\bar{\lambda}_1 = \lambda - 1$, $\bar{\lambda}_2 = \frac{1}{\gamma_1} - 1 - \frac{1}{2\gamma_2^2} - \frac{1}{2} (h_1 \bar{\rho}_1 \lambda \bar{\rho}_{e+})^2$, $\bar{\lambda}_3 = \frac{1}{\gamma_2} - \frac{1}{2} - \frac{1}{2} \bar{\rho}_{e+}^2 - \frac{1}{2} (h_1 \bar{\rho}_1)^2 - \frac{1}{2} \bar{\rho}_e^2$. Furthermore

$$\dot{V} \leq -\beta V + D,\tag{2.42}$$

where $\beta = 2 \times \min(\bar{\lambda}_1, \bar{\lambda}_2, \bar{\lambda}_3)$. The relation (2.42) invokes uniformly ultimately bounded (UUB) stability. However, the analysis is valid within the compact set C . In order to ensure that the solutions remain within the set C for all time, sufficient gain conditions must be satisfied. Choose $\bar{e}$, $\bar{e}_v$ and $\bar{e}_{v_p}$ such that the set $C_v = \{(e, e_v, e_{v_p}) \| V \leq \delta\}$, where δ is an upper-bound of $V(0)$, is a subset of the set C , i.e., $C_v \subset C$. Further, choose β such that $\frac{D}{\beta} \leq \delta$, which will ensure semi-global UUB stability. $\square$

The above theorem establishes UUB stability of the closed-loop error dynamics for time-varying target velocity. The case of constant target velocity is analysed in the subsequent theorem as a special case.

Theorem 2. *For system (2.4) with velocity estimator (2.17) and controller (2.18), the closed-loop error dynamics of $e_c(t) = [e^T(t), e_v^T(t), e_{v_p}^T(t)]^T$ given in (2.21) is asymptotically stable, provided the target is moving with constant velocity and the gain conditions*

given in Theorem 1 are satisfied.

Proof. When the target velocity is constant, equation (2.38) can be reformulated as

$$\begin{aligned}\dot{V} \leq & -\lambda \|e\|^2 - \frac{1}{\gamma_1} \|e_v\|^2 - \frac{1}{\gamma_2} \|e_{v_p}\|^2 \\ & + \bar{\rho}_e \|e\| \|e_{v_p}\| + h_1 \bar{\rho}_1 \|e_v\| (\lambda \bar{\rho}_{e^+} \|e\| + \|e_{v_p}\|) + \frac{1}{\gamma_2} \bar{\rho}_{e^+} \|e_{v_p}\| \|e_v\|.\end{aligned}\quad (2.43)$$

Solving equation (2.43) using Cauchy inequality will yield

$$\begin{aligned}\dot{V} \leq & -\lambda \|e\|^2 - \frac{1}{\gamma_1} \|e_v\|^2 - \frac{1}{\gamma_2} \|e_{v_p}\|^2 + \frac{1}{2} \|e\|^2 + \frac{1}{2} \bar{\rho}_e^2 \|e_{v_p}\|^2 + \frac{1}{2} \|e\|^2 + \frac{1}{2} \|e_v\|^2 \\ & + \frac{1}{2} (h_1 \bar{\rho}_1 \lambda \bar{\rho}_{e^+})^2 \|e_v\|^2 + \frac{1}{2} (h_1 \bar{\rho}_1)^2 \|e_{v_p}\|^2 + \frac{1}{2\gamma_2^2} \|e_v\|^2 + \frac{1}{2} \bar{\rho}_{e^+}^2 \|e_{v_p}\|^2.\end{aligned}\quad (2.44)$$

Furthermore we can write

$$\dot{V} \leq -\bar{\lambda}'_1 \|e\|^2 - \bar{\lambda}'_2 \|e_v\|^2 - \bar{\lambda}'_3 \|e_{v_p}\|^2, \quad (2.45)$$

where $\bar{\lambda}'_1 = \lambda - 1$, $\bar{\lambda}'_2 = \frac{1}{\gamma_1} - \frac{1}{2\gamma_2^2} - \frac{1}{2} - \frac{1}{2} (h_1 \bar{\rho}_1 \lambda \bar{\rho}_{e^+})^2$, $\bar{\lambda}'_3 = \frac{1}{\gamma_2} - \frac{1}{2} \bar{\rho}_e^2 - \frac{1}{2} \bar{\rho}_{e^+}^2 - \frac{1}{2} (h_1 \bar{\rho}_1)^2$ are positive based on (2.30)-(2.32). The inequality (2.45) invokes asymptotic stability, however, along the similar lines of the proof of Theorem 1 the above result is valid within the set C . Hence, similar choice of $\bar{e}$, $\bar{e}_v$ and $\bar{e}_{v_p}$ has to be maintained such that $C_v \subset C$ is ensured to establish semi-global asymptotic stability. $\square$

2.4 Results

We evaluate the performance of the proposed control and estimation scheme using simulations. An environment is created in MATLAB consisting of a camera and the target points (markers). A camera with focal length $15mm$, resolution 1024×1024 , $\rho_h = \rho_w = 10^{-5}m/pixel$ and principal point $u_0 = v_0 = 512$ is considered. Four co-planar, non-concyclic markers are considered with initial positions $P_1 = [-0.25, -0.25, 3]^T m$, $P_2 = [0.25, -0.25, 3]^T m$, $P_3 = [0.36, 0.25, 3]^T m$, $P_4 = [-0.25, 0.25, 3]^T m$, expressed in the global frame. The initial position of the camera is considered as $[1, 1, 0]^T m$ with the angular orientation (RPY) $[0.3, -0.1, 0.6]^T rad$, also expressed in the world frame, which demands desired rotation along all the camera axis. To generate the desired image

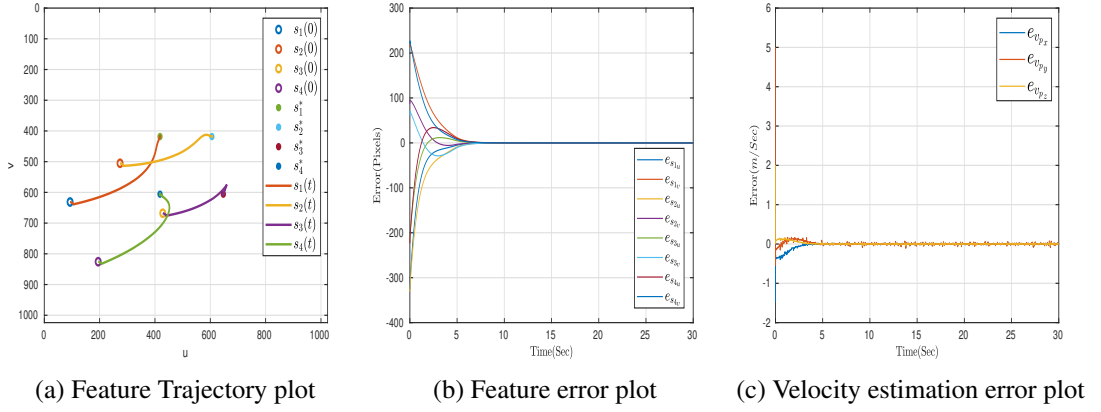


Figure 2.2: Simulation results for constant target velocity

plane feature coordinates, the desired camera position is chosen such that the camera's X and Y coordinates align to the markers' centroid while maintaining a depth of 4 meters. The controller gain λ is set to 1.1, and the filter variables are chosen as $\gamma_1 = 10^{-4}$, $\gamma_2 = 10^{-2.5}$. The known upper-bound of the markers' velocity is taken to be $20m/sec$ with the tolerance $\varepsilon = 0.1$. To make the simulations more realistic we have also considered the actuator constraints for the camera motion such that it's individual spatial linear and the angular velocities can't exceeds more than $\pm 10m/sec$ and $\pm \frac{\pi}{10} rad/sec$. In the real-world scenarios it may not be possible to get the exact intrinsic parameters for a camera as well as the depth. To evaluate the proposed scheme for near real-world scenarios we have assumed 5% camera calibration error in the camera focal length and the feature depths are corrupted by adding a Gaussian noise with 0 mean and 0.01 variance. The results for constant and time-varying target motions are presented in the subsequent sections.

2.4.1 Target Moving with Constant Velocity

The simulation results for a scenario where target is moving with a constant linear velocity $v_p = [4, 5, 2]^T m/sec$ are presented in this section. Fig. 2.2a, 2.2b, and 2.2c shows the feature trajectory, feature error and target velocity estimation error respectively. It can be seen clearly that the features errors are converging to zero along with the velocity estimation error despite the noisy depth measurements.

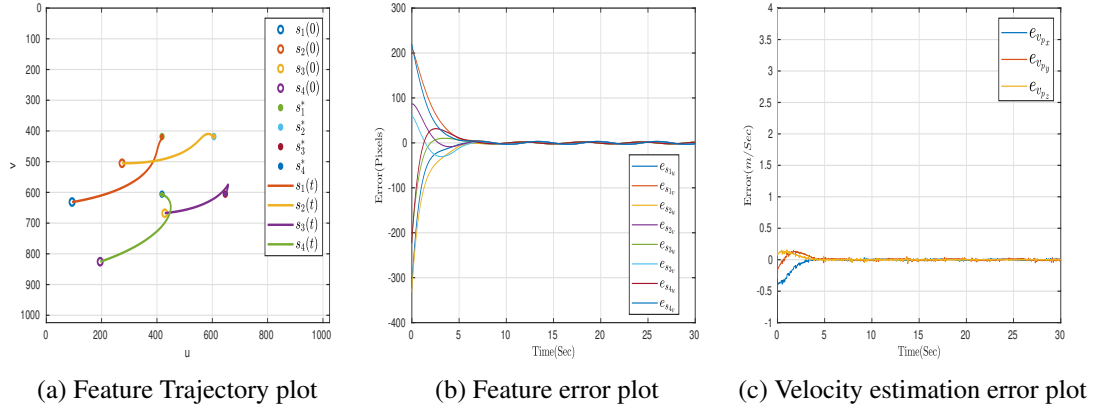


Figure 2.3: Simulation results for time-varying target velocity

2.4.2 Target Moving with Time-Varying Velocity

The simulation results for a scenario where the target is moving with a time-varying linear velocity $v_p = [2\cos(t), 3\sin(t), 4e^{-0.05t}]^T m/sec$ are presented in this section. Fig. 2.3a, 2.3b, and 2.3c show the feature trajectory, feature error, and target velocity estimation error, respectively. It can be seen clearly that the feature errors are converging to an ultimate bound within zero along with the velocity estimation error despite the noise depth measurements. In fact, the feature error remains within the bound of ± 10 pixels.

2.5 Conclusion

In this chapter, we presented an IBVS scheme for tracking moving targets with arbitrary translational motion in a 3D plane. A novel projection-based target velocity estimator, inspired by dynamic surface filters, was developed to enhance tracking performance. The proposed control law, integrated with the velocity estimator, ensures convergence of tracking and estimation errors to an ultimate bound for time-varying target velocities and to zero for constant target velocities. Simulations further demonstrate the algorithm's robustness against disturbances and inaccuracies in depth information and camera parameters, highlighting its reliability for real-world applications.

Chapter 3

Dynamically Feasible IBVS for Quadrotor UAVs Tracking Arbitrary Moving Planar Targets

This chapter extends the IBVS framework developed in Chapter 2 to handle unknown yaw motion of the target while ensuring compatibility with the underactuated nature of quadrotors, which can track only four degrees of freedom—three translational and one rotational (yaw). The presence of target angular motion introduces significant challenges, particularly due to the coupled Jacobian structure in the original point feature-based formulation.

This chapter presents a generalized IBVS scheme applied to under-actuated quadrotor kinematics to track a planar target that exhibits arbitrary time-varying translational velocity and angular velocity along the z-axis. To ensure dynamic feasibility of the control laws, we employ the virtual image plane method using a transformed representation of the real image plane in which the image point feature dynamics become invariant to the quadrotor’s roll and pitch angles. We discuss two methods to construct a virtual image plane considering different camera attachments. The first method uses a pan-tilt camera, while the second, for a fixed camera, utilizes the quadrotor’s orientation information for the virtual transformation. Using measurements from the virtual plane, we define image moment-based features that allow for the design of decoupled and dynamically feasible 4-DOF references for translational and yaw motion.

We propose kinematic controllers that incorporate a smooth target motion estimator. These differential estimators can effectively approximate bounded target velocity profiles model-free. Our method eliminates the need to introduce a feedback domination term in the control law to enhance robustness. Instead, it directly compensates for the unknown target motion through the proposed target velocity estimation feedback. Assuming the existence of an inner-loop controller similar to Bourquardez *et al.* (2009), we establish the stability and convergence properties of the outer-loop IBVS controller. We also consider the mismatch in reference tracking due to the inner-loop controller and provide a theoretical analysis of its effect on the error dynamics. We show that the

ultimate bound due to the time-varying target velocity can be made arbitrarily small by choosing filter gains appropriately without burdening the controller gains. The effectiveness of the proposed framework is demonstrated through comprehensive MATLAB simulations, Software-In-The-Loop (SITL) simulations using a fixed camera setup, and hardware experiments conducted in an outdoor environment. The hardware setup includes a quadrotor equipped with a pan-tilt camera and a ground rover tagged with an ArUco marker, successfully validating the approach through vision-based moving target tracking.

Since the problem formulation boils down to a tracking problem with time-varying uncertainties, it is worthwhile to compare the proposed scheme with state-of-the-art sliding mode techniques. Sliding mode control is one of the most promising robust control techniques with superior theoretical guarantees to completely reject the model uncertainties and external disturbances altogether Shtessel *et al.* (2014). However, the chattering effect makes it less attractive for practical applications, as it leads to undesired oscillations in the system's output and control signal, possibly damaging the system. Besides the obvious and unavoidable reasons for chattering, such as actuator dynamics and time delays, there have been multiple developments related to the chattering of the discrete implementation of these sliding mode methods Koch and Reichhartinger (2019). Owing to these recent developments, we compare our proposed method with the state-of-the-art Super-Twisting (ST) algorithm Koch *et al.* (2019). The results show that the proposed scheme yields superior performance when implemented on the real system while maintaining comparable theoretical guarantees.

3.1 Problem Formulation

3.1.1 Virtual Image Plane

Let $\{I\}$, $\{b\}$ and $\{c\}$ be inertial, the quadrotor body frames and the camera frame, with origins O_I and O_b , respectively, as illustrated in Figure 3.1. Next, we create a virtual image plane that is invariant to quadrotor roll (ϕ) and pitch (θ) angles to design a dynamically feasible control law. For a fixed camera, we do this by transforming the obtained real image plane coordinates to a virtual image plane obtained through a

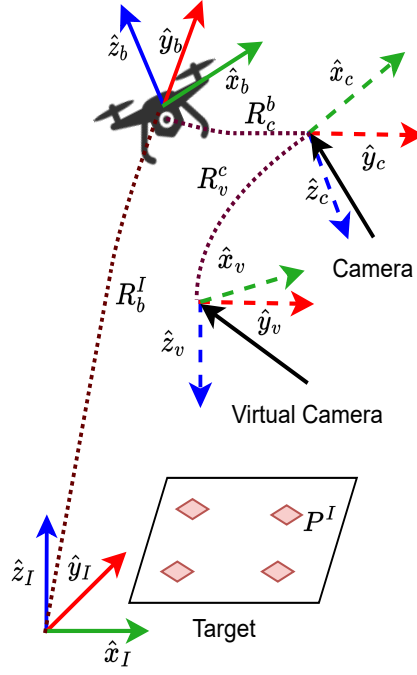


Figure 3.1: Coordinate frames illustration. A coordinate frame $\{A\}$ is represent by axes $\hat{x}_A$, $\hat{y}_A$ and $\hat{z}_A$.

virtual camera (frame $\{v\}$) with an origin similar to that of the camera frame Li *et al.* (2021a). Another method is to use a pan-tilt camera where the gimbal ensures that the camera is always downward-looking, irrespective of the drone maneuvers, creating the desired virtual image plane. Here, the image measurements from the pan-tilt camera can be used directly as virtual image plane measurements.

For the fixed camera case, the virtual image plane coordinates of a 3-D point P^I , expressed in the inertial frame with its image plane point feature coordinates $s = [u, v]^T$, is reconstructed as follows

$$s_v = \begin{bmatrix} u_v \\ v_v \end{bmatrix} = \frac{f'}{\bar{R}_3 \bar{p}} \begin{bmatrix} \bar{R}_1 \bar{p} \\ \bar{R}_2 \bar{p} \end{bmatrix}, \quad (3.1)$$

where $\bar{p} = [u - c_x, v - c_y, f']^T$ and $\bar{R}_i, i = 1, 2, 3$ denotes the i^{th} row of the rotation matrix $\bar{R} = R_c^{bT} R_y(\theta) R_x(\phi) R_c^b$ which represents the rotation from $\{c\}$ to $\{v\}$. (c_x, c_y) is the principal point offset of the camera and f' is the focal length of the camera in pixels. The matrix R_c^b represents the rotation from $\{c\}$ to $\{b\}$, which remains constant as the camera has a fixed pose with respect to the quadrotor body frame. One can compute this matrix by analyzing the fixed camera attachment with the quadrotor body frame.

3.1.2 Image Features & Dynamics

We define the visual features and their respective dynamics in this section. Consider a planar target with N observable visual points $s_{v_k} = [u_{v_k}, v_{v_k}]^T$, $k = 1, 2, \dots, N$, we define an image features set $s'(t) = [x_n(t), y_n(t), a_n(t)] \in \mathbb{R}^3$ for translational kinematics of quadrotor as

$$x_n = \frac{a_n}{f} x_g, \quad y_n = \frac{a_n}{f} y_g, \quad a_n = h_d \sqrt{\frac{a_d}{a}}, \quad (3.2)$$

where $[x_g, y_g]^T = [\frac{1}{N} \sum_{k=1}^N u_{v_k}, \frac{1}{N} \sum_{k=1}^N v_{v_k}]^T$ and $a = \sum_{k=1}^N ((u_{v_k} - x_g)^2 + (v_{v_k} - y_g)^2)$. a_d is the desired value of a , and h_d is the desired height of the virtual camera frame from the planar target. An illustration of these features is succinctly depicted in Figure 3.2.

For a target, moving with translational velocity $\bar{v}_p(t) \in \mathbb{R}^3$ and angular rate $[0, 0, \dot{\psi}_p(t)]^T \in \mathbb{R}^3$, expressed in $\{v\}$, the translational feature dynamics is given as

$$\dot{s}' = -(\bar{v} - \bar{v}_p) - S_k([0, 0, \dot{\psi}]^T) s', \quad (3.3)$$

where $[0, 0, \dot{\psi}(t)]^T \in \mathbb{R}^3$ and $\bar{v}(t) \in \mathbb{R}^3$, are the angular rate and the linear velocity of the virtual camera $\{v\}$, expressed in the $\{v\}$.

We next define a scalar feature $\alpha(t)$ representing the orientation of an object in the image plane for the heading controller of the quadrotor as

$$\alpha = \frac{1}{2} \arctan \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right), \quad (3.4)$$

where $\mu_{ij} = \sum_{k=1}^N ((u_{v_k} - x_g)^i (v_{v_k} - y_g)^j)$ is the centered moment of order $i + j$. It is worth noting that the feature $\alpha(t)$ in (3.4) is defined based on the elliptical primitive, which becomes degenerate if the target is not skewed (for example, a square) Chaumette (2004). In such cases, a rather simpler alternative is to use the slope angle of the line that joins any two visual points in the image plane. The feature $\alpha(t)$ has the dynamics

$$\dot{\alpha} = -\dot{\psi} + \dot{\psi}_p, \quad (3.5)$$

where $\dot{\psi}_p(t)$ is the angular rate of the target expressed in $\{v\}$.

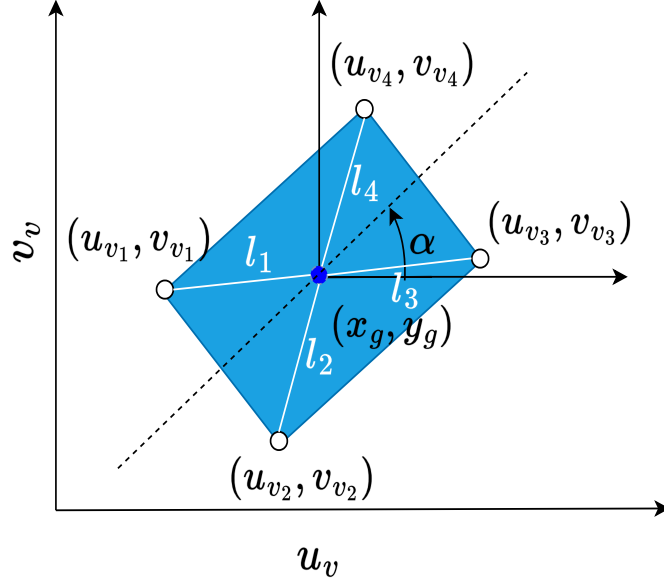


Figure 3.2: Virtual features illustration for a target with four visual points features (i.e., $N = 4$) represented by $\circ$ in the virtual plane. The point (x_g, y_g) represents the centroid of the target in the virtual plane. a in the a_n represents the sum of the squared distances (i.e. $\sum_{k=1}^4 l_k^2$) of each point feature from the centroid. The scalar feature $\alpha(t)$ is calculated using the centered image moments of order two as given in equation (3.4).

3.1.3 Control Objective

The control objective is defined such that the quadrotor must keep the target in the center of the virtual image plane while maintaining a user-defined desired heading and a height h_d w.r.t. the target. This corresponds to $s'^* = [0, 0, h_d]$ as a desired of s' . The angular feature α^* depends on the geometry of the convex hull made by feature points in the virtual image plane. One can calculate the desired α^* based on the given desired feature points (obtained using the desired image) in the virtual image plane using the relation (3.4). We define the feature-tracking errors as

$$e_s(t) = s'(t) - s'^*, \quad (3.6)$$

$$e_\alpha(t) = \alpha(t) - \alpha^*. \quad (3.7)$$

For the aforementioned choice of features, the feature tracking error dynamics becomes

$$\dot{e}_s = -(\bar{v} - \bar{v}_p) - S_k([0, 0, \dot{\psi}]^T) e_s, \quad (3.8)$$

$$\dot{e}_\alpha = -(\dot{\psi} - \dot{\psi}_p). \quad (3.9)$$

The control inputs in dynamics (3.8) and (3.9) are $\bar{v}$ and $\dot{\bar{\psi}}$. However, the accuracy with which the quadrotor follows these commands depends on the performance of the available inner-loop controller. To incorporate and analyze this phenomenon, we define the control inputs in the following form

$$\bar{v}(t) = \bar{v}_d(t) + d_v(t), \quad (3.10)$$

$$\dot{\bar{\psi}}(t) = \dot{\bar{\psi}}_d(t) + d_\psi(t), \quad (3.11)$$

where $\bar{v}_d(t)$ and $\dot{\bar{\psi}}_d(t)$ are the desired translational and angular velocity commands to be designed as part of our outer-loop controller such that the control objective is achieved. The quantities $d_v(t)$ and $d_\psi(t)$ are the mismatch between the designed command velocities ($\bar{v}_d(t)$ and $\dot{\bar{\psi}}_d(t)$) and the actual quadrotor velocities ($\bar{v}(t)$ and $\dot{\bar{\psi}}(t)$) available from measurements using onboard sensors. The exact profile of the mismatch disturbances ($d_v(t)$ and $d_\psi(t)$) depends on the chosen inner-loop controller, which is not the focus of the current work. However, the following assumption is considered regarding the mismatch disturbances, which the state-of-the-art inner-loop controllers can ensure.

Assumption 3. *The norm of the disturbances has a finite upper bound, i.e.,*

$$\|d_v(t)\| \leq \bar{d}_v \quad (3.12)$$

$$\|d_\psi(t)\| \leq \bar{d}_\psi \quad (3.13)$$

where $\bar{d}_v$ and $\bar{d}_\psi$ are finite positive constants.

3.1.4 Quadrotor Kinematic Control

Since the virtual camera motion is controlled by quadrotor kinematics, it is essential to express the designed control inputs (expressed in $\{v\}$) in terms of desired kinematic velocities. Considering the displacement between $\{b\}$ and $\{c\}$ (or $\{v\}$) is small, the control input $\bar{v}_d(t)$ can be converted to the desired linear velocity $v_d(t)$ of frame $\{b\}$ and expressed in $\{I\}$ as

$$v_d = R_z(\psi) R_c^b \bar{v}_d, \quad (3.14)$$

where $R_z(\psi)R_c^b$ is the rotation matrix from frame $\{v\}$ to $\{I\}$ (i.e. R_v^I). Similarly, based on the definition of the virtual image plane, we have $R_z(\bar{\psi}) = R_z(\psi)R_c^b$. Thus, an explicit relation can be obtained between $\psi(t)$ (yaw angle of the quadrotor body frame with respect to the body fixed inertial frame) and $\bar{\psi}(t)$ (yaw angle of the virtual frame with respect to the fixed inertial frame) which shall later be used to generate the desired yaw rate for the quadrotor based on $\dot{\bar{\psi}}_d(t)$.

Assumption 4. *The target translational velocity $v_p(t)$ and angular velocity $\dot{\psi}_p(t)$ along the z -axis, both expressed in the inertial frame, are bounded such that*

$$\sup_{t \geq 0} \|v_p(t)\| = h_1, \quad \sup_{t \geq 0} \|\dot{v}_p(t)\| = h_2, \quad (3.15)$$

$$\sup_{t \geq 0} \|\dot{\psi}_p(t)\| = h_3, \quad \sup_{t \geq 0} \|\ddot{\psi}_p(t)\| = h_4, \quad (3.16)$$

where $h_1, h_2, h_3, h_4 \in \mathbb{R}_+$ are finite constants.

Based on assumption 4 and the fact that matrix R_b^c is constant, we can derive the following upper bounds on the target motion expressed in $\{v\}$

$$\sup_{t \geq 0} \|\bar{v}_p(t)\| = \sup_{t \geq 0} \|R_c^{bT} R_z(\psi)^T v_p(t)\| \leq h_1. \quad (3.17)$$

$$\sup_{t \geq 0} \|\ddot{\bar{\psi}}_p(t)\| \leq \|\ddot{\psi}_p(t)\| \leq h_4. \quad (3.18)$$

3.2 Control Design

In our approach, we utilize a control architecture as depicted in Figure 3.3. Our first step involves creating a virtual image plane, from which we derive image moment-based features using equations (3.2) and (3.4). These features are used to calculate tracking errors and design target velocity estimators for both translational and angular velocity. The next step is to design desired translational and angular velocity commands ($\bar{v}_d(t)$ and $\dot{\bar{\psi}}_d(t)$) for the quadrotor to be tracked by the inner loop, thereby completing the architecture. The details of the velocity estimators and the control law design are presented next.

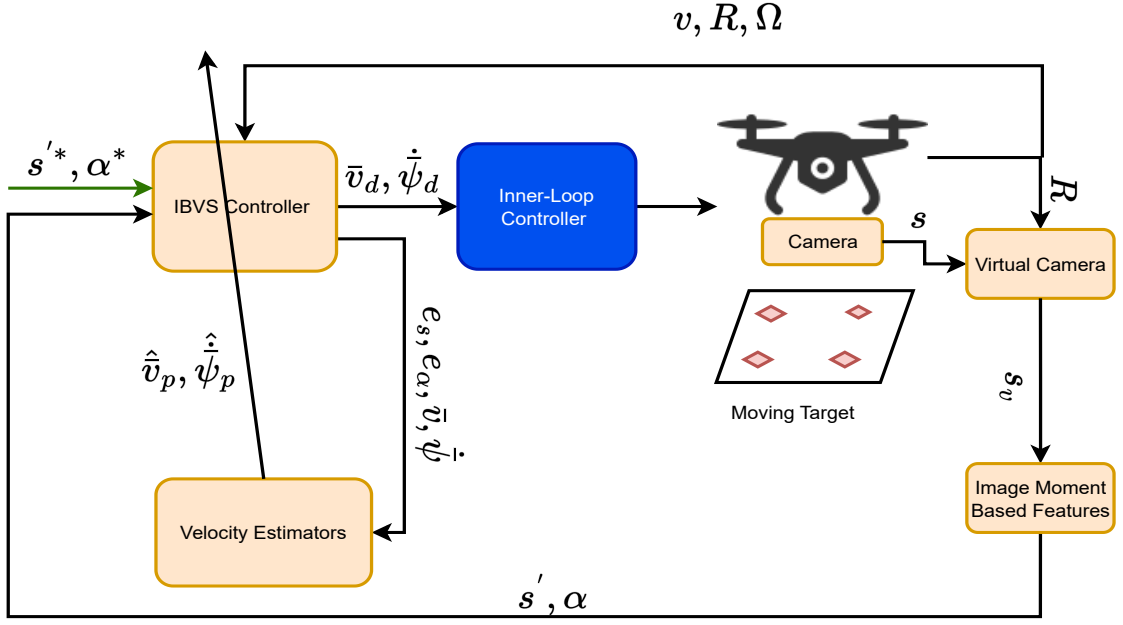


Figure 3.3: Control architecture

3.2.1 Translational Velocity Estimator

We leverage error dynamics (3.8) to design the translational target velocity considering filter dynamics with design parameter $\gamma_1 \in \mathbb{R}_+$ for the measurable signals $e_s(t)$, $\dot{\psi}(t)$ and $\bar{v}(t)$ as follows

$$\gamma_1 \dot{e}_{fs} + e_{fs} = e_s, \quad e_{fs}(0) = 0, \quad (3.19)$$

$$\gamma_1 \dot{h}_s + h_s = S_k([0, 0, \dot{\psi}]^T) e_s + \bar{v}, \quad h_s(0) = 0, \quad (3.20)$$

with filtered outputs $e_{fs}(t) \in \mathbb{R}^3$ and $h_s(t) \in \mathbb{R}^3$. The following lemma provides the solution for the corresponding filtered signal of the unknown signal $\bar{v}_p(t)$, which is used for the velocity estimator design.

Lemma 3. Let $S_s(t) \in \mathbb{R}^3$ be a filtered variable corresponding to unknown variable $\bar{v}_p(t)$ with the following filter dynamics

$$\dot{S}_s = \frac{1}{\gamma_1} (\bar{v}_p - S_s), \quad S_s(0) = 0. \quad (3.21)$$

Then, we have

$$S_s(t) = h_s(t) + \frac{1}{\gamma_1} \left(e_s(t) - e_{fs}(t) - e^{(-\frac{1}{\gamma_1}t)} e_s(0) \right). \quad (3.22)$$

Proof. Let's consider another filter with filter variable $g_s(t) \in \mathbb{R}^3$ as

$$\gamma_1 \dot{g}_s + g_s = \dot{e}_s, \quad g_s(0) = 0. \quad (3.23)$$

We can write the solution $g_s(t)$ of above equation as

$$g_s(t) = \frac{1}{\gamma_1} \int_0^t \exp\left(-\frac{1}{\gamma_1}(t-\tau)\right) \dot{e}_s(\tau) d\tau. \quad (3.24)$$

Further solving (3.24) will yield

$$g_s(t) = \frac{1}{\gamma_1} \left(e_s(t) - e_{fs}(t) - \exp\left(-\frac{1}{\gamma_1}t\right) e_s(0) \right). \quad (3.25)$$

Furthermore using the error dynamics (3.8) in equation (3.24)

$$\begin{aligned} g_s(t) = & \frac{1}{\gamma_1} \int_0^t \exp\left(-\frac{1}{\gamma_1}(t-\tau)\right) (-S_k([0, 0, \dot{\psi}(\tau)]^T) e_s(\tau)) \\ & - \bar{v}(\tau)) d\tau + \frac{1}{\gamma_1} \int_0^t \exp\left(-\frac{1}{\gamma_1}(t-\tau)\right) (\bar{v}_p(\tau)) d\tau. \end{aligned} \quad (3.26)$$

The relation (3.22) can be obtained using (3.25) and (3.26). $\square$

We design the following translational velocity estimator for the target

$$\hat{\bar{v}}_p(t) = S_s(t). \quad (3.27)$$

3.2.2 Angular Velocity Estimator

Similarly, we define another filter dynamics for the error dynamics (3.9) with measurable signals $e_\alpha(t)$, $\dot{\psi}(t)$, and design parameter $\gamma_2 \in \mathbb{R}_+$ as

$$\gamma_2 \dot{e}_{f\psi} + e_{f\psi} = e_\alpha, \quad e_{f\psi}(0) = 0, \quad (3.28)$$

$$\gamma_2 \dot{h}_\psi + h_\psi = \dot{\psi}, \quad h_\psi(0) = 0, \quad (3.29)$$

with filtered output $e_{f\psi}(t)$ and $h_\psi(t)$. Now, we define the filtered signal for unknown signal $\dot{\psi}_p(t)$ in terms of measurable signals and their filtered output.

Lemma 4. Consider a filtered variable $S_\psi(t)$ corresponding to unknown variable $\dot{\psi}_p(t)$ as

$$\dot{S}_\psi = \frac{1}{\gamma_2}(\dot{\psi}_p - S_\psi), \quad S_\psi(0) = 0. \quad (3.30)$$

Then, we have

$$S_\psi(t) = h_\psi(t) + \frac{1}{\gamma_2} \left(e_\alpha(t) - e_{f\psi}(t) - e^{\left(-\frac{1}{\gamma_2}t\right)} e_\alpha(0) \right). \quad (3.31)$$

Proof. Let's consider a filter for unknown variable $\dot{e}_\alpha$ with filter output variable $g_\alpha(t)$ as

$$\gamma_2 \dot{g}_\alpha + g_\alpha = \dot{e}_\alpha, \quad g_\alpha(0) = 0. \quad (3.32)$$

Solving (3.32) for $g_\alpha(t)$ yields

$$g_\alpha(t) = \frac{1}{\gamma_2} \int_0^t \exp\left(-\frac{1}{\gamma_2}(t-\tau)\right) \dot{e}_\alpha(\tau) d\tau. \quad (3.33)$$

Further solving (3.33) will yield

$$g_\alpha(t) = \frac{1}{\gamma_2} \left(e_\alpha(t) - e_{f\psi}(t) - \exp\left(-\frac{1}{\gamma_2}t\right) e_\alpha(0) \right). \quad (3.34)$$

Furthermore, using the error dynamics (3.9) in equation (3.32)

$$\begin{aligned} g_\alpha(t) = & -\frac{1}{\gamma_2} \int_0^t \exp\left(-\frac{1}{\gamma_2}(t-\tau)\right) \dot{\psi}(\tau) d\tau \\ & + \frac{1}{\gamma_2} \int_0^t \exp\left(-\frac{1}{\gamma_2}(t-\tau)\right) \dot{\psi}_p(\tau) d\tau. \end{aligned} \quad (3.35)$$

The relation (3.31) can be obtained using (3.34) and (3.35). $\square$

Equipped with Lemma 4, we propose the following velocity estimator for the angular velocity of the target expressed in $\{v\}$

$$\hat{\dot{\psi}}_p(t) = S_\psi(t). \quad (3.36)$$

Remark 2. The translational velocity estimator proposed in section 3.2.1 can be straightforwardly generalized for the cases where the requirements are limited to translational motion control of the quadrotor Jabbari Asl (2019); Lin et al. (2022).

3.2.3 Control Law

We design the IBVS kinematic control laws for quadrotor as follows

$$\bar{v}_d(t) = k_1 e_s(t) + \hat{\bar{v}}_p(t), \quad (3.37)$$

$$\dot{\bar{\psi}}_d(t) = k_2 e_\alpha(t) + \hat{\dot{\bar{\psi}}}_p(t), \quad (3.38)$$

where $k_1 \in \mathbb{R}_+$ and $k_2 \in \mathbb{R}_+$ are positive control gains. $\hat{\bar{v}}_p(t)$ and $\hat{\dot{\bar{\psi}}}_p(t)$ are the target velocity estimates given in (3.27) and (3.36), respectively.

3.2.4 Stability and Convergence Proofs

To analyze the error convergence property of tracking and estimation errors, we first derive the expression for their closed-loop dynamics. We define the velocity estimation error $e_{\bar{v}_p}(t), e_{\bar{\psi}_p}(t)$ as

$$e_{\bar{v}_p} = \bar{v}_p - \hat{\bar{v}}_p, \quad (3.39)$$

$$e_{\bar{\psi}_p} = \dot{\bar{\psi}}_p - \hat{\dot{\bar{\psi}}}_p. \quad (3.40)$$

Using the control and estimation laws (3.37), (3.38), (3.27) and (3.36), the closed-loop error dynamics for the extended error system $e'(t) = [e_s(t), e_\alpha(t), e_{\bar{v}_p}(t), e_{\bar{\psi}_p}(t)]$ becomes

$$\begin{bmatrix} \dot{e}_s \\ \dot{e}_\alpha \\ \dot{e}_{\bar{v}_p} \\ \dot{e}_{\bar{\psi}_p} \end{bmatrix} = \begin{bmatrix} -k_1 e_s - S_k([0, 0, \dot{\bar{\psi}}]^T) e_s + e_{\bar{v}_p} - d_v \\ -k_2 e_\alpha + e_{\bar{\psi}_p} - d_\psi \\ -\frac{1}{\gamma_1} e_{\bar{v}_p} + \dot{\bar{v}}_p \\ -\frac{1}{\gamma_2} e_{\bar{\psi}_p} + \ddot{\bar{\psi}}_p \end{bmatrix}. \quad (3.41)$$

We now show that the proposed controller keeps the errors bounded despite the arbitrary motion of the target and inner-loop controller mismatch.

Theorem 3. The closed-loop error dynamics of $e'(t) = [e_s(t), e_\alpha(t), e_{\bar{v}_p}(t), e_{\bar{\psi}_p}(t)]$ given

in (3.41) is uniformly ultimately bounded (UUB) with the proposed outer-loop control inputs given in (3.37) and (3.38) despite the inner-loop mismatch, provided the following gain conditions are satisfied

$$k_1 > 1, \quad k_2 > \frac{3}{2}, \quad \frac{1}{\gamma_1} > 4h_1^2(k_2^2 + 1) + 2, \quad \frac{1}{\gamma_2} > 2. \quad (3.42)$$

Proof. Consider the following Lyapunov candidate as

$$V_s = \frac{1}{2}e_s^T e_s + \frac{1}{2}e_{\bar{v}_p}^T e_{\bar{v}_p} + \frac{1}{2}e_\alpha^2 + \frac{1}{2}e_{\psi_p}^2. \quad (3.43)$$

Taking the time derivative of $V_s(t)$ using equations (3.41), (3.27) and (3.36), yields

$$\begin{aligned} \dot{V}_s = & -k_1 e_s^T e_s + e_s^T e_{\bar{v}_p} - e_s^T d_v - \frac{1}{\gamma_1} e_{\bar{v}_p}^T e_{\bar{v}_p} + e_{\bar{v}_p}^T \dot{\bar{v}}_p \\ & - k_2 e_\alpha^2 + e_\alpha e_{\psi_p} - \frac{1}{\gamma_2} e_{\psi_p}^2 + e_{\psi_p} \ddot{\psi}_p - e_\alpha d_\psi \end{aligned} \quad (3.44)$$

Further, taking the bounds on $\dot{V}_s(t)$, we have

$$\begin{aligned} \dot{V}_s \leq & -k_1 \|e_s\|^2 - \frac{1}{\gamma_1} \|e_{\bar{v}_p}\|^2 + \|e_s\| \|e_{\bar{v}_p}\| + \|e_{\bar{v}_p}\| \|\dot{\bar{v}}_p\| + \|e_{\psi_p}\| \|\ddot{\psi}_p\| \\ & + \|e_s\| \|d_v\| - k_2 \|e_\alpha\|^2 - \frac{1}{\gamma_2} \|e_{\psi_p}\|^2 + \|e_\alpha\| \|e_{\psi_p}\| + \|e_\alpha\| \|d_\psi\| \end{aligned} \quad (3.45)$$

Before proceeding further, we calculate the upper bounds of $\dot{\bar{v}}_p(t)$ using the fact that

$$\dot{\bar{v}}_p(t) = -S_k([0, 0, \dot{\psi}]^T) R_z(\bar{\psi})^T v_p(t) + R_z(\bar{\psi})^T \dot{v}_p(t). \quad (3.46)$$

The equation (3.46) with the control law (3.38) results in

$$\|\dot{\bar{v}}_p(t)\| \leq (2k_2 \|e_\alpha\| + 2\|e_{\psi_p}\| + 2h_3 + 2\|d_\psi\|)h_1 + h_2. \quad (3.47)$$

Substituting the above upper bound and using Assumption 3, 4 in (3.45), we get the

following

$$\begin{aligned}
\dot{V}_s \leq & -k_1 \|e_s\|^2 - \frac{1}{\gamma_1} \|e_{\bar{v}_p}\|^2 + \|e_s\| \|e_{\bar{v}_p}\| + 2k_2 h_1 \|e_{\bar{v}_p}\| \|e_\alpha\| + \|e_\alpha\| \|e_{\bar{\psi}_p}\| \\
& + 2h_1 \|e_{\bar{v}_p}\| \|e_{\bar{\psi}_p}\| + \|e_s\| \bar{d}_v + 2h_1 \|e_{\bar{v}_p}\| \bar{d}_\psi + h_4 \|e_{\bar{\psi}_p}\| \\
& + 2h_3 h_1 \|e_{\bar{v}_p}\| + h_2 \|e_{\bar{v}_p}\| - k_2 \|e_\alpha\|^2 - \frac{1}{\gamma_2} \|e_{\bar{\psi}_p}\|^2 + \|e_\alpha\| \bar{d}_\psi. \tag{3.48}
\end{aligned}$$

Further solving (3.48) using Cauchy inequalities yields

$$\begin{aligned}
\dot{V}_s \leq & - \underbrace{[k_1 - 1]}_{\lambda_1} \|e_s\|^2 - \underbrace{\left[\frac{1}{2\gamma_1} - 2h_1^2(k_2^2 + 1) - 1 \right]}_{\lambda_2} \|e_{\bar{v}_p}\|^2 - \underbrace{\left[k_2 - \frac{3}{2} \right]}_{\lambda_3} \|e_\alpha\|^2 \\
& - \underbrace{\left[\frac{1}{2\gamma_2} - 1 \right]}_{\lambda_4} \|e_{\bar{\psi}_p}\|^2 + \underbrace{4\gamma_1 h_1^2 h_3^2 + \gamma_1 h_2^2 + \frac{\gamma_2}{2} h_4^2}_{D_1} + \underbrace{\frac{1}{2} (\bar{d}_v^2 + \bar{d}_\psi^2 (4h_1^2 + 1))}_{D_2}. \tag{3.49}
\end{aligned}$$

Equation (3.49) implies

$$\dot{V}_s \leq -\beta V_s + D, \tag{3.50}$$

where $D = D_1 + D_2$ and $\beta = 2 \times \min(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$. The relation (3.50) invokes global uniformly ultimately bounded (UUB) stability (based on Theorem 4.18 in Khalil (2002)). $\square$

Remark 3. Note that the ultimate bound $\left(\frac{D}{\beta}\right)$ consists of two components: D_1 and D_2 . The term D_1 captures the contribution due to the unknown target motion and can be made arbitrarily small by appropriately selecting the design parameters γ_1 and γ_2 . On the other hand, D_2 arises from the inner-loop mismatch and depends nonlinearly on the control gains k_1, k_2 . Due to this nonlinear dependence, it is not straightforward to predict or minimize the magnitude of the term $\left(\frac{D_2}{\beta}\right)$ directly through tuning β as this might result in different D_2 .

Next, we present the corollary to analyze the stability and convergence of the extended closed loop error dynamics for pure constant translational motion of target (i.e., $h_2 = h_3 = h_4 = 0$ in Assumption 4) with $\lim_{t \rightarrow \infty} d_v(t), d_\psi(t) \rightarrow 0$ as a special case.

Corollary 3.1. The error system $e'(t) = [e_s(t), e_\alpha(t), e_{\bar{v}_p}(t), e_{\bar{\psi}_p}(t)]$ given in (3.41) asymptotically converges to zero when the target is moving with pure constant translational

motion and the vanishing inner-loop controller mismatch, provided the following gain conditions are satisfied.

$$k_1 > 1, \quad k_2 > \frac{3}{2}, \quad \frac{1}{\gamma_1} > 2k_2^2 h_1^2 + 1, \quad \frac{1}{\gamma_2} > \frac{1}{2}. \quad (3.51)$$

Proof. We use a similar Lyapunov candidate to that given in (3.43) for the proof of this corollary. For the sake of compactness, we omit the obvious steps and only include the necessary ones. Using the fact that for the pure translational motion $h_2 = h_3 = h_4 = 0$, from (3.17), (3.18) and (3.46) with the control law (3.38), we have the following

$$\|\dot{v}_p(t)\| \leq 2k_2 \|e_\alpha\| h_1 + 2h_1 \|d_\psi\|. \quad (3.52)$$

$$\sup_{t \geq 0} \|\ddot{\psi}_p(t)\| = 0. \quad (3.53)$$

Substituting the upper bounds from (3.52), and (3.53) in equation (3.45) produces

$$\begin{aligned} \dot{V}_s \leq & -k_1 \|e_s\|^2 - \frac{1}{\gamma_1} \|e_{\bar{v}_p}\|^2 + \|e_s\| \|d_v\| + \|e_s\| \|e_{\bar{v}_p}\| \\ & - \frac{1}{\gamma_2} \|e_{\psi_p}\|^2 + \|e_\alpha\| \|e_{\psi_p}\| + \|e_\alpha\| \|d_\psi\|. \end{aligned} \quad (3.54)$$

Applying the Cauchy inequality in equation (3.54) produces

$$\begin{aligned} \dot{V}_s \leq & -k_1 \|e_s\|^2 - \frac{1}{\gamma_1} \|e_{\bar{v}_p}\|^2 + \frac{1}{2} \|e_s\|^2 + \frac{1}{2} \|e_{\bar{v}_p}\|^2 + \frac{4k_2^2 h_1^2}{2} \|e_{\bar{v}_p}\|^2 + \frac{1}{2} \|e_\alpha\|^2 \\ & + \frac{1}{2} \|e_s\|^2 + \frac{1}{2} \|d_v\|^2 - k_2 \|e_\alpha\|^2 - \frac{1}{\gamma_2} \|e_{\psi_p}\|^2 + \frac{1}{2} \|e_{\bar{v}_p}\|^2 \\ & + \frac{1}{2} \|e_\alpha\|^2 + \frac{1}{2} \|e_{\psi_p}\|^2 + \frac{1}{2} \|e_\alpha\|^2 + \frac{1}{2} \|d_\psi\|^2 + \frac{4h_1^2}{2} \|d_\psi\|^2. \end{aligned} \quad (3.55)$$

Further simplifying the expressions in (3.55) yields

$$\begin{aligned} \dot{V}_s \leq & -\underbrace{[k_1 - 1]}_{\lambda'_1} \|e_s\|^2 - \underbrace{\left[\frac{1}{\gamma_1} - 2k_2^2 h_1^2 - 1\right]}_{\lambda'_2} \|e_{\bar{v}_p}\|^2 - \underbrace{\left[k_2 - \frac{3}{2}\right]}_{\lambda'_3} \|e_\alpha\|^2 \\ & - \underbrace{\left[\frac{1}{\gamma_2} - \frac{1}{2}\right]}_{\lambda'_4} \|e_{\psi_p}\|^2 + \underbrace{\frac{1}{2} (\|d_v\|^2 + \|\bar{d}_\psi\|^2 (4h_1^2 + 1))}_{D'(t)}. \end{aligned} \quad (3.56)$$

Furthermore,

$$\dot{V}_s \leq -\beta' V_s + D'(t), \quad (3.57)$$

where $\beta' = 2 \times \min(\lambda'_1, \lambda'_2, \lambda'_3, \lambda'_4)$ and $\lim_{t \rightarrow \infty} D'(t) \rightarrow 0$. Then, using the comparison lemma Khalil (2002), we can claim that the errors converge to zero asymptotically. $\square$

3.3 Results

We validate the performance and the associated theoretical claims of the proposed IBVS scheme through realistic simulations in SITL and an outdoor experiment using a quadrotor system as proof of concept. We consider the case of fixed cameras in the SITL simulation, where (3.1) is used to obtain the virtual image plane coordinates. We employ a pan-tilt camera for the outdoor experiments to provide a comprehensive algorithm performance analysis. The choice of a pan-tilt camera also aimed to minimize the impact of environmental disturbances and image blurring due to quadrotor vibrations on the virtual plane reconstruction. Furthermore, we present a comparison with the state-of-the-art super-twisting controller to demonstrate the advantages of the proposed method at the implementation level in addressing unknown time-varying target motions. A video containing the simulation and experimental results can be found here - Video.

3.3.1 Discrete Implementation

While implementing the controller in SITL and outdoor experimental setting, the continuous filters of form $\dot{y}(t) = -ay(t) + au(t)$, used for target velocity estimation, are discretized leveraging the linear system theory as

$$y(t+T) = e^{-aT}y(t) + (1 - e^{-aT})u(t), \quad (3.58)$$

where T is the sampling time. In both the SITL simulation and the experimental setting, the inertial frame is right-handed ENU, and the fixed rotation matrix is $R_c^b = R_x(\pi)R_z(\frac{\pi}{2})$. We consider a rover with a planar visual marker as the target, which

moves on the ground plane. The inertial frame translational velocity of the target plane (rover) can be calculated as $v_p(t) = [v_f(t)\cos(\theta_r(t)), v_f(t)\sin(\theta_r(t)), 0]^T$ where $\theta_r(t)$ is the heading of the target. This shows that even if the forward velocity of the rover is constant, the resulting inertial frame velocity of the target plane can be time-varying if it has an angular motion. The camera intrinsic parameters are considered unknown and obtained through calibration techniques for both cases.

3.3.2 SITL Simulations

To test the efficacy of the proposed scheme, we consider the case of a quadrotor tracking a moving ground rover in a robot operating system (ROS) Stanford Artificial Intelligence Laboratory *et al.* (2018) integrated GAZEBO Koenig and Howard (2004) environment. We create an environment consisting of a hector quadrotor with a camera Meyer *et al.* (2012) and AprilTag equipped husky ground rover Clearpath Robotics (n.d.) as shown in Figure 3.4. The corners of the AprilTag are used as markers to implement the proposed IBVS scheme using ROS nodes. The initial position of the quadrotor in the inertial frame is set to $[1.4, 1.5, 6]^T m$ with the angular orientation (RPY) $[0, 0, \pi]^T rad$. We calculate the desired features by placing the quadrotor at a relative height and heading $3m$ and πrad to the ground rover. The sampling time is set to $T = \frac{1}{30} sec$ to match the camera streaming rate, i.e., $30fps$. Two test cases are considered where the ground rover is exhibiting a constant translational motion and a circular motion with time-varying linear velocity. We choose the following controller gains $k_1 = 0.35$, $k_2 = 0.5$, $\gamma_1 = 0.6$, $\gamma_2 = 0.7$, to design the velocity reference commands for the quadrotor, which are then sent to the internal PID controller Meyer *et al.* (2012) through a ROS node.

Linear and Constant Target Motion

We consider a scenario where the ground rover moves with a constant forward velocity of magnitude $v_f(t) = 0.9m/sec$ without exhibiting any angular motion. The Figures 3.5a, 3.5b, and 3.5c present the results of the scenario, showcasing the trajectory of both the quadrotor and target in frame $\{I\}$, feature tracking error ($e_s = [e_{x_n}, e_{y_n}, e_{a_n}]^T$ and e_α), and target velocity estimation error ($e_{\tilde{v}_p}(t) = [e_{v_{p_x}}, e_{v_{p_y}}, e_{v_{p_z}}]^T$ and $e_{\tilde{\psi}_p}(t)$) re-

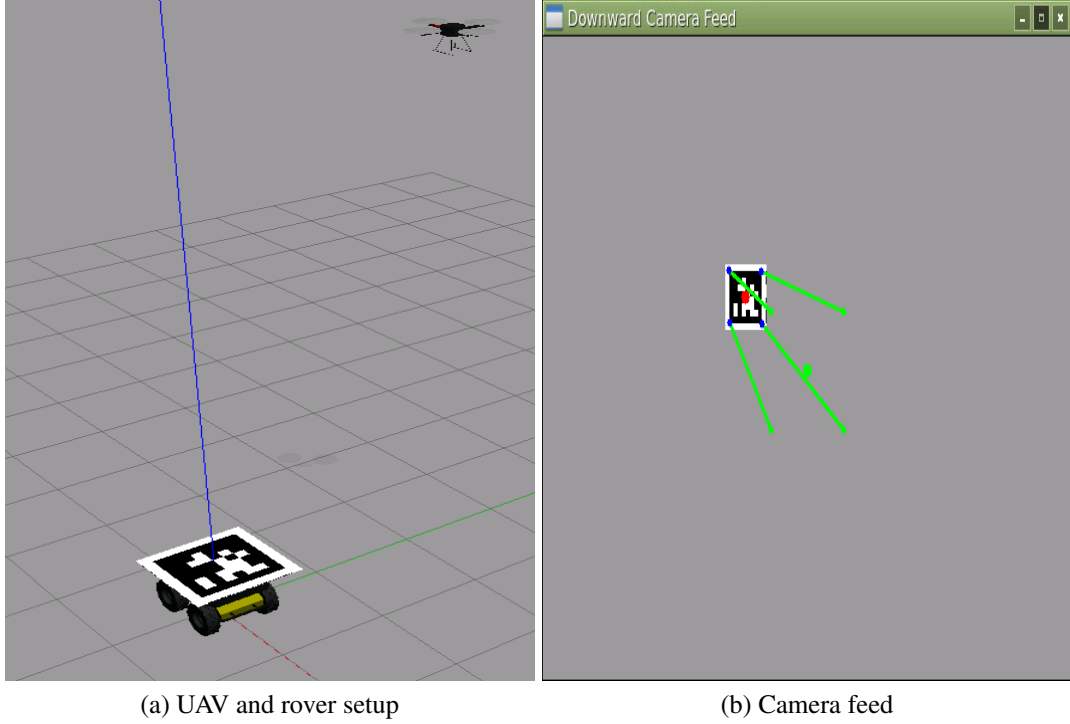


Figure 3.4: SITL environment with ground rover and quadrotor.

spectively. The convergence of the desired error is evident from the feature tracking error and estimation error plot. Moreover, the trajectory plot demonstrates that the proposed IBVS scheme enables the quadrotor to successfully track the target from the desired height in the task space. Remarkably, we observed that the feature tracking errors converged asymptotically despite the inner loop being a simple PID controller. This suggests that the inner-loop mismatch disappears for scenarios like these, where the desired commands become almost constant after the transient.

Time-Varying Target Motion

To evaluate the performance of the proposed scheme in a scenario where the target velocity is time-varying, we considered a case where the ground rover is moving with a time-varying forward velocity $v_f(t) = 0.7 + 0.2\sin(\frac{t}{10})$ and angular velocity $\omega_z(t) = 0.2 \text{ rad/sec}$. The results are presented in Figures 3.6a, 3.6b, and 3.6c showing the trajectory of quadrotor and target, feature tracking error, and target velocity estimation error, respectively. Since the inertial velocity $v_p(t)$ is time-varying now, the converges of estimation error and feature tracking error to a small ultimate bound around origin supports the theoretical claims presented in Theorem 3. Note that this bound is due to

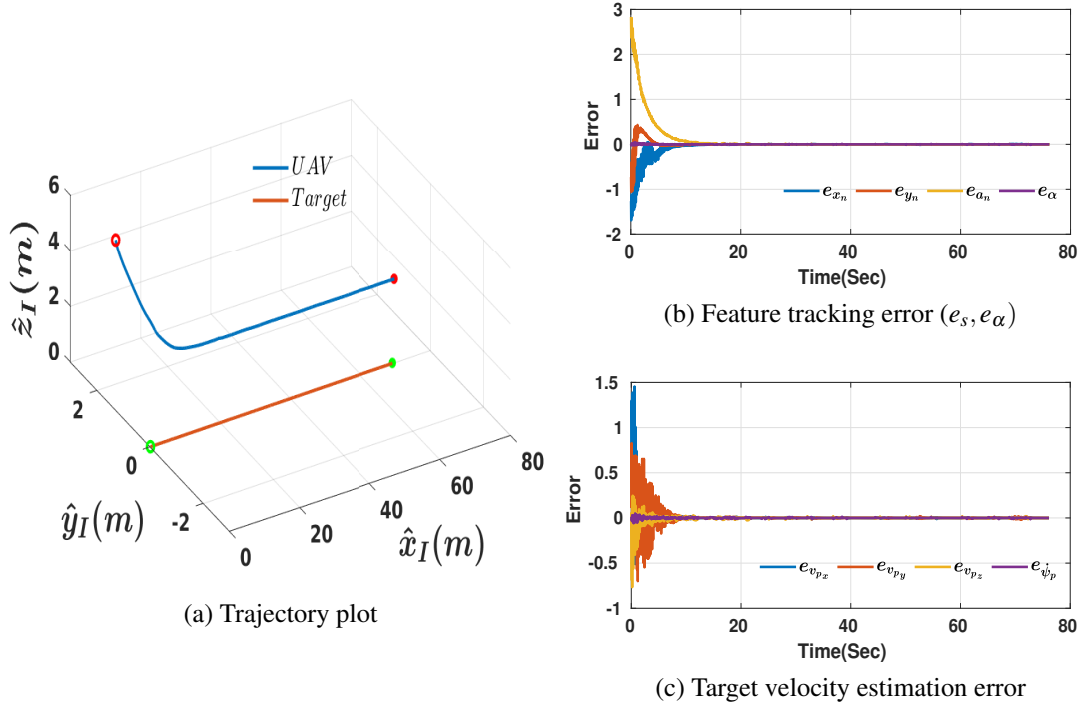


Figure 3.5: SITL results for linear and constant target motion

non-constant target velocity and the inner-loop PID controller for non-constant desired velocity commands.

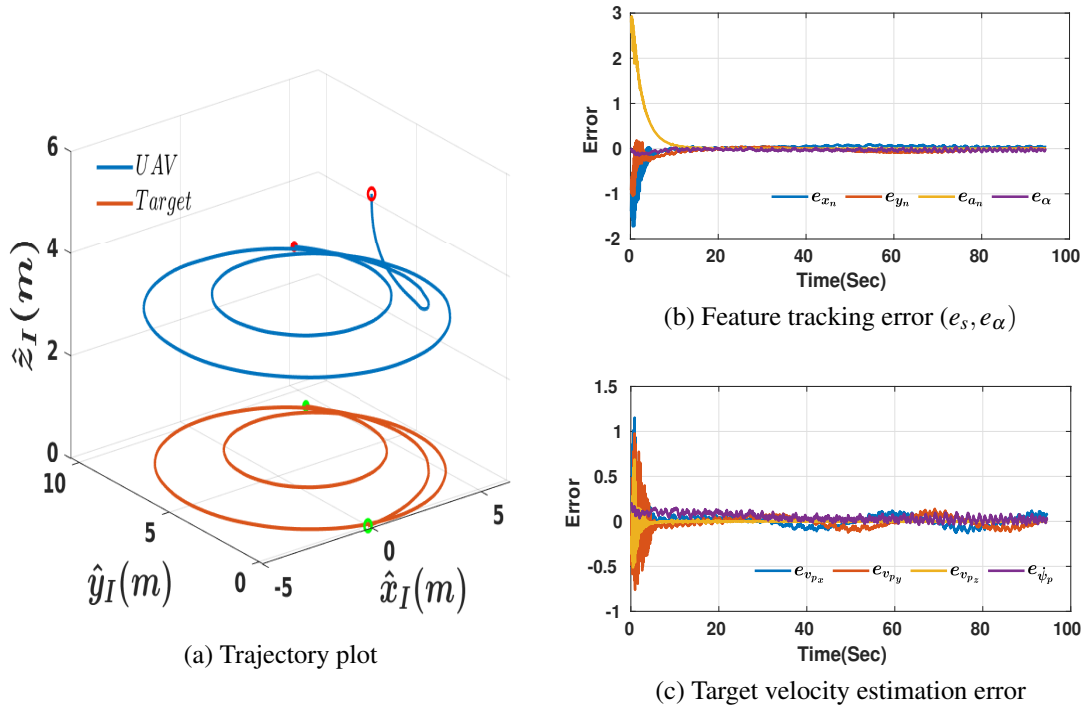


Figure 3.6: SITL results for time-varying target motion

3.3.3 Outdoor Experiments

We also performed outdoor experiments using the 3DR Solo quadrotor with a pan-tilt camera. The complete experimental setup is shown in Figure 3.7. We used the drone’s Wi-Fi network to set up a Mavlink Koubâa *et al.* (2019) interface with the drone. Through this interface, we accessed all the necessary information about the drone and transmitted velocity set-points to the drone using a mavros The MAVROS Website (n.d.) node. The video stream from the camera was also accessed through the drone’s Wi-Fi network. A ground rover with an ArUco marker was used as the target, and its motion was controlled by mimicking the forward and yaw channel transmission commands of a transmitter.

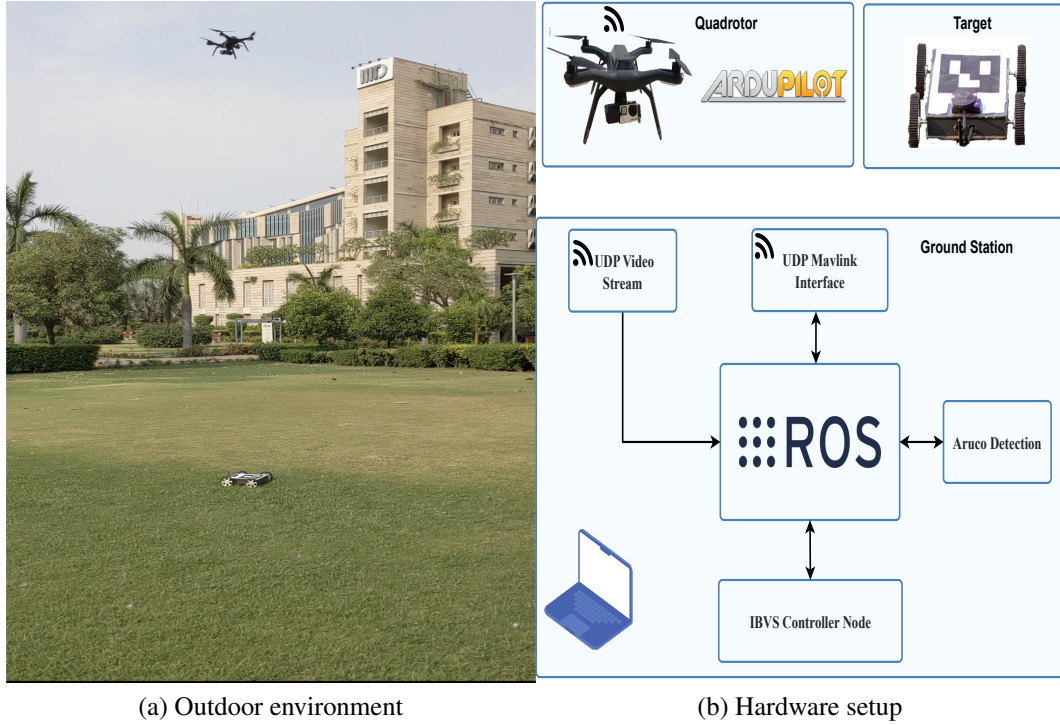


Figure 3.7: Outdoor setup

We controlled the rover to generate a circular motion with unknown velocities. The corners of the ArUco marker were used as the point features to generate the features described in equations (3.2) and (3.4). The sampling time was set to $T = \frac{1}{30}$ sec. The controller and estimator were implemented with the following gains: $k_1 = 0.45$, $k_2 = 0.6$, $\gamma_1 = 0.8$, and $\gamma_2 = 0.8$. The desired height h_d was set to 2 m, and the desired feature points were calculated accordingly. The feature tracking error and target inertial velocity estimates are represented in Figures 3.8a and 3.8b, respectively. The convergence of the error around the origin shows the robustness of the proposed IBVS scheme for track-

ing the target using a quadrotor under outdoor experimental conditions. The observed oscillations around the origin can be attributed to many key factors associated with the outdoor environment and the off-board computation method. The introduction of off-board computation inherently introduced transmission delays, impacting the real-time implementation of control commands. These delays, coupled with the dynamic and time-varying nature of the target's motion, outdoor disturbances notably contributed to the increased error oscillations during tracking. As evident from the supplementary video, these oscillations, while visually apparent, did not significantly compromise the performance or safety of the quadrotor system.

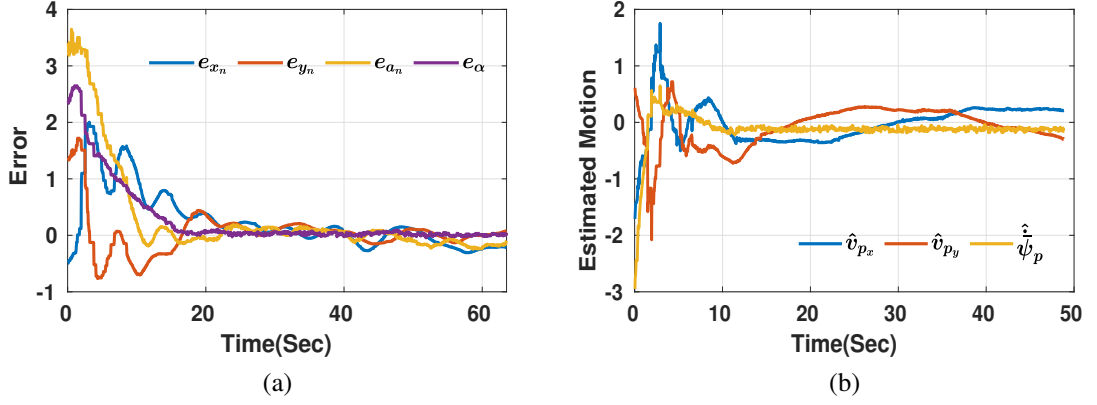


Figure 3.8: Outdoor experiment results. (a) Feature tracking error (e_s, e_{α}) (b) Target motion estimates ($\hat{v}_{p_x}$ and $\hat{v}_{p_y}$ are the estimates of first two elements of target velocity vector $v_p(t)$)

3.3.4 Comparison with Super-twisting Sliding Mode Control

We now compare the proposed scheme with the state-of-the-art super-twisting sliding mode control algorithm. The first step involves designing a control law for the continuous time error dynamics (3.8) and (3.9) and verifying its performance in the MATLAB simulation environment. We then implement the effectively discretized version of this continuous control law in the SITL environment.

Control Law

Based on the error dynamics given in (3.8) and (3.9), we design the following differential ST controller

$$\bar{v}_i = \alpha_1 \sqrt{|e_{s_i}|} \text{sign}(e_{s_i}) - \zeta_{s_i}, \quad \forall i = 1, 2, 3 \quad (3.59)$$

$$\dot{\zeta}_{s_i} = -\beta_1 \text{sign}(e_{s_i}), \quad (3.60)$$

$$\dot{\psi} = \alpha_2 \sqrt{|e_\alpha|} \text{sign}(e_\alpha) - \zeta_\psi, \quad (3.61)$$

$$\dot{\zeta}_\psi = -\beta_2 \text{sign}(e_\alpha), \quad (3.62)$$

where $\bar{v} = [\bar{v}_1, \bar{v}_2, \bar{v}_3]^T$ and $e_s = [e_{s_1}, e_{s_2}, e_{s_3}]^T$. The design parameters $\alpha_1, \beta_1, \alpha_2$ and β_2 are the positive design parameters and are chosen according to Koch *et al.* (2019).

Discretization

Efficient discretization is essential during the real-time implementation of sliding mode controllers, as this may cause additional chattering if not handled carefully. We use the matching method proposed in Koch *et al.* (2019), and the following discretization for sampling time T is used for the control laws proposed in section 3.3.4

$$\begin{aligned} \bar{v}_i(t) &= \alpha_1 \Theta_1(e_{s_i}(t), \alpha_1, \beta_1) - \zeta_{s_i}(t+T), \\ \zeta_{s_i}(t+T) &= \zeta_{s_i}(t) - T\beta_1 \Theta_2(e_{s_i}(t), \alpha_1, \beta_1), \quad \forall i = 1, 2, 3 \end{aligned} \quad (3.63)$$

$$\begin{aligned} \dot{\psi}(t) &= \alpha_2 \Theta_1(e_\alpha(t), \alpha_2, \beta_2) - \zeta_\psi(t+T), \\ \zeta_\psi(t+T) &= \zeta_\psi(t) - T\beta_2 \Theta_2(e_\alpha(t), \alpha_2, \beta_2), \end{aligned} \quad (3.64)$$

where the state-dependent functions $\Theta_1(\circ, \alpha_i, \beta_i)$ and $\Theta_2(\circ, \alpha_i, \beta_i)$ are defined as follows

$$\begin{aligned} \Theta_1(\circ, \alpha_i, \beta_i) &= -\frac{1}{\alpha_i T} \left(e^{\frac{p_1 T}{\sqrt{\|\circ\|}}} + e^{\frac{p_2 T}{\sqrt{\|\circ\|}}} - 2 \right) \circ \\ &\quad - \frac{T\beta_i}{\alpha_i} \Theta_2(\circ, \alpha_i, \beta_i), \end{aligned} \quad (3.65)$$

$$\Theta_2(\circ, \alpha_i, \beta_i) = \frac{1}{T^2 \beta_i} \left(e^{\frac{p_1 T}{\sqrt{\|\circ\|}}} - 1 \right) \left(e^{\frac{p_2 T}{\sqrt{\|\circ\|}}} - 1 \right) \circ, \quad (3.66)$$

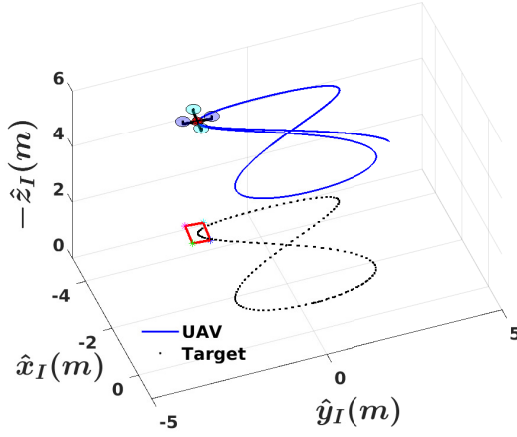
with the parameters p_1, p_2 as described in Koch *et al.* (2019).

Comparative MITL Simulations

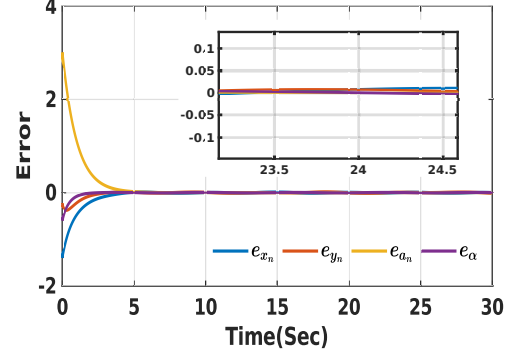
We first present the performance of the ST controller in the continuous time implementation setting compared to our method. We create an environment in MATLAB consisting of a camera-attached quadrotor with $R_c^b = I$ and the target. We create a camera model with the focal length 15 mm , principal point $u_0 = v_0 = 512 \text{ pixel}$, pixel height and width $\rho_h = \rho_w = 10^{-5} \text{ m/pixel}$, and resolution 1024×1024 pixels. We consider four markers on a planar target with the following initial positions $P_1 = [-0.36, -0.25, -1]^T \text{ m}$, $P_2 = [0.36, -0.25, -1]^T \text{ m}$, $P_3 = [0.36, 0.25, -1]^T \text{ m}$, $P_4 = [-0.36, 0.25, -1]^T \text{ m}$, expressed in the inertial frame. The initial position of the quadrotor is considered as $[0, 0, -8]^T \text{ m}$ with the angular orientation (RPY) $[0, 0, 0.6]^T \text{ rad}$, also expressed in the inertial frame. The desired features are selected based on the requirement that the quadrotor tracks the moving target while maintaining a constant height $h_d = 4 \text{ m}$ above it. The target is considered to move with linear and angular velocities varying over time $v_p(t) = [\cos(0.5t + \pi/2), 2\cos(t), -0.2\sin(t)]^T \text{ m/sec}$ and $\dot{\psi}_p(t) = \frac{\pi}{10} - \frac{\pi}{30}\sin(t) \text{ rad/sec}$, respectively. The IBVS controller parameters are chosen as follows

- Proposed Method $\rightarrow k_1 = 1, \quad k_2 = 2, \quad \gamma_1 = 10^{-2}, \quad \gamma_2 = 10^{-1}.$
- ST $\rightarrow \alpha_{1/2} = 1.5\sqrt{\Lambda_{1/2}}, \quad \beta_{1/2} = 1.1\Lambda_{1/2}, \quad \Lambda_1 = 5, \quad \Lambda_2 = 2.$

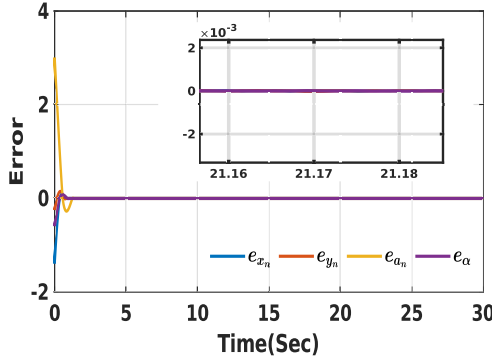
Since we performed this comparison to show the efficacy of the proposed method in handling arbitrary target motions with a theoretical performance almost similar to the state-of-the-art sliding-mode controller, we assume that the inner-loop controller perfectly tracks the provided velocity commands without any mismatch. The MITL simulation results are compiled in Figure 3.9. Figure 3.9a shows the trajectory plot of the quadrotor and the target with the proposed controller. Figures 3.9b and 3.9c present the feature tracking error with the proposed control and ST IBVS controller, respectively. It can be observed from the feature tracking error plots that the error converges faster and perfectly to the origin when we use the ST controller. Whereas the tracking error convergence is comparatively slow in the case of the proposed method to the ultimate bound within the origin as claimed in Theorem 3. Further, the velocity estimation error with the proposed controller is presented in Figure 3.9d, showing its convergence to the small ultimate bound within the origin.



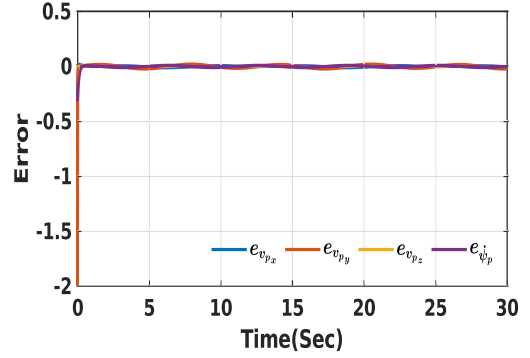
(a) Trajectory plot with proposed controller



(b) Feature tracking error with proposed controller



(c) Feature tracking error with ST controller



(d) Target velocity estimation error with proposed controller

Figure 3.9: Comparative MIL simulation results.

Comparative SITL Simulation

We consider a gazebo environment similar to section 3.3.2 with the arbitrary target motion scenario and perform the simulations with the discrete ST controller given in (3.63) and (3.64). The following controller gains and discretization parameters are chosen after tuning for the best possible performance in the considered environment

$$\begin{aligned}\alpha_{1/2} &= 1.5\sqrt{\Lambda_{1/2}}, \quad \beta_{1/2} = 1.1, \quad \Lambda_{1/2}, \\ \Lambda_1 &= 1.5, \quad \Lambda_2 = 1.5, \quad p_1, p_2 = -0.31.\end{aligned}\tag{3.67}$$

Figures 3.10a and 3.10b show the trajectory plot and the feature tracking error for the considered target motion. Compared to the results with the proposed method in Figure 3.6, one can notice the oscillatory behavior of the error signals and the quadro-

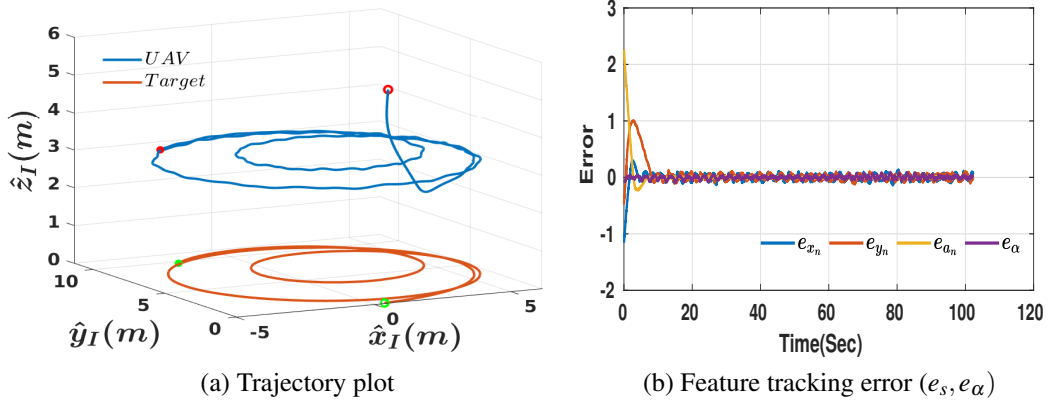


Figure 3.10: SITL results with ST controller

tor trajectory in Figure 3.10. This oscillatory behavior is due to chattering phenomena. To better understand, we plot the control input for both the controllers and the results are shown in Figure 3.11. As captured in Figure 3.11 our method provides a smooth controller, while the ST controller exhibits chattering and hence yields the undesirable oscillatory quadrotor trajectory. We have seen from the results in Figure 3.9 that the ST controller performs better when implemented in the continuous setting. However, when implemented in a discrete setting (real-time implementation), it displays undesirable chattering behavior, which is particularly unfavorable for mechanical systems like quadrotors.

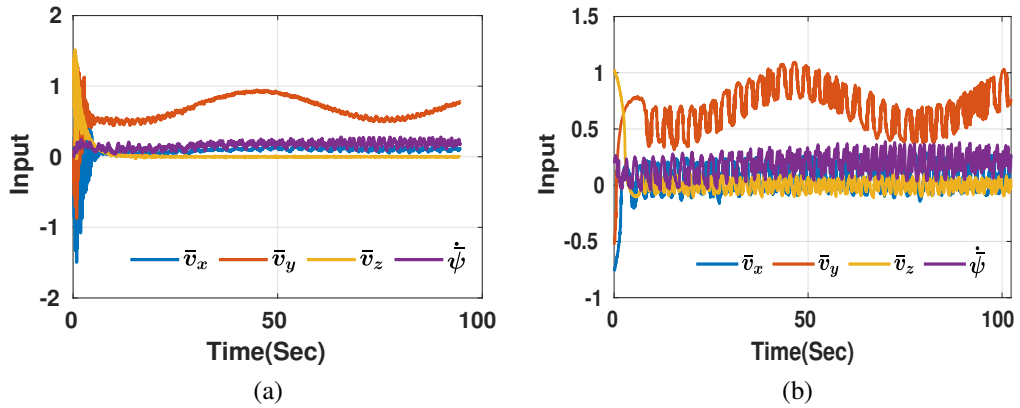


Figure 3.11: SITL simulation control input. (a) Control input with proposed controller (b) Control input with ST controller

3.4 Conclusion

This chapter advanced the development of IBVS control by adapting it for underactuated quadrotor systems. In this chapter, we addressed the challenge of tracking an arbitrarily moving planar target using a quadrotor with visual feedback. A target velocity estimator was proposed to estimate bounded arbitrary translational and angular motions along the target's z-axis, coupled with a kinematic IBVS control law for the quadrotor. The rigorous analysis demonstrated that the tracking and estimation errors converge to an ultimate bound, which can be minimized by appropriately tuning the estimator gains. However, the ultimate bound caused by model mismatch must be managed through effective inner-loop controller design. Simulation and experimental results validated the proposed method's effectiveness in tracking arbitrarily moving targets. Additionally, SITL simulations revealed that the proposed controller outperforms the ST controller, further highlighting its utility.

Chapter 4

Adaptive IBVS-based Planar Non-Holonomic Target Tracking for Quadrotors

This chapter builds upon the kinematic IBVS controller developed in Chapter 3, which used virtual image plane-based features to ensure dynamic feasibility and eliminate explicit dependence on depth measurements by introducing the concept of desired feature depth. While this approach simplified the control design, it assumed that the desired depth is known. However, tracking moving targets with unknown desired depth remains a significant challenge. To address this, the current chapter presents adaptive IBVS kinematic controllers for a quadrotor to track a planar non-holonomic target moving with constant body-fixed linear velocity and arbitrary angular velocity about the z-axis, with an unknown desired feature depth. Using optical measurements from the virtual image plane, image moment-based features are derived to design decoupled references for the translational and angular motion of the quadrotor. By parameterizing the target's constant translational velocity through target heading reconstruction, the adaptive control problem is formulated. Two methods for translational feature kinematics are introduced: (1) a bilinear estimator-augmented adaptive IBVS controller that guarantees feature tracking error convergence and (2) an adaptive IBVS controller also ensuring parameter convergence under the initial excitation (IE) condition Roy *et al.* (2016). This is unlike the classical adaptation, where parameter convergence is achieved under a restrictive condition of persistence of excitation (PE) Narendra and Annaswamy (2012). The IE condition is significantly milder than the classical PE condition Roy *et al.* (2018); Goel and Roy (2024).

A heading controller is developed to address the target's unknown angular motion, assuming the availability of an inner-loop controller similar to Bourquardez *et al.* (2009), which efficiently tracks reference velocity commands for the quadrotor's translational and angular kinematics. Lyapunov analysis is employed to establish that the translational feature error dynamics are asymptotically convergent for Method I and exponentially convergent for Method II. The closed-loop error dynamics for the rotational subsystem are proven to be uniformly ultimately bounded. The efficacy of both

methods is validated through realistic SITL simulations in a high-fidelity environment and model-in-the-loop simulations. Results highlight improved feature tracking performance for translational kinematics with Method II, as it achieves parameter convergence, outperforming Method I.

4.1 Problem Formulation

We briefly revisit the reconstruction of the virtual image plane, followed by the definition of image moment-based features and their associated error dynamics. This includes the constant parametrization of the unknown target motion. First, Let's define the coordinate frames used throughout this chapter. Let $\{I\}$, $\{b\}$, and $\{c\}$ be the inertial, the quadrotor body frames, and the camera frame, with origins O_I and O_b . The virtual image plane is constructed by introducing a virtual camera frame $\{v\}$ whose motion is invariant to the roll and pitch motion of the quadrotor, distinguishing it from the actual camera. For detailed instructions on reconstructing the virtual image plane, including configurations with fixed and pan-tilt camera attachments, please refer to Section 3.1.1 in Chapter 3.

We use these virtual image plane visual coordinates to define the image features to control the quadrotor motion. Let's consider N observable points on a planar target and the corresponding virtual image plane coordinates $s_{v_k} = [u_{v_k}, v_{v_k}]^T, k = 1, \dots, N$. Further, $[x_g, y_g]^T = [\frac{1}{N} \sum_{k=1}^N u_{v_k}, \frac{1}{N} \sum_{k=1}^N v_{v_k}]^T$ be the centroid of these points in the virtual image plane. The features $s'(t) = [x_n(t), y_n(t), a_n(t)] \in \mathbb{R}^3$ for translational motion, and $\alpha(t) \in \mathbb{R}$ for yaw motion are defined as

$$x_n = \frac{a_n}{f} x_g, y_n = \frac{a_n}{f} y_g, a_n = \sqrt{\frac{a_d}{a}}, \quad (4.1)$$

$$\alpha = \frac{1}{2} \arctan \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right), \quad (4.2)$$

where $a = \sum_{k=1}^N ((u_{v_k} - x_g)^2 + (v_{v_k} - y_g)^2)$. a_d is the desired value of a , calculated based on the desired image measurements. $\mu_{ij} = \sum_{k=1}^N (u_{v_k} - x_g)^i (v_{v_k} - y_g)^j$ is the centered moment of order $i + j$. Note that, due to the unknown desired depth, the features derived in equations (4.1) and (4.2) differ from those presented in Section 3.1.2 of Chapter 3.

4.1.1 Image Feature Dynamics

The dynamics of $s'(t)$ and $\alpha(t)$ when the target is moving with translational velocity $\bar{v}_p(t) \in \mathbb{R}^3$ and angular rate $[0, 0, \dot{\psi}_p(t)]^T \in \mathbb{R}^3$, expressed in $\{v\}$, can be given as

$$\dot{s}' = -\frac{1}{h_d}(\bar{v} - \bar{v}_p) - S_k([0, 0, \dot{\psi}]^T)s', \quad (4.3)$$

$$\dot{\alpha} = -\dot{\psi} + \dot{\psi}_p. \quad (4.4)$$

where $[0, 0, \dot{\psi}(t)]^T \in \mathbb{R}^3$ and $\bar{v}(t) \in \mathbb{R}^3$, are the angular rate and the linear velocity of the virtual camera $\{v\}$, expressed in the $\{v\}$. $S_k(.) : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$ denotes the skew-symmetric map. h_d is the unknown desired height (also called desired depth) of the virtual camera from the planar target.

Remark 4. Note that the desired image is required to calculate the desired features and a_d , which is utilized in the s' calculations. In cases where one wants to track the target from a user-defined altitude in $\{I\}$, demands explicit knowledge of target height (height of marker) in $\{I\}$ frame to calculate the h_d . We consider such a scenario where only the desired image is available but not the h_d .

4.1.2 Error Formulation

The objective is to control the quadrotor kinematics such that the target remains in the center of the virtual image plane while maintaining an unknown height h_d above the target in reference to a known a_d . Hence, the desired translational feature is chosen as $s'^* = [0, 0, 1]$. For heading, a suitable α^* is defined depending on the shape of the convex hull of feature points in the image plane and can be obtained through the desired feature points in the virtual image plane using the relation (4.2). We define feature-tracking errors as

$$e_s(t) = s'(t) - s'^*, \quad (4.5)$$

$$e_\alpha(t) = \alpha(t) - \alpha^*. \quad (4.6)$$

Using the fact that the target translational velocity $\bar{v}_p(t) = R_c^{bT} R_z(\psi(t))^T v_p(t)$, given $v_p(t)$ is translational velocity expressed in $\{I\}$, the open-loop feature tracking error

dynamics becomes

$$\dot{e}_s = -S_k([0, 0, \dot{\tilde{\psi}}]^T)^T e_s - \lambda \left(\bar{v} - R_c^{bT} R_z(\psi(t))^T v_p(t) \right), \quad (4.7)$$

$$\dot{e}_\alpha = -\dot{\tilde{\psi}} + \dot{\tilde{\psi}}_p, \quad (4.8)$$

where λ is an unknown positive constant such that $\lambda = \frac{1}{h_d}$.

Remark 5. *An effective controller can be designed for heading control based on error dynamics (4.8), also tackling unknown $\dot{\tilde{\psi}}_p(t)$ simultaneously following the approach given in Chapter 3. However, it's non-trivial to handle arbitrary $v_p(t)$ in the adaptive control framework using bilinear estimation for the error dynamics (4.7) Lavretsky (2009); Patil et al. (2022). One can apply the results from Patil et al. (2022), however, with the stringent consideration that only the constant part of $v_p(t)$ (if at all present) will be estimated and the time-varying part will be handled by the robustifying gain in the control law. For the problem under consideration, the target's motion can be divided into two categories: (i) holonomic and (ii) non-holonomic. In both categories, we can only design the control law for translational motion using bilinear estimation when there is a parameterization of the target motion available in the form of constant parameters. This shrinks the span of possible motions the target can exhibit, i.e., constant v_p . For non-holonomic targets, this requirement limits the target motion to a pure translational motion, as any angular motion will make the $v_p(t)$ time-varying. In this chapter, we are addressing the vital question of whether we can expand this span for non-holonomic targets to incorporate the effect of angular motion in the formulation. We also aim to analyze the conditions under which parameter convergence can be guaranteed and investigate the transient behavior of the feature-tracking errors in those scenarios. We propose the following constant parameterization of target motion based on novel target heading reconstruction to investigate the same.*

4.1.3 Target Velocity Parameterization

For a non-holonomic target, the unknown translational velocity can be expressed as $\bar{v}_p(t) = R_c^{bT} R_z(\psi(t))^T R_z(\psi_p(t)) v_{pb}(t)$, where $\psi_p(t) \in (-\pi, \pi]$ and $v_{pb}(t) \in \mathbb{R}^3$ are the heading of target with respect to the body fixed inertial frame and body-fixed linear velocity. To design the adaptive control for error dynamics (4.7), we define a recon-

struction of $\bar{\psi}_p(t)$ using the feature tracking error $e_\alpha(t)$ and it's dynamics (4.8). The heading of the target $\psi_p(t)$ expressed in $\{v\}$ can be reconstructed as

$$\bar{\psi}_p(t) = e_\alpha(t) + \bar{\psi}(t) + \bar{\psi}_{Offset}, \quad (4.9)$$

where $\bar{\psi}_{Offset}$ is the desired relative heading of the target with respect to the quadrotor. One can further obtain the target heading ψ_p using the explicit relation $R_z(\bar{\psi}_p) = R_z(\psi_p)R_c^b$. The open-loop feature tracking error dynamics thus becomes

$$\dot{e}_s = -S_k([0, 0, \dot{\bar{\psi}}]^T)e_s - \lambda \bar{v} + \lambda R_c^{bT} R_z(\psi)^T R_z(\psi_p) v_{pb}, \quad (4.10)$$

$$\dot{e}_\alpha = -(\dot{\bar{\psi}} - \dot{\bar{\psi}}_p). \quad (4.11)$$

Assumption 5. *The target motion is such that v_{pb} is constant and*

$$\sup_{t \geq 0} \|\dot{\bar{\psi}}_p(t)\| = h_1, \quad \sup_{t \geq 0} \|\ddot{\bar{\psi}}_p(t)\| = h_2, \quad (4.12)$$

where $h_1, h_2 \in \mathbb{R}_+$ are finite constants.

4.1.4 Control Objective

Given the error dynamics (4.10) and (4.11), the control objective is to devise the translational ($\bar{v}(t)$) and rotational ($\dot{\bar{\psi}}(t)$) velocities for the virtual camera frame such that feature tracking errors $e_s(t)$, $e_\alpha(t) \rightarrow 0$ as $t \rightarrow \infty$, while compensating for unknown target motion and uncertain parameter λ . Note that the control input considered here represents the velocities of the virtual camera expressed in the virtual camera frame. For kinematic control of quadrotor, one can use the relations $v(t) = R_z(\psi(t))R_c^b \bar{v}(t)$ and $R_z(\bar{\psi}) = R_z(\psi)R_c^b$ to obtain the linear velocity $v(t)$ of frame $\{b\}$ expressed in $\{I\}$ and an explicit relation between $\psi(t)$ (yaw angle of the quadrotor body frame with respect to the body fixed inertial frame) and $\bar{\psi}(t)$ (yaw angle of the virtual frame with respect to the body fixed inertial frame). Please refer to Section 3.1.4 in Chapter 3 for more details.

4.2 Control Design

Assuming the availability of an inner-loop controller, the outer-loop control design for error dynamics (4.10) and (4.11) is done separately, exploiting the decoupled feature dynamics.

4.2.1 Control Law- Translational Kinematics

First Layer Filtering

To design the control law for error dynamics given in equation (4.10), first consider the following filter structures with the measurable input signals $e_s(t)$, $\dot{\psi}(t)$, $\bar{v}(t)$, $R_z(\psi_p(t))$, and $R_z(\psi(t))$ as

$$\gamma_1 \dot{e}_{fs} + e_{fs} = e_s, \quad e_{fs}(0) = 0, \quad (4.13)$$

$$\gamma_1 \dot{h}_s + h_s = S_k([0, 0, \dot{\psi}]^T) e_s, \quad h_s(0) = 0, \quad (4.14)$$

$$\gamma_1 \dot{v}_f + v_f = \bar{v}, \quad v_f(0) = 0, \quad (4.15)$$

$$\gamma_1 \dot{R}_f + R_f = R_c^{b^T} R_z(\psi)^T R_z(\psi_p), \quad R_f(0) = 0 \in \mathbb{R}^{3 \times 3}. \quad (4.16)$$

$\gamma_1 > 0$ is a (small) positive constant. Note that in principle, we can make the output signal of the above filters arbitrarily close to the input signals by choosing a small γ_1 closer to 0. Let's consider another filter to obtain the information about the unknown signal $\dot{e}_s(t)$ as

$$\gamma_1 \dot{g}_s + g_s = \dot{e}_s, \quad g_s(0) = 0, \quad (4.17)$$

where $g_s(t)$ is the filter output. Although $\dot{e}_s(t)$ is not measurable, the analytical solution of equation (4.17) for $g_s(t)$ can be computed in terms of known quantities as

$$g_s(t) = \frac{1}{\gamma_1} \int_0^t \exp\left(-\frac{1}{\gamma_1}(t-\tau)\right) \dot{e}_s(\tau) d\tau \quad (4.18)$$

$$= \frac{1}{\gamma_1} \left(e_s(t) - e_{fs}(t) - \exp\left(-\frac{1}{\gamma_1}t\right) e_s(0) \right). \quad (4.19)$$

Lemma 5. Consider a filtered variable $S_s(t) \in \mathbb{R}^3$ such that

$$\dot{S}_s = \frac{1}{\gamma_1} \left(\lambda \left(\bar{v} - R_c^{bT} R_z(\psi)^T R_z(\psi_p) v_{pb} \right) - S_s \right), \quad S_s(0) = 0. \quad (4.20)$$

Then, we have

$$S_s(t) = \lambda(v_f(t) - R_f(t)v_{pb}) = -(g_s + h_s). \quad (4.21)$$

Proof. Substituting the error dynamics (4.10) in equation (4.18), we get

$$\begin{aligned} g_s(t) = & -\frac{1}{\gamma_1} \int_0^t \exp\left(-\frac{1}{\gamma_1}(t-\tau)\right) \left(S_k([0, 0, \dot{\psi}]^T) e_s(\tau) \right. \\ & \left. + \lambda \left(\bar{v}(\tau) - R_c^{bT} R_z(\psi(\tau))^T R_z(\psi_p(\tau)) v_{pb} \right) \right) d\tau \end{aligned} \quad (4.22)$$

Further solving equation (4.22) using equations (4.14), (4.15) and (4.16) yields the relation (4.21). $\square$

Method I

We utilize the measurable signal $S_s(t)$ to design the control law $\bar{v}(t)$ incorporating the estimation laws for the uncertainties. Consider a bi-linear predictor model $\hat{S}_s(t)$ for $S_s(t)$ inspired from Lavretsky (2009) as

$$\hat{S}_s(t) = \hat{\lambda}(t)(v_f(t) - R_f(t)\hat{v}_{pb}(t)), \quad (4.23)$$

where the differential laws for $\hat{\lambda}(t)$ and $\hat{v}_{pb}(t)$ are defined as

$$\begin{aligned} \dot{\hat{v}}_{pb} = & -k_p R_f^T (S_s - \hat{S}_s) + R_z(\psi_p)^T R_z(\psi) R_c^b e_s, \\ \dot{\hat{\lambda}} = & (v_f - R_f \hat{v}_{pb})^T (S_s - \hat{S}_s), \end{aligned} \quad (4.24)$$

where k_p is a positive design parameter.

The control law $\bar{v}(t)$ is then designed as

$$\bar{v}(t) = k_s e_s(t) + R_c^{bT} R_z^T(\psi(t)) R_z(\psi_p(t)) \hat{v}_{pb}(t), \quad (4.25)$$

where k_s is a positive controller gain.

Remark 6. *The control and estimation laws derived following Method I would only ensure the asymptotic convergence of feature tracking error $e_s(t)$ and boundedness of the parameter estimation errors corresponding to v_{pb} and λ . To achieve parameter convergence, we introduce a second layer of filters. These filters will be instrumental in designing enhanced parameter estimation laws, ensuring boundedness and convergence of the parameter estimates to their true values.*

Second Layer Filtering and Estimation Laws

In this section, we introduce a filtering technique for measurable signal $S_s(t)$ to design the parameter estimation laws based on it. The estimation laws defined subsequently ensure that under an online-verifiable condition called initial excitation (IE), parameter estimation error converges to zero. Let's write (4.21) in the compact format as

$$S_s(t) = \lambda(v_f(t) - R_f(t)v_{pb}) = \Phi_f^T \Theta, \quad (4.26)$$

where

$$\Theta = \begin{bmatrix} \lambda \\ \lambda v_{pb} \end{bmatrix}. \quad (4.27)$$

$$\Phi_f = \begin{bmatrix} v_f & -R_f \end{bmatrix}^T. \quad (4.28)$$

The new parameter $\Theta \in \mathbb{R}^4$ is strategically designed to handle the parameter estimation with provable convergence in the bilinear estimation settings such as given in (4.21). It is worthwhile to note that an estimation $\hat{\Theta} = [\hat{\Theta}_1, \hat{\Theta}_2, \hat{\Theta}_3, \hat{\Theta}_4]^T = [\hat{\lambda}, \hat{\lambda} \hat{v}_{pb}^T]^T$ such that $\Theta - \hat{\Theta} \rightarrow 0$ implies that $\lambda - \hat{\lambda}, v_{pb} - \hat{v}_{pb} \rightarrow 0$.

Let's define another layer of filtering for computable signals Φ_f and S_s as

$$\dot{P} = \alpha(t) \Phi_f \Phi_f^T, \quad P(0) = 0, \quad (4.29)$$

$$\dot{Q} = \alpha(t) \Phi_f S_s, \quad Q(0) = 0, \quad (4.30)$$

where the signal $\alpha(t)$ a scalar function with following properties

$$0 < \alpha(t) < \bar{\alpha} < \infty, \text{ for } t \in [t_0, \infty), \text{ and } \alpha(t) \in \mathcal{L}_1 \cap \mathcal{L}_\infty,$$

where $\bar{\alpha}$ is an upper bound of $\alpha(t)$.

Analytically solving (4.29) and (4.30) yields

$$Q = P\Theta \quad (4.31)$$

Assumption 6. The filtered measurable signal $\Phi_f \in \mathbb{R}^{3 \times 4}$ is initially exciting, i.e., $\exists \Lambda, T_{IE} > 0$, such that

$$\int_0^{T_{IE}} \Phi_f(r) \Phi_f^T(r) dr \geq \Lambda I_4. \quad (4.32)$$

Lemma 6. Given assumption 6, there exists $\alpha > 0$ such that $P(t) \geq \alpha \Lambda I_4 > 0, \forall t \geq T_{IE}$.

Proof. Please refer to (Garg *et al.*, 2022, Lemma 1) for the proof. $\square$

Next, we design another set of control and parameter estimation laws for the feature error related to translational kinematics.

Method II

We propose the following control and estimation laws

$$\bar{v} = \hat{\Theta}_1^{-1} (k_s e_s + R_c^{b^T} R_z(\psi)^T R_z(\psi_p) \hat{\Theta}') + u_{ex}(t), \quad (4.33)$$

$$\dot{\hat{\Theta}}_1 = Proj(\hat{\Theta}_1, \Delta_1), \quad \hat{\Theta}_1(0) > 0 \quad (4.34)$$

$$\hat{\Theta}' = \begin{bmatrix} \dot{\hat{\Theta}}_2 \\ \dot{\hat{\Theta}}_3 \\ \dot{\hat{\Theta}}_4 \end{bmatrix} = \begin{bmatrix} \Delta_2 \\ \Delta_3 \\ \Delta_4 \end{bmatrix}, \quad (4.35)$$

where $\Phi = \begin{bmatrix} -\bar{v} & R_c^{b^T} R_z(\psi)^T R_z(\psi_p) \end{bmatrix}^T$ and $\Delta = [\Delta_1, \Delta_2, \Delta_3, \Delta_4]^T = \Gamma \Phi e_s - \Gamma \gamma_3 (P\hat{\Theta} - Q) \in \mathbb{R}^4 \times 4$. $u_{ex}(t) \in \mathcal{L}_\infty$ is an exponentially decaying sinusoidal probing signal such that $u_{ex}(t) \rightarrow 0$ as $t \rightarrow \infty$. k_s, γ_3 are positive scalar design parameters and $\Gamma \in \mathbb{R}^{4 \times 4}$ is a positive definite gain matrix for parameter estimator.

The projection operator is applied for the estimation of $\hat{\Theta}_1(t)$ (i.e. $\hat{\lambda}(t)$) to ensure

a boundedness of control law $\bar{v}(t)$ ($\hat{\lambda}(t) \neq 0$). Further, from a practical perspective, the desired height can never be less than the focal length(in meters) of a camera, and hence $\lambda < 1/(\text{focal length})$. Since λ is a positive quantity by design, these constraints are imposed by the following projection operator

$$Proj(\hat{\Theta}_1, \Delta_1) = \begin{cases} \left(1 + \frac{\hat{\Theta}_1 - \lambda}{\varepsilon}\right) \Delta_1, & \text{if } \hat{\Theta}_1 < \lambda \text{ \& } \Delta_1 < 0, \\ \left(1 + \frac{\bar{\lambda} - \hat{\Theta}_1}{\varepsilon}\right) \Delta_1, & \text{if } \hat{\Theta}_1 > \bar{\lambda} \text{ \& } \Delta_1 > 0, \\ \Delta_1, & \text{otherwise} \end{cases} \quad (4.36)$$

Here $\bar{\lambda}$ and λ is the user-supplied upper and lower bound of λ . The projection operator-based update law ensures $0 < \lambda - \varepsilon < \hat{\Theta}_1 < \bar{\lambda} + \varepsilon$ where $\varepsilon > 0$ is the tolerance. Note that $\Theta(0)$ and ε are chosen in compliance with the aforementioned constraints. Next, we present the control law for the heading motion of the quadrotor based on error dynamics (4.11).

4.2.2 Control Law- Rotational Kinematics

Consider the following filter dynamics for measurable signals $e_\alpha(t)$ and $\dot{\psi}(t)$ with design parameter $\gamma_2 > 0$ for the error dynamics (4.11)

$$\gamma_2 \dot{e}_{f\psi} + e_{f\psi} = e_\alpha, \quad e_{f\psi}(0) = 0, \quad (4.37)$$

$$\gamma_2 \dot{h}_\psi + h_\psi = \dot{\psi}, \quad h_\psi(0) = 0, \quad (4.38)$$

with filtered output $e_{f\psi}(t)$ and $h_\psi(t)$. Following the developments similar to Section 3.2.2 in Chapter 3, the control law is designed as

$$\dot{\psi}(t) = k_\alpha e_\alpha(t) + \hat{\psi}_p(t), \quad (4.39)$$

where k_α is a positive controller gain and

$$\hat{\psi}_p(t) = h_\psi(t) + \frac{1}{\gamma_2} \left(e_\alpha(t) - e_{f\psi}(t) - \exp\left(-\frac{1}{\gamma_2}t\right) e_\alpha(0) \right). \quad (4.40)$$

4.2.3 Stability and Convergence - Translation Kinematics

We define the following errors for the bi-linear predictor model used to define the estimation laws

$$e_V(t) = v_{pb} - \hat{v}_{pb}(t), \quad (4.41)$$

$$e_\lambda(t) = \lambda - \hat{\lambda}(t), \quad (4.42)$$

$$\begin{aligned} e_P(t) &= S_s(t) - \hat{S}_s(t) \\ &= -\lambda R_f(t) e_V(t) + e_\lambda(t) (v_f(t) - R_f(t) \hat{v}_{pb}(t)). \end{aligned} \quad (4.43)$$

Next, we provide the stability and convergence proof for translational kinematic feature tracking errors under the control and estimation laws of Method I.

Theorem 4. (Method I). Consider the estimation and control laws defined in (4.25) and (4.24), the closed-loop error dynamics of $e'_s(t) = [e_s(t), e_\lambda(t), e_V(t)]$ is Lyapunov stable and the errors $e_s(t), e_P(t) \rightarrow 0$ as $t \rightarrow \infty$ asymptotically in a global sense.

Proof. Consider the following Lyapunov candidate for the complete error system as

$$V_s = \frac{1}{2} \lambda e_V^T e_V + \frac{1}{2} k_p e_\lambda^2 + \frac{1}{2} e_s^T e_s \quad (4.44)$$

Differentiating $V_s(t)$ and substituting the control law (4.25) and estimation laws (4.24) yields

$$\begin{aligned} \dot{V}_s &= \lambda e_V^T \left(k_p R_f^T e_P - R_z(\psi_p)^T R_z(\psi) R_c^b e_s \right) - k_p e_\lambda (L^T e_P) \\ &\quad - k_s \lambda e_s^T e_s + \lambda e_s^T R_c^{bT} R_z(\psi)^T R_z(\psi_p) (v_{pb} - \hat{v}_{pb}), \end{aligned} \quad (4.45)$$

where $L(t) = (v_f(t) - R_f(t) \hat{v}_p(t))$. Furthermore,

$$\begin{aligned} \dot{V}_s &= \lambda k_p e_V^T (-\lambda R_f^T R_f e_V + e_\lambda R_f^T L) - k_s \lambda e_s^T e_s \\ &\quad - k_p e_\lambda (-\lambda L^T R_f e_V + e_\lambda L^T L) \\ &\quad + \lambda e_s^T R_c^{bT} R_z(\psi)^T R_z(\psi_p) e_V - \lambda e_V^T R_z(\psi_p)^T R_z(\psi) R_c^b e_s \end{aligned} \quad (4.46)$$

Further solving (4.46), we get,

$$\dot{V}_s = -k_p e_p^T e_p - k_s \lambda e_s^T e_s \leq 0. \quad (4.47)$$

Since $\dot{V}_s$ is negative semi-definite, we use Barbalat's Lemma Slotine and Li (1991) to prove the convergence of the error variables. The relation (4.47) proves that the error system $e_s'(t) \in \mathcal{L}_\infty$ and $e_s(t), e_p(t) \in \mathcal{L}_2$. The bounded $e_\lambda(t)$ and $e_v(t)$ from equations (4.41) and (4.42), proves that the $\hat{\lambda}(t), \hat{v}_{pb}(t) \in \mathcal{L}_\infty$ for the bounded λ and v_{pb} . The signals $R_c^b, R_z(\psi(t)), R_z(\psi_p(t)) \in \text{SO}(3)$ are bounded by design and thus $\bar{v} \in \mathcal{L}_\infty$. Furthermore, the signals $\dot{e}(t), \dot{e}_p(t) \in \mathcal{L}_\infty$ as they are the function of the above-mentioned bounded quantities. Thus, using equation (4.47), we have $\dot{V}_s(t) \in \mathcal{L}_\infty$ and therefore $\dot{V}_s(t)$ is uniformly continuous. Using the fact that $V_s(t)$ has a lower bound and Barbalat's Lemma we have $\lim_{t \rightarrow \infty} \dot{V}_s(t) = 0$, which proves that $e_s(t), e_p(t) \rightarrow 0$ as $t \rightarrow \infty$. This proves asymptotic convergence of the feature tracking error $e_s(t)$ despite the unknown target motion and the parameter uncertainty in the desired feature depth. $\square$

Note that Theorem 4 only guarantees the asymptotic convergence of feature tracking error $e_s(t)$. Convergence of parameter estimation errors $e_v(t)$ and $e_\lambda(t)$ is not guaranteed; rather, it is proved that these quantities are bounded. Next, we present the stability and convergence properties of translational feature tracking errors and parameter estimation errors under the control and estimation laws of Method II.

Theorem 5. (Method II). *If Assumptions 5 and 6 hold, the error system $e_s'(t) = [e_s(t)^T, \tilde{\Theta}^T]^T$ exponentially converges to zero with the estimation and control laws defined in (4.33), (4.34) and (4.35).*

Proof. To prove the claim, let's consider the following Lyapunov candidate

$$V_s = \frac{1}{2} e_s^T e_s + \frac{1}{2} \tilde{\Theta}^T \Gamma^{-1} \tilde{\Theta}, \quad (4.48)$$

where $\tilde{\Theta} = \Theta - \hat{\Theta}$. Taking the time derivative of V along the error dynamics (4.10) yield

$$\dot{V}_s = e_s^T (-\lambda \bar{v} + R_c^b R_z(\psi)^T R_z(\psi_p) \lambda v_{pb}) + \tilde{\Theta}^T \Gamma^{-1} (-\dot{\tilde{\Theta}}). \quad (4.49)$$

Further simplifying equation (4.49), we obtain the following

$$\begin{aligned}\dot{V}_s &= e_s^T (\Phi^T \Theta \pm \Phi^T \hat{\Theta}) + \frac{1}{2} \tilde{\Theta}^T \Gamma^{-1} (-\dot{\hat{\Theta}}) \\ &= e_s^T (\Phi^T \tilde{\Theta} - \hat{\Theta}_1 \bar{v} + R_c^{bT} R_z(\psi)^T R_z(\psi_p) \hat{\Theta}') + \tilde{\Theta}^T \Gamma^{-1} (-\dot{\hat{\Theta}})\end{aligned}\quad (4.50)$$

Substituting control and estimation laws (4.33) - (4.35) in (4.50) and using the fact that $(\Theta_1 - \hat{\Theta}_1(t)) \left(\frac{\hat{\Theta}_1(t) - \lambda}{\varepsilon} \Delta_1 \right), (\Theta_1 - \hat{\Theta}_1(t)) \left(\frac{\bar{\lambda} - \hat{\Theta}_1(t)}{\varepsilon} \Delta_1 \right) \geq 0$, we get the following

$$\dot{V}_s = -k_s e_s^T e_s - \hat{\Theta}_1 e_s^T u_{ex} - \gamma_3 \tilde{\Theta}^T P \tilde{\Theta}. \quad (4.51)$$

Taking bound on $\dot{V}_s$ using the Cauchy inequality and Lemma 6 yields

$$\dot{V}_s \leq -\frac{k_s}{2} \|e_s\|^2 - \gamma_3 \alpha \Lambda \|\tilde{\Theta}\|^2 + \frac{\bar{\lambda}^2}{2k_s} \|u_{ex}\|^2. \quad (4.52)$$

Furthermore,

$$\dot{V}_s \leq -\beta_1 V_s + D_1, \quad (4.53)$$

where $\beta_1 = 2 \times \min\{\frac{k_s}{2}, \gamma_3 \alpha \Lambda\}$ and $D_1(t) = \frac{\bar{\lambda}^2}{2k_s} \|u_{ex}\|^2$. Since $u_{ex}(t)$ is an exponentially decaying signal, using the comparison lemma Khalil (2002), we can claim that $V_s(t)$ converges to zero exponentially. Hence $e_s'(t)$ exponentially converge to zero. $\square$

The stability and convergence analysis for rotational kinematics is presented next.

4.2.4 Stability and Convergence - Rotational Kinematics

We define the velocity estimation error $e_{\psi_p}(t)$ as

$$e_{\psi_p}(t) = \dot{\psi}_p(t) - \hat{\dot{\psi}}_p(t). \quad (4.54)$$

Thus, closed-loop estimation error dynamics with estimation law (4.40) becomes

$$\dot{e}_{\psi_p}(t) = \dot{\psi}_p(t) - \frac{1}{\gamma_2} e_{\psi_p}(t). \quad (4.55)$$

Theorem 6. Consider the control law defined in (4.39), the closed-loop error dynamics of $e'_\alpha(t) = [e_\alpha(t), e_{\psi_p}(t)]$ is globally uniformly ultimately bounded (UUB) for gains $k_\alpha > \frac{1}{2}$ and $\gamma_2 < 1$, if the target is having a time-varying angular velocity. Moreover, the closed-loop error dynamics of $e'_\alpha(t)$ is globally uniformly exponentially stable if $h_2 = 0$ (i.e. the angular motion of target is constant).

Proof. Consider the following Lyapunov candidate for the error system $e_\alpha(t)$

$$V_\psi = \frac{1}{2}e_\alpha^2 + \frac{1}{2}e_{\psi_p}^2. \quad (4.56)$$

Differentiating $V_\psi(t)$ and substituting control law (4.39) yields

$$\dot{V}_\psi = -k_\alpha e_\alpha^2 - \frac{1}{\gamma_2} e_{\psi_p}^2 + e_\alpha e_{\psi_p} + e_{\psi_p} \ddot{\psi}_p. \quad (4.57)$$

Given Assumption 5, we have $\sup_{t \geq 0} \|\ddot{\psi}_p(t)\| \leq \sup_{t \geq 0} \|\ddot{\psi}_p(t)\| \leq h_2$. The bounds on $\dot{V}_\psi(t)$ using the Cauchy inequality thus can be given as

$$\dot{V}_\psi \leq -(k_\alpha - \frac{1}{2})e_\alpha^2 - (\frac{1}{\gamma_2} - \frac{1}{2} - \frac{1}{2\gamma_2})e_{\psi_p}^2 + \gamma_2 h_2^2. \quad (4.58)$$

Furthermore, we can write

$$\dot{V}_\psi \leq -\beta V_\psi + D, \quad (4.59)$$

with $\beta = 2 \times \min\left(k_\alpha - \frac{1}{2}, \frac{1}{2\gamma_2} - \frac{1}{2}\right)$ and a tunable $D = \gamma_2 h_2^2$. The relation (4.59) invokes global uniformly ultimately bounded (UUB) stability (based on Theorem 4.18 in Khalil (2002)). Furthermore, if $h_2 = 0$, we yield the global uniform exponential stability as $D = 0$. $\square$

4.3 Results

This section presents the comprehensive evaluation of our proposed adaptive IBVS method through two distinct simulation environments: MATLAB simulations and SITL simulations. The MATLAB simulations offer a detailed examination of the theoretical

underpinnings of our approach, allowing us to model the system’s behavior precisely and showcase the academic findings with a high degree of accuracy. Conversely, the SITL simulations provide a more practical assessment by integrating our algorithm into a high-fidelity environment, enabling us to evaluate its adaptability and robustness in a sampled data implementation setting.

4.3.1 Simulation Environments

4.3.2 MATLAB Environment

In our MATLAB environment, we construct a scenario comprising a camera-equipped quadrotor and a target in a North-East-Down (NED) inertial frame with $R_c^b = I$. The camera model is configured with a focal length of 15 mm , a principal point located at $u_0 = v_0 = 512\text{ pixel}$, and pixel dimensions of $\rho_h = \rho_w = 10^{-5}\text{ m/pixel}$, operating at a resolution of 1024×1024 pixels. The target consists of four markers positioned initially at $P_1 = [-0.36, -0.25, -1]^T\text{ m}$, $P_2 = [0.36, -0.25, -1]^T\text{ m}$, $P_3 = [0.36, 0.25, -1]^T\text{ m}$, and $P_4 = [-0.36, 0.25, -1]^T\text{ m}$, specified in the inertial frame. We consider the desired features that enable the quadrotor to keep the moving target’s center at the center of the image plane while maintaining a constant height of 4 m (i.e., $\lambda = 0.25\text{ m}^{-1}$) above it.

4.3.3 SITL Environment

To test the efficacy of the proposed scheme in a realistic scenario, we consider the case of a quadrotor tracking a moving ground rover in a robot operating system (ROS) Stanford Artificial Intelligence Laboratory *et al.* (2018) integrated GAZEBO Koenig and Howard (2004) environment as shown for Method I in Fig. 4.1 and Method II in 4.2. We create an environment consisting of a Hector quadrotor with a camera Meyer *et al.* (2012) and an AprilTag-equipped Husky ground rover Clearpath Robotics (n.d.). The corners of the AprilTag are used as visual features to implement the proposed IBVS scheme using ROS nodes. The camera’s intrinsic parameters are considered unknown and obtained through calibration techniques. For SITL simulations, the inertial frame is right-handed ENU i.e. $R_c^b = R_x(\pi)R_z(\frac{\pi}{2})$. The initial position of the quadrotor in the inertial frame is set to $[1.4, 1.5, 6]^T\text{ m}$ with a yaw of $\pi\text{ rad}$. The desired features are

calculated by placing the quadrotor at an inertial height of $3m$ from the ground and the relative heading of π rad to the ground rover.

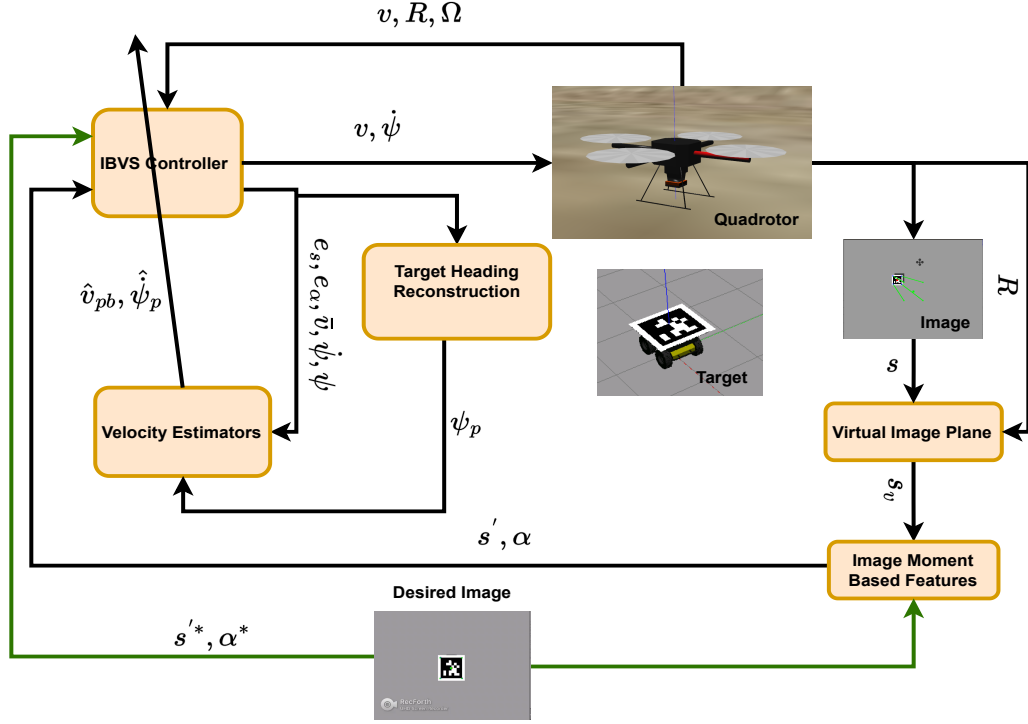


Figure 4.1: Control architecture and the simulation environment flow chart for Method-I

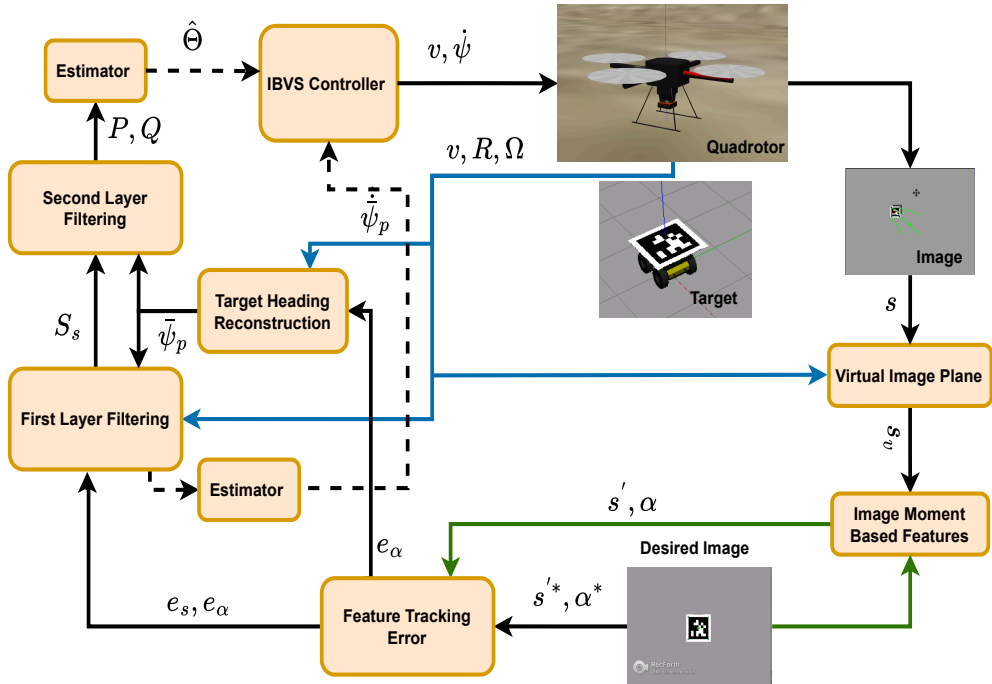


Figure 4.2: Control architecture and the simulation environment flow chart for Method-II

4.3.4 Discretization

Efficient discretization is essential during the real-time implementation of controllers with differential components to replicate the theoretical establishments in sampled data settings. The proposed control laws involve both linear (filters (4.13) - (4.16) and (4.37) - (4.38)) and nonlinear (equations (4.24), (4.29), (4.30), (4.34), and (4.35)) differential structures. The linear filters of form $\dot{y}(t) = -\Upsilon y(t) + \Upsilon u(t)$, are discretized leveraging the linear system theory as

$$y(t+T) = \exp(-\Upsilon T)y(t) + (1 - \exp(-\Upsilon T))u(t). \quad (4.60)$$

where T is the sampling time. We use the Euler implicit discretization (Butcher, 2016, Chapter 2) for the nonlinear differential structures. This yields the following discretization for Method I

$$\begin{aligned} \hat{v}_{pb}(t+T) = & \hat{v}_{pb}(t) + T \left(-k_p R_f^T(t+T) e_P(t+T) \right) \\ & + T \left(R_z(\psi_p(t+T))^T R_z(\psi(t+T)) R_c^b e_s(t+T) \right) \end{aligned} \quad (4.61)$$

$$\begin{aligned} \hat{\lambda}(t+T) = & \hat{\lambda}(t) + T v_f^T(t+T) e_P(t+T) \\ & - T \hat{v}_{pb}(t+T) R_f^T(t+T) e_P(t+T), \end{aligned} \quad (4.62)$$

where $e_P(t+T) = S_s(t+T) - \hat{S}_s(t+T)$. To solve the above set of nonlinear equations for $\hat{v}_{pb}(t+T)$ and $\hat{\lambda}(t+T)$, we use python SciPy *fsolve*. Similarly, the discretization of nonlinear filter structures in Method II yields the following

$$P(t+T) = P(t) + T \alpha(t+T) \Phi_f(t+T) \Phi_f(t+T)^T, \quad (4.63)$$

$$Q(t+T) = Q(t) + T \alpha(t+T) \Phi_f(t+T) S_s(t+T), \quad (4.64)$$

$$\hat{\Theta}(t+T) = \frac{(\hat{\Theta}(t) + T \Gamma \Phi(t+T) e_s(t+T) + \Gamma \gamma_2 Q(t+T))}{I_4 + T \Gamma \gamma_2 P(t+T)} \quad (4.65)$$

Moreover, a saturation function is used for the discrete implementation of the *Proj()* operator further upon the resulting scalar solution of $\hat{\Theta}_1(t+T)$ from equation (4.65). We next present the MATLAB simulation results using Method I and II.

4.3.5 MATLAB Simulations-Method I

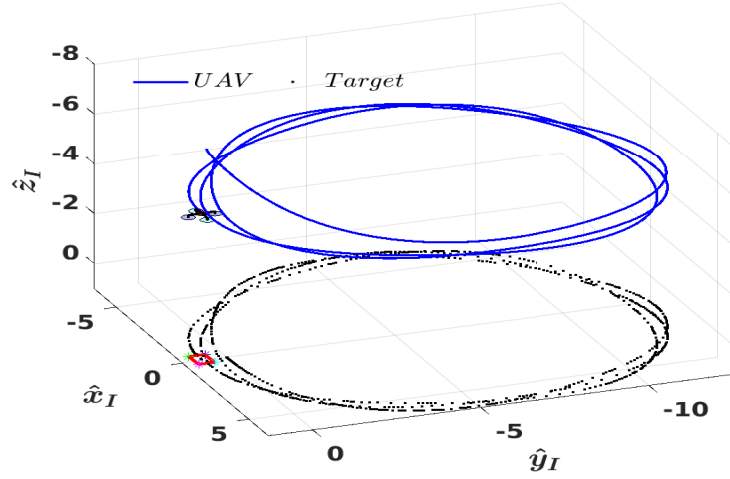
We consider, the quadrotor positioned at $[0, 0, -8]^T m$ with angular orientation (roll, pitch, yaw) of $[0, 0, 0.6]^T rad$, also expressed in the inertial frame. We use the following controller and observer parameters: $k_s = 2$, $k_\alpha = 2$, $k_p = 20$, $\gamma_1 = 10^{-2}$, $\gamma_2 = 10^{-2}$. We simulate with $v_{pb} = [2, 0, 0]^T m/sec$ and $\psi_p(t) = \frac{\pi}{20} \cos(t) - \frac{\pi}{10} rad/sec$ and the simulation results are presented in Fig. 4.3. Fig. 4.3a represents the trajectory of the quadrotor and the target. From Fig. 4.3b, it can be observed that the tracking error $e_s(t)$ and the predictor model error $e_P(t)$ are converging to zeros asymptotically as proven in Theorem 4 despite the unknown depth and the target linear velocity parameters. Further, the tracking error $e_\alpha(t)$ and angular velocity estimation error $e_{\psi_p}(t)$ are plotted in Fig. 4.3d, which shows their convergence to a small ultimate bound around the origin as the angular velocity of the target is time-varying. The plots for bounded quantities $\lambda(t)$ and $\hat{v}_{pb}(t)$ are presented in Fig. 4.3c, showing that $\hat{v}_{pb}$ converging to true values of v_{pb} . However, $\hat{\lambda}$ converges to a value different from the true λ . Note that the results in Theorem 4 don't guarantee the convergence of these parameters to their true values.

4.3.6 MATLAB Simulation-Method II

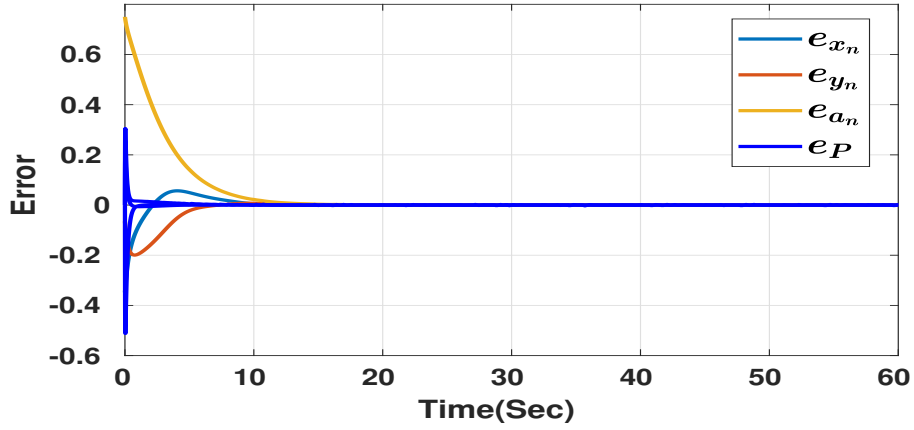
For these simulations, we consider the quadrotor to be positioned at $[1, 1, -8]^T m$ with angular orientation (roll, pitch, yaw) of $[0, 0, 0.6]^T rad$, expressed in the inertial frame. The controller and observer design parameters are selected as follows: $k_s = k_\alpha = 2$, $\Gamma = 20I_4$, $\gamma_1 = 10^{-2}$, $\gamma_2 = 0.3$, $\gamma_3 = 10^{-2}$, $\lambda = 0.1$, $\varepsilon = 0.01$. The control input $\bar{v}$ is supplied with a probing signal $u_{ex} = \frac{1}{5} \exp(-0.4t) [1, \sin(3t), \cos(3t)]^T$.

For the simulation, the desired height that the quadrotor needs to maintain above the target is $4m$ (i.e., $\lambda = 0.25m^{-1}$) while $v_{pb} = [5, 0, 0]^T m/sec$ and $\psi_p(t) = \frac{\pi}{20} \cos(t) - \frac{\pi}{10} rad/sec$. The simulation results with the Method II IBVS scheme are presented in Fig. 4.4. Plots 4.4a and 4.4b show the trajectory plot of both the vehicles in the inertial frame $\{I\}$ and the feature tracking error, respectively. It can be seen that the feature tracking error is converging to the origin, indicating that the control objective is achieved. Further, the parameter estimation error plot given in Fig. 4.4c shows the convergence of the parameters to the origin.

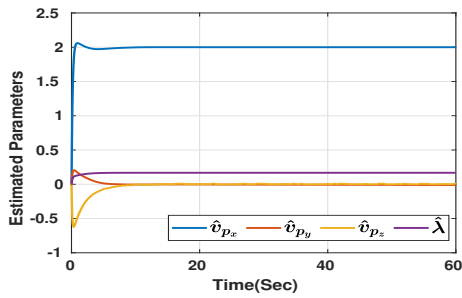
To evaluate the impact of parameter convergence on the transient performance of



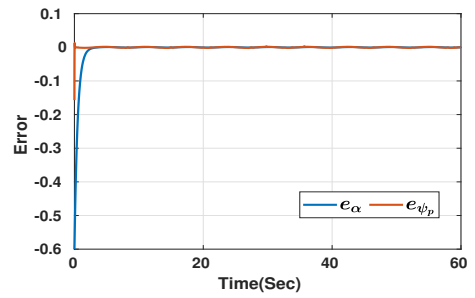
(a) Trajectory plot



(b) Tracking error ($e_s(t), e_P(t)$) plot



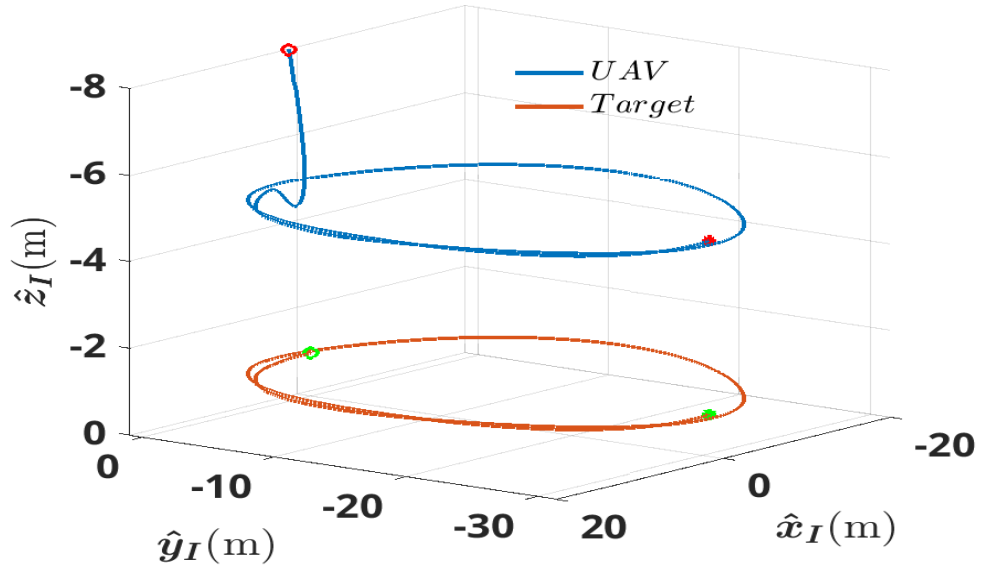
(c) Parameter estimates ($\hat{v}_{pb}(t), \hat{\lambda}(t)$) plot



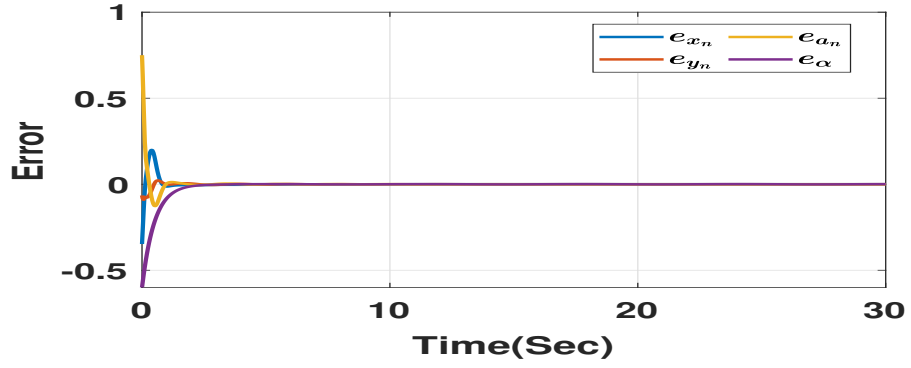
(d) Error plot for $e_{\alpha}(t)$ and $e_{\psi_p}(t)$

Figure 4.3: MATLAB simulation results considering translational kinematics controller as given in Method I.

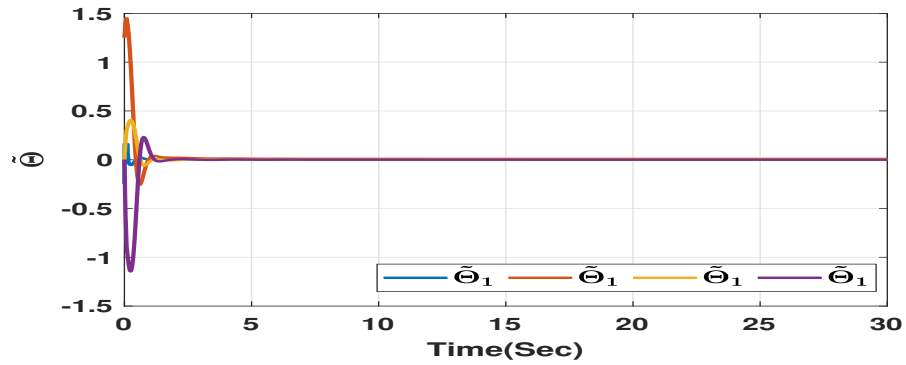
feature tracking error, we compare the MATLAB simulation results using control Method I and Method II. Note that the control design parameters and other common filter parameters are kept the same while simulating for both controllers. The comparative results are illustrated in Fig. 4.5. Fig. 4.5a highlights the successful tracking of the target by



(a) Trajectory plot (○ and ● represent the initial and final positions of the vehicles, respectively)



(b) Feature tracking error plot (e_s, e_α)

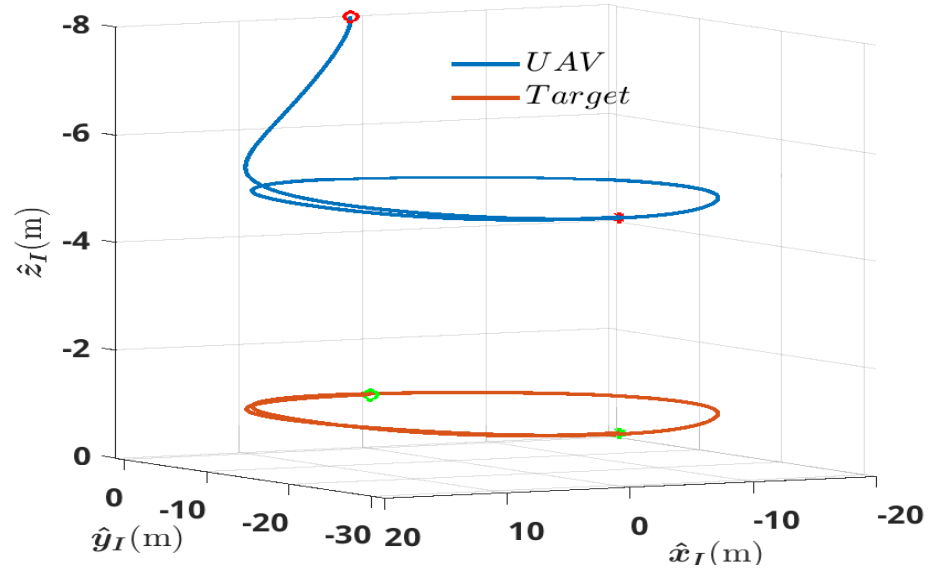


(c) Parameter estimation error ($\tilde{\Theta}$)

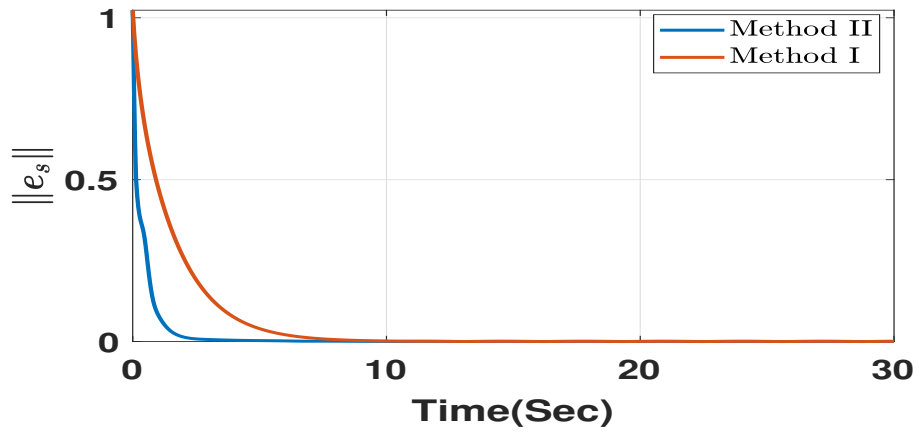
Figure 4.4: MATLAB simulation results considering translational kinematics controller as given in Method II.

the quadrotor from the desired height with the controller given in Method I. As shown in Fig. 4.5b, both control schemes achieve feature tracking error convergence; however, the proposed controller exhibits improved transient performance. This improvement is attributed to parameter convergence, as demonstrated in Fig. 4.5c. Next, we test the

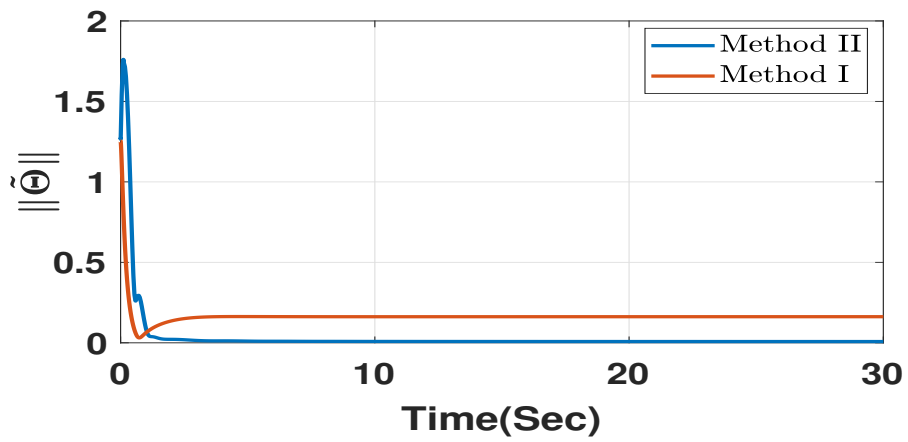
proposed scheme with high-fidelity SITL simulations.



(a) Trajectory plot with controller given in Method I



(b) Feature tracking error norm plot



(c) Plot for $\|\hat{\Theta}\|$

Figure 4.5: Comparative MATLAB simulation results with the controller given in Method I and Method II

4.3.7 SITL Simulations-Method I

We use the following controller and observer parameters for Method I: $k_s = 0.40$, $k_\alpha = 0.40$, $k_p = 20$, $\gamma_1 = 0.6$, $\gamma_2 = 0.6$ for the results presented in this section. The SITL simulation results for two scenarios utilizing the translational kinematics controller from Method I are presented next.

Scenario I

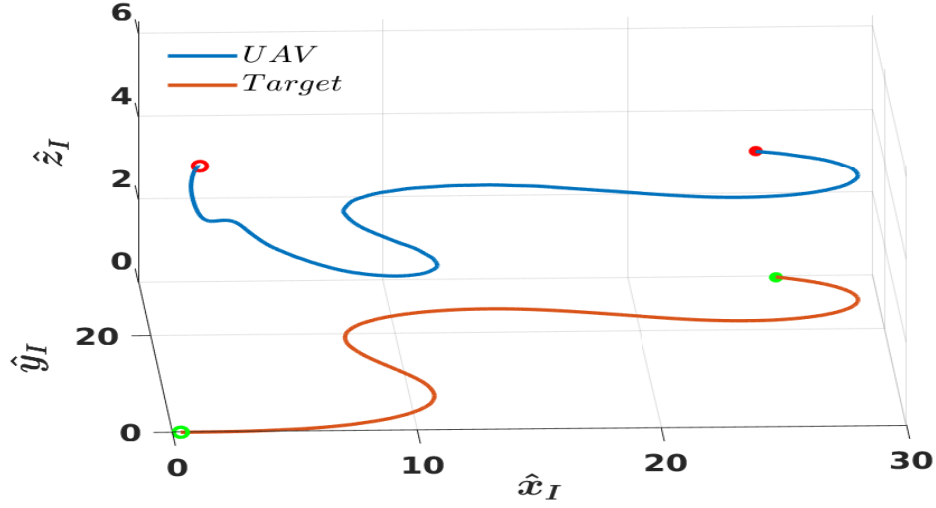
First, we consider that the ground rover is moving with a constant body-fixed linear velocity of magnitude $0.7m/sec$ (i.e. $v_{pb} = [0.7, 0, 0]^T m/sec$) and an angular velocity $\psi_p(t) = 0.2\sin(\frac{t}{10}) rad/sec$. Fig. 4.6a and 4.6b shows the trajectory of the quadrotor and target and feature tracking error, respectively. The results show the convergence of feature tracking errors, proving that the quadrotor successfully tracks the target from the desired height in the task space. Fig. 4.6d and 4.6c show the bounded depth parameter estimate and target velocity estimates. Moreover, the estimates for angular velocity $\hat{\psi}_p$ as plotted in 4.6c that matches the actual profile of $\psi_p(t)$.

Scenario II

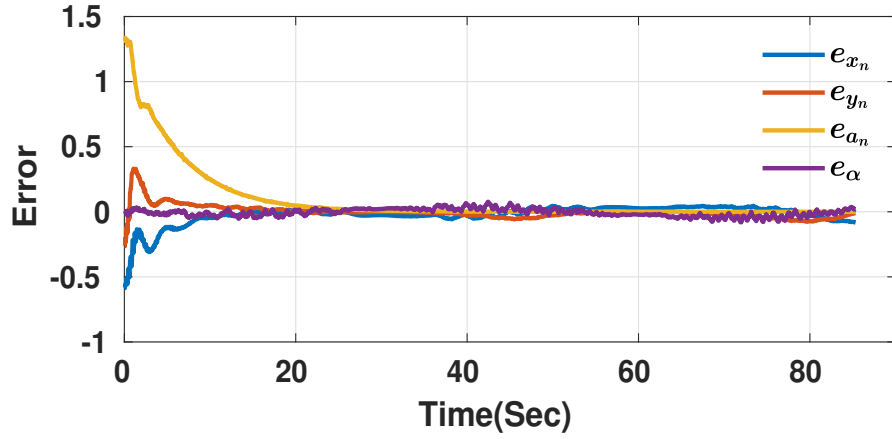
We consider another scenario, with the following piecewise constant linear and angular velocity profile of the target, to test its adaptations in such scenarios

$$[v_{pb}(t), \psi_p(t)] = \begin{cases} [[0.7, 0, 0]^T, \frac{\pi}{15}], & \text{if } 10 < t \leq 25, \\ [[0.7, 0, 0]^T, \frac{\pi}{15}], & \text{if } 35 < t \leq 47, \\ [[0.7, 0, 0]^T, -\frac{\pi}{15}], & \text{if } 57 < t \leq 69, \\ [[0.5, 0, 0]^T, 0], & \text{otherwise} \end{cases}.$$

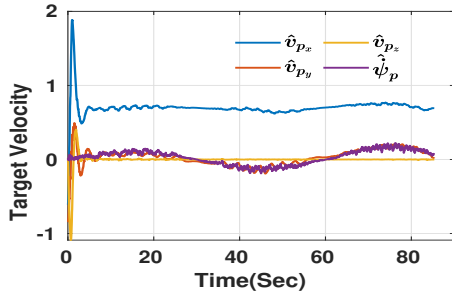
Fig. 4.7a and 4.7b shows the trajectory of the quadrotor and target and feature tracking error, respectively. From the results, it's evident that the quadrotor can track the target successfully while maintaining the desired heading with respect to the target for the given velocity profile of the target whose first derivative is not bounded at the points of switching. Fig. 4.7d and 4.7c show the bounded depth parameter estimate and



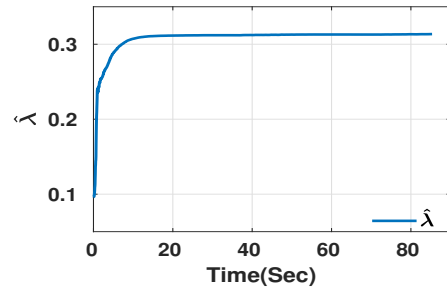
(a) Trajectory plot (○ and ● represent the initial and final positions of the vehicles, respectively)



(b) Feature tracking error (e_s, e_{α})



(c) Velocity estimates



(d) Depth parameter estimate

Figure 4.6: GAZEBO simulation results for Scenario I

target velocity estimates. The results indicate that the controller in Method I can handle piecewise constant parameterized target motion profiles. Further analysis along these lines is left as a subject for future work.

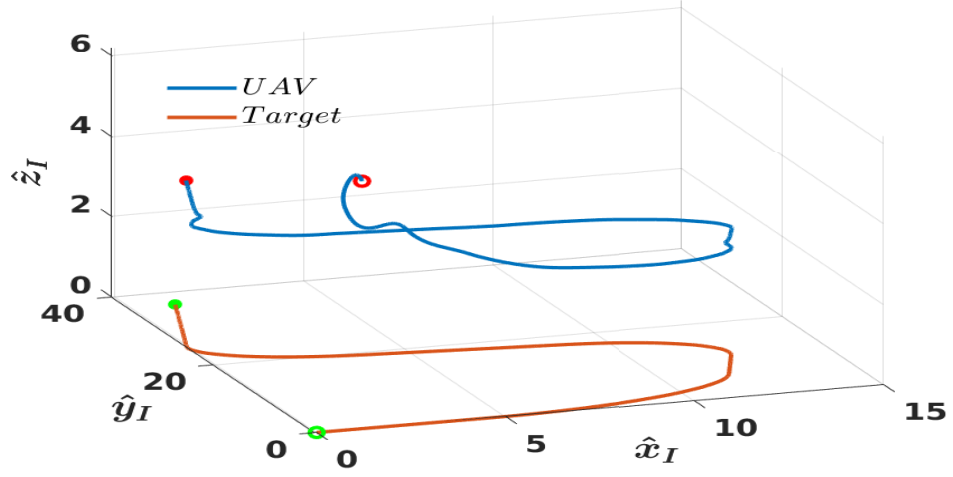
4.3.8 SITL Simulations-Method II

We choose the following controller and observer parameters: $k_s = 0.4$, $k_\alpha = 0.4$, $\Gamma = 20I_4$, $\gamma_1 = 0.6$, $\gamma_2 = 0.6$, $\gamma_3 = 0.7$, $\lambda = 0.1$, $\varepsilon = 0.01$ for the results presented in this section. We consider that the ground rover is moving with a constant body-fixed linear velocity of magnitude $0.7m/sec$ (i.e. $v_{pb} = [0.7, 0, 0]^T m/sec$) and an angular velocity $\dot{\psi}_p(t) = 0.2\sin(\frac{t}{10}) rad/sec$. We use the probing signal $u_{ex} = \frac{3}{10}\exp(-0.1t)[0.1\cos(30t), 0.2\sin(10t), 0.3\cos(15t)]^T$

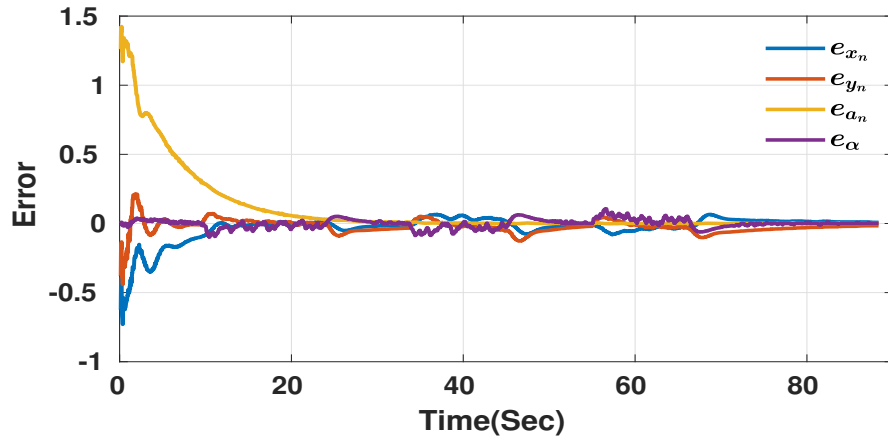
The results with the Method II controller are presented in Fig. 4.8. Fig. 4.8a and 4.8b show the trajectory of the quadrotor and target and feature tracking error, respectively. The results show the convergence of feature tracking errors, proving that the quadrotor successfully tracks the target from the desired height in the task space. Further, Fig. 4.8c presents the estimated parameters which remain bounded. The comparative results with Method I controller are shown in Fig. 4.9. The trajectory plot 4.9a of quadrotor and target, along with the feature tracking error, validates that the control objective is achieved with the Method I controller also. However, the feature tracking error norm shows that the proposed controller makes the transient faster. A video containing the SITL simulation results for both methods can be found here - [Video](#).

4.4 Conclusion

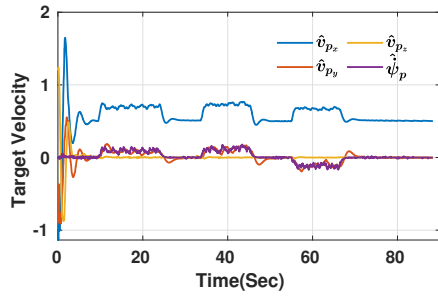
Chapter 4 addressed the critical challenge of tracking planar, non-holonomic moving targets with unknown desired feature depth in IBVS frameworks by introducing adaptive kinematic controllers for quadrotor tracking. Two control methods were proposed: a bilinear estimator-augmented adaptive IBVS controller ensuring feature tracking error convergence and an adaptive IBVS controller that achieved both feature tracking and parameter convergence under the less restrictive initial excitation (IE) condition. These methods decoupled the translational and angular motions of the quadrotor using



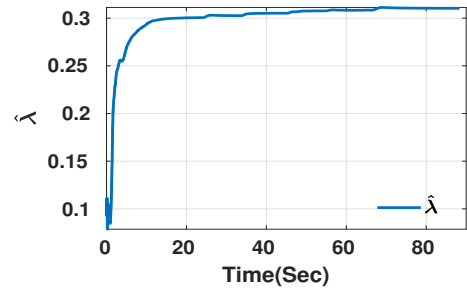
(a) Trajectory plot



(b) Feature tracking error (e_s, e_{α})

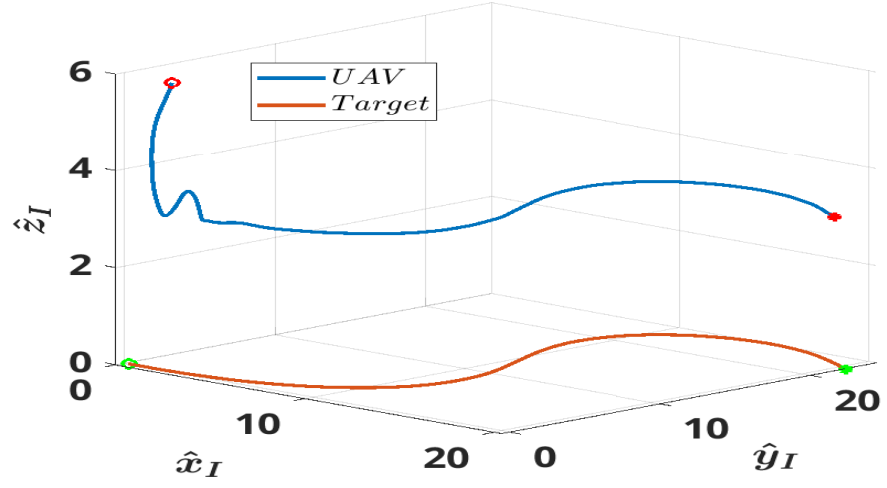


(c) Velocity estimates

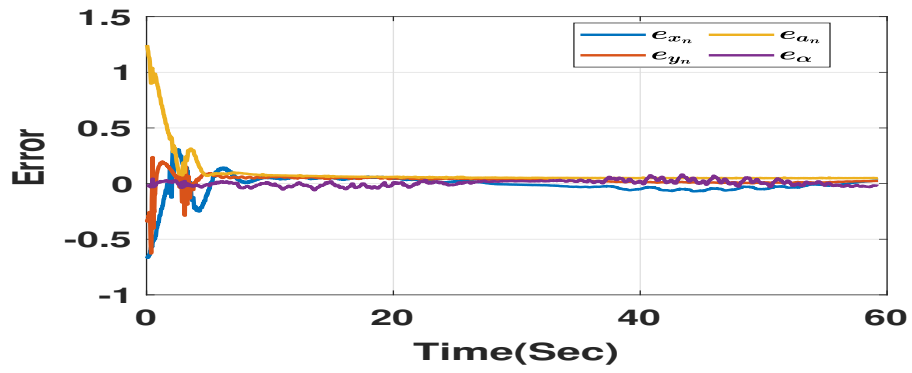


(d) Depth parameter estimate

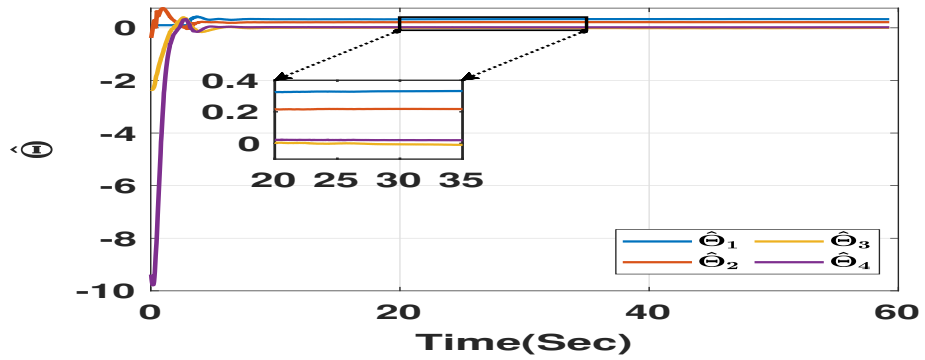
Figure 4.7: GAZEBO simulation results for Scenario II



(a) Trajectory plot

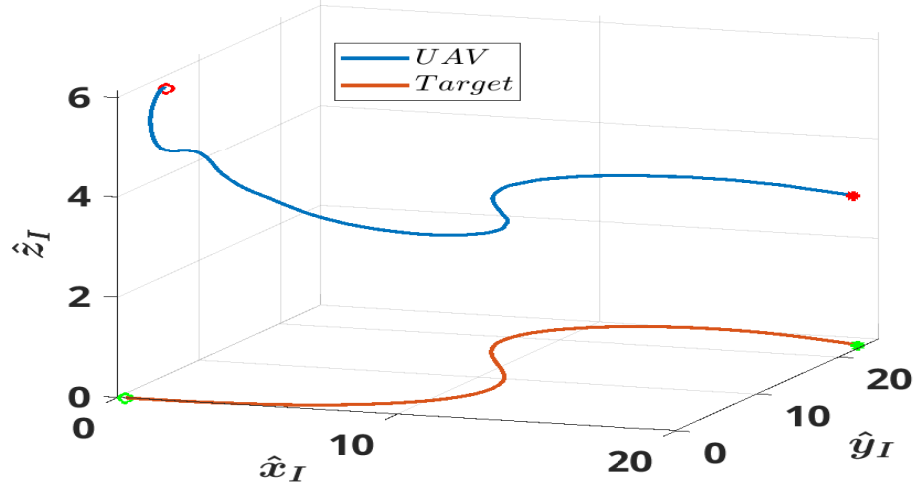


(b) Feature tracking error plot (e_s, e_{α})

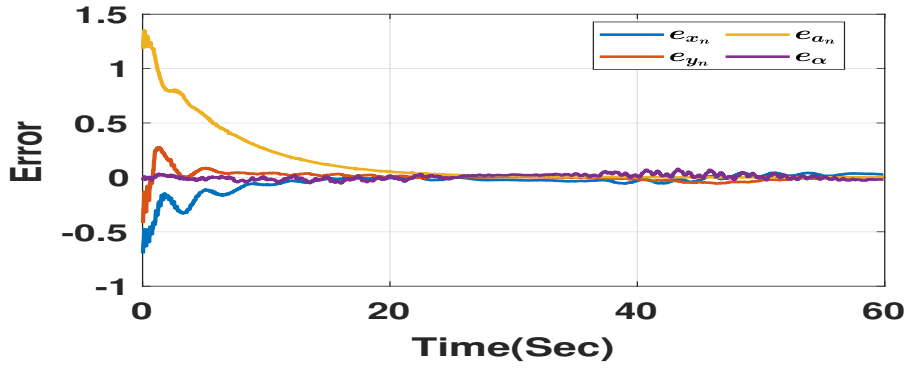


(c) Estimated parameters plot

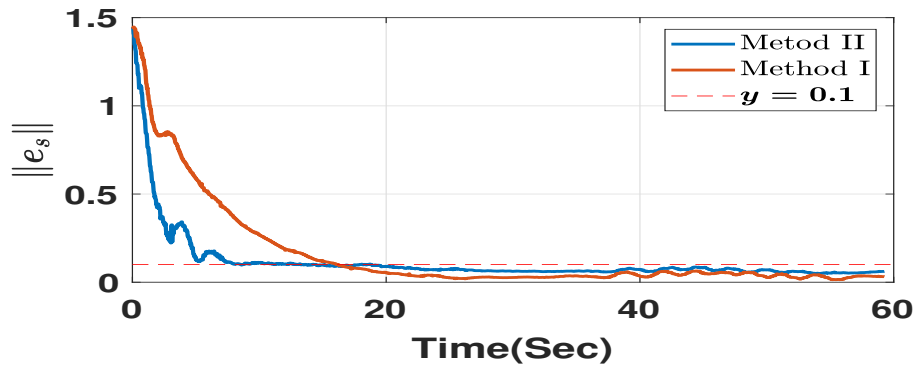
Figure 4.8: SITL simulation results with the Method II controller



(a) Trajectory plot with Method I controller



(b) Feature tracking error plot with Method I controller



(c) Feature tracking error norm comparison

Figure 4.9: Comparative SITL simulation results with the controller given in Method I and II

virtual image plane-based features and effectively reconstructed the target's heading to address unknown translational velocity. Lyapunov-based analysis showed asymptotic convergence of translational feature error dynamics for Method I and exponential convergence for Method II, with both methods ensuring uniformly ultimately bounded rotational error dynamics. Simulation results validated the effectiveness of both methods, with Method II outperforming Method I by achieving parameter convergence and improved feature tracking performance. This chapter successfully extended the IBVS framework, making it more robust and adaptable for dynamic and uncertain environments.

Chapter 5

Partially Known Trajectory Tracking Control for Quadrotors

This chapter addresses the general problem of designing a trajectory-tracking controller for the multi-rotor UAV when only the desired position and heading are available. Building upon the state-of-the-art geometric controller proposed in Lee *et al.* (2010); Lee (2012), we develop a robust solution tailored for partially known trajectories. Developed on the special Euclidean group $SE(3)$, the classical geometric controller enables us to design an almost global trajectory tracking controller for multi-rotor UAVs. However, two major limitations of this controller are (i) the thrust is not saturated, hence no guarantee of the existence of the desired angular orientation, and (ii) the designed thrust can be negative when the angle between the third body axis of a multi-rotor UAV and desired thrust direction is greater than 90° Sharma and Kar (2021). Our proposed solution addresses these limitations in the current context of designing the controller for partially known trajectories.

The main contributions of this chapter are as follows

1. We consider the problem of multi-rotor tracking a partially known trajectory and develop a trajectory-tracking controller by suitably modifying the state-of-the-art geometric controller Lee *et al.* (2010).
2. Unlike Emran and Najjaran (2021), an auxiliary filter dynamics in $SO(3)$ space is proposed for the UAV attitude system, which tackles the requirement of higher-order derivatives of the desired position in desired attitude derivatives. A similar approach using a command filter is presented in Zhao *et al.* (2013); however, considering a quaternion-based formulation. To the best of our knowledge, this is the first time the consequences of such filter dynamics for attitude representations in $SO(3)$ are quantified on the overall system through a Lyapunov-based analysis.
3. To ensure the non-zero desired thrust requirements for the existence of the desired attitude Jabbari Asl (2019); Dasgupta *et al.* (2019), we propose a projection-based filter dynamics to yield a bounded second-order derivative of the desired position. Further, we discuss the effect of large attitude errors on the feasibility of the thrust control inputs proposed in Lee *et al.* (2010); Sharma and Kar (2021) in detail and introduce a novel control strategy.

4. The effect of the proposed control scheme, along with the auxiliary filter dynamics, is theoretically analyzed. The overall system, consisting of translational, angular, and auxiliary dynamics, is shown to be uniformly ultimately bounded through rigorous mathematical arguments. The careful design and extensive theoretical analysis enable us to achieve an ultimate bound that can be made arbitrarily small by choosing high enough auxiliary dynamics gains without burdening the controller gains.

5.1 Problem Formulation and Preliminaries

5.1.1 Notations

The map $(\hat{\cdot}) : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$ is the skew-symmetric map whereas its inverse is denoted by $(\cdot)^\vee : \mathfrak{so}(3) \rightarrow \mathbb{R}^3$. For a vector v , $\|v\|$ denotes the 2-norm and for a matrix M , $\|M\|$ denotes the 2-norm until unless specified. $\ln(\cdot)$ represents the natural logarithmic operator. For a vector $v = [v_1, v_2, v_3]^T$,

$$\tanh(v) = [\tanh(v_1), \tanh(v_2), \tanh(v_3)]^T \in \mathbb{R}^3, \quad (5.1)$$

$$\text{Cosh}(v) = \text{daig}\{\cosh(v_1), \cosh(v_2), \cosh(v_3)\} \in \mathbb{R}^{3 \times 3}, \quad (5.2)$$

$$\text{Tanh}(v) = \text{daig}\{\tanh(v_1), \tanh(v_2), \tanh(v_3)\} \in \mathbb{R}^{3 \times 3}. \quad (5.3)$$

$$\text{Sech}(v) = \text{Cosh}^{-1}(v). \quad (5.4)$$

5.1.2 Multi-rotor Dynamics

Consider a multi-rotor UAV illustrated in Figure 5.1. Let $\{I\}$ and $\{B\}$ be the inertial and the vehicle body frames, with origins O_I and O_B , respectively. The configuration of UAV is defined by the position $x \in \mathbb{R}^3$ of the center of mass and attitude $R = [x_B, y_B, z_B] \in \text{SO}(3)$ in the inertial frame.

Let m denote the mass of quadrotor, J denote the inertia matrix expressed in $\{B\}$, and g denote the gravitational acceleration. We model the multi-rotor dynamics as

$$\dot{x} = v, \quad (5.5)$$

$$m\dot{v} = mge_3 - fRe_3, \quad (5.6)$$

$$\dot{R} = R\hat{\Omega}, \quad (5.7)$$

$$J\dot{\Omega} = -\Omega \times J\Omega + M, \quad (5.8)$$

where inertial frame position $x(t) \in \mathbb{R}^3$, inertial frame velocity $v(t) \in \mathbb{R}^3$, the rotation matrix from $\{B\}$ to $\{I\}$, $R(t) \in SO(3)$ and the body-fixed angular velocity $\Omega(t) \in \mathbb{R}^3$ represents the multi-rotor states. The total thrust $f \in \mathbb{R}$ and moment $M = [M_1, M_2, M_3]^T \in \mathbb{R}^3$ expressed in the body frame are considered as the control inputs to the multi-rotor model. Let $\omega_i, \forall i = 1, 2, \dots, n$ denote the rotor speed for a multi-rotor with n rotors and $e_3 = [0, 0, 1]^T$. For a given pair of total thrust and moment, the rotor speeds can be calculated using the following relation

$$\begin{bmatrix} f \\ M \end{bmatrix} = \Gamma \omega_T, \quad \omega_T = [\omega_1^2, \dots, \omega_n^2]^T, \quad (5.9)$$

where $\Gamma \in \mathbb{R}^{4 \times n}$ is a constant mixing matrix constructed based on the rotor configuration parameters such as placement of rotors, thrust, and moment coefficient of rotors, etc. For a comprehensive understanding of multirotor dynamics, readers are recommended to delve into the detailed discussions provided in references such as Lee *et al.* (2010); Mahony *et al.* (2012).

5.1.3 Control Objective

The objective is to design a nonlinear trajectory tracking controller for a multi-rotor UAV to track a position reference $x_d(t)$ and a heading direction $x_{B_d}(t)$ or a heading angle $\psi_d(t)$, where $x_{B_d}(t) = [\cos(\psi_d(t)), \sin(\psi_d(t)), 0]^T$. Moreover, we consider that the higher derivatives of the position reference and heading direction (i.e. $\frac{d^n x_d(t)}{dt^n}$ and $\frac{d^n x_{B_d}(t)}{dt^n}$, $n \geq 1$) are not available, which are essential for an accurate and efficient trajectory tracking controller.

Remark 7. In real-world applications of UAVs, various scenarios present challenges in

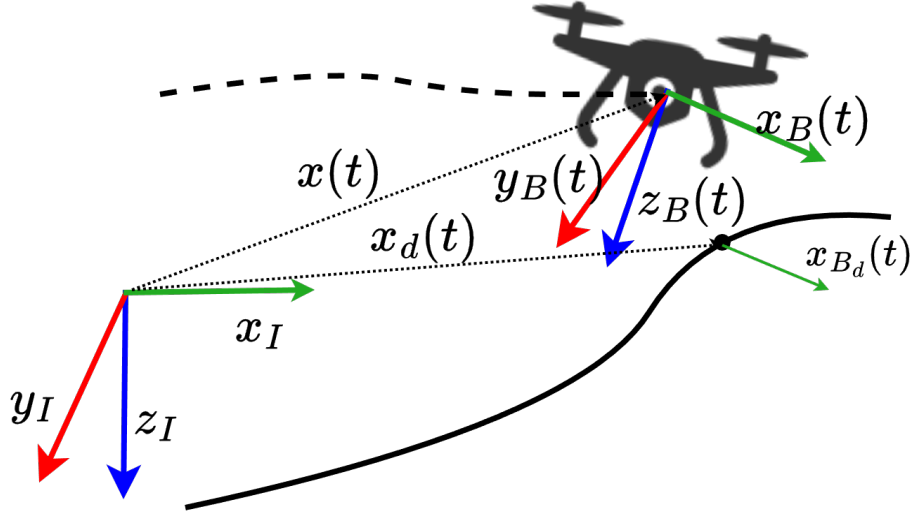


Figure 5.1: Coordinate frames illustration. The x, y, z axes of a coordinate frame $\{A\}$ are denoted by x_A, y_A and z_A , respectively.

obtaining higher derivatives of the desired trajectory, including velocity, acceleration, or jerk. One such scenario arises when UAVs track moving objects exhibiting unpredictable dynamics, such as wildlife or noncooperative vehicles. An example of this can be found in B  a et al. (2019), where a Kalman filter is used with restrictive assumptions of the target's motion to generate the higher derivatives of trajectory. In these instances, acquiring accurate and timely information regarding the target's motion beyond its position and orientation poses a significant challenge due to their unknown dynamics. Additionally, in collaborative missions requiring UAVs to maintain formation Oh et al. (2015) or execute coordinated maneuvers with other UAVs, obstacles like delays in data exchange, sensor limitations, or security concerns, particularly in military or surveillance operations, can hinder access to the higher derivatives of reference trajectories.

Assumption 7. We assume that the reference trajectory has bounded derivatives as

$$\sup_{t \geq 0} \left\| \frac{d^n x_d(t)}{dt^n} \right\| = h_n, \quad n = 1, 2, 3, 4, \quad \sup_{t \geq 0} \left\| \frac{dx_{B_d}(t)}{dt} \right\| = h_5, \quad \sup_{t \geq 0} \left\| \frac{d^2 x_{B_d}(t)}{dt^2} \right\| = h_6. \quad (5.10)$$

where $h_1, h_2, h_3, h_4, h_5, h_6 \in \mathbb{R}_+$ are finite constants.

5.2 Control Design

We follow a multistage procedure illustrated in Figure 5.2 to design the trajectory tracking controller. We first design the position tracking controller to obtain the control input $f(t)$. Subsequently, a desired rotation matrix $R_c(t) \in \text{SO}(3)$ constructed exploiting unit vector direction of f and desired heading direction $x_{B_d}(t)$. We then pass $R_c(t)$ through an auxiliary filter dynamics to obtain the filtered desired attitude reference, which the proposed attitude tracking controller would track.

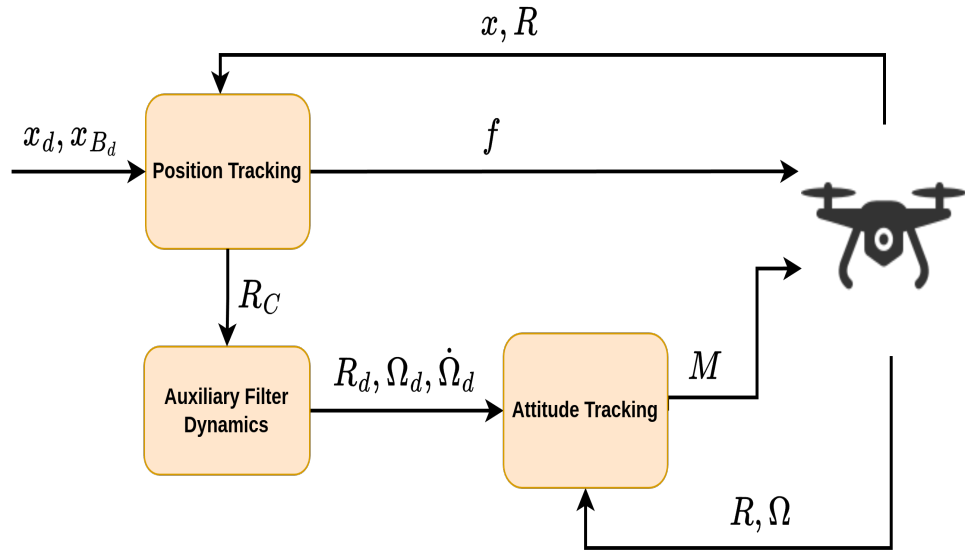


Figure 5.2: Control architecture

5.2.1 Position Tracking Controller

To design a smooth, saturated position tracking controller, we consider an error system Dasgupta *et al.* (2019) with position tracking error $e_x(t)$ and filter tracking error $e_\alpha(t)$ as

$$e_x = x - x_d = [e_{x_1}, e_{x_2}, e_{x_3}]^T, \quad (5.11)$$

$$e_\alpha = \dot{e}_x + \alpha_x \tanh(e_x) + \tanh(e_f), \quad (5.12)$$

where $e_f(t) = [e_{f_x}(t), e_{f_y}(t), e_{f_z}(t)]^T$ is an auxiliary filter variable with the dynamics

$$\dot{e}_f = \text{Cosh}^2(e_f) \left(-k_\alpha e_\alpha + \tanh(e_x) - \alpha_f \tanh(e_f) \right), \quad (5.13)$$

and $\{k_\alpha, \alpha_f, \alpha_x \in \mathbb{R}_+\}$ are positive constant control design parameters.

The thrust control input $f(t)$ is designed as

$$f = \|f_d\| \sqrt{\frac{1 + e_3^T R_c^T R e_3}{2}} \quad (5.14)$$

where $f_d(t)$ is a virtual control input defined as

$$f_d = mge_3 - mg_1 + 2m \tanh(e_x) - mk_\alpha \tanh(e_f), \quad (5.15)$$

and $g_1(t)$ is an estimate of $\ddot{x}_d(t)$, which is not directly available rather estimated using filters proposed in the subsequent section. The careful construction of the desired rotation matrix $R_c(t)$ and an auxiliary attitude dynamics is discussed in the section subsequent to the estimate of $\ddot{x}_d(t)$. Note that the thrust expression requires term $\dot{e}_x$, i.e., velocity feedback, in the e_f . Since we don't have the explicit knowledge of $\dot{x}_d$, one can circumvent the requirement of unknown term $\dot{e}_x$ in thrust input calculation following the output-feedback formulation in Zhang *et al.* (2000). This also makes the position dynamics loop output feedback.

5.2.2 Estimate of $\ddot{x}_d(t)$

To obtain the smooth and bounded estimate $g_1(t)$ of $\ddot{x}_d(t)$, we start by considering the following filter structure with filter input $x_d(t)$ and output $x_{fd}(t)$ with filter design parameter $\gamma_1 \in \mathbb{R}_+$

$$\gamma_1 \dot{x}_{fd} + x_{fd} = x_d, \quad x_{fd}(0) = 0. \quad (5.16)$$

Consider another filter for the unknown variable $\dot{x}_d(t)$ as

$$\gamma_1 \dot{g}_{x_d} + g_{x_d} = \dot{x}_d, \quad g_{x_d}(0) = 0. \quad (5.17)$$

Integrating (5.17) for $g_{x_d}(t)$ yield

$$g_{x_d}(t) = \frac{1}{\gamma_1} \left(x_d(t) - x_{fd}(t) - \exp\left(-\frac{1}{\gamma_1}t\right)x_d(0) \right). \quad (5.18)$$

Since $g_{x_d}(t)$ represents the filtered output of unknown variable $\dot{x}_d(t)$, we further pass it through a filter with design parameter $\gamma_2 \in \mathbb{R}_+$ as

$$\gamma_2 \dot{g}_{f_d} + g_{f_d} = g_{x_d}, \quad g_{f_d}(0) = 0, \quad (5.19)$$

to obtain the estimate of $\ddot{x}_d(t)$. Further following the filter structure similar to (5.17) for the unknown variable $\dot{g}_{x_d}$ and integrating it for its filter variable $g_{v_d}(t)$, we get

$$g_{v_d}(t) = \frac{1}{\gamma_2} \left(g_{x_d}(t) - g_{f_d}(t) - \exp\left(-\frac{1}{\gamma_2}t\right)g_{x_d}(0) \right). \quad (5.20)$$

For clarity, the filter structures discussed above are illustrated in Figure 5.3. Filter block 1 includes structures given by equations (5.16) and (5.18), with input $x_d(t)$ and output signal $g_{x_d}(t)$. Similarly, filter block 2 takes $g_{x_d}(t)$ as input and provides $g_{v_d}(t)$ using equations (5.19) and (5.20).

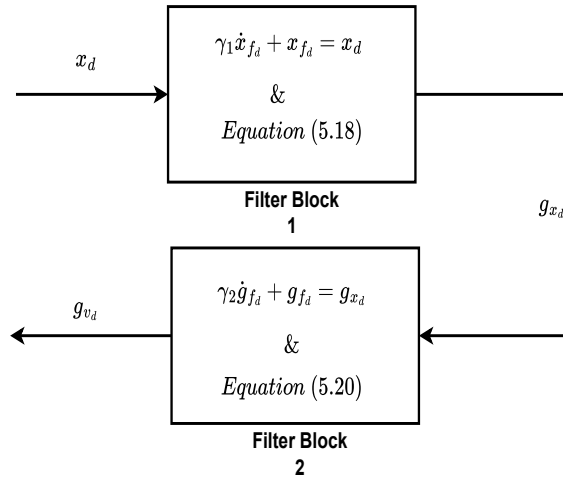


Figure 5.3: Filters illustration

Following the rigorous analysis presented in the subsequent sections, we can use $g_{v_d}(t)$ as an estimate of $\ddot{x}_d(t)$. However, this estimate can suffer from a peaking phenomenon due to the low value of γ_1 and γ_2 , which might jeopardize the saturated thrust requirements to design a smooth attitude tracking controller. To get a saturated thrust input, a differential projection filter for a bounded estimate $g_1(t)$ of $\ddot{x}_d(t)$ with design

parameter $\gamma_{21} \in \mathbb{R}_+$ and $g_1(0) = 0$ is proposed as

$$\dot{g}_1 = Proj(g_1, \phi) = \begin{cases} \phi - \frac{\nabla f(g_1)(\nabla f(g_1))^T}{\|\nabla f(g_1)\|^2} \phi f(g_1), & \text{if } f(g_1) > 0 \text{ \& } \phi^T \nabla f(g_1) > 0 \\ \phi, & \text{otherwise} \end{cases}, \quad (5.21)$$

where

$$\phi = \frac{1}{\gamma_{21}}(g_{vd} - g_1), \quad f(g_1) = \frac{\|g_1\|^2 - h_2^2}{\varepsilon_0 h_2^2}, \quad \nabla f(g_1) = \frac{2g_1}{\varepsilon_0 h_2^2}.$$

h_2 is the known upper bound of the $\ddot{x}_d(t)$ and $\varepsilon_0 > 0$ is the tolerance.

5.2.3 Auxiliary Desired Attitude Construction

The desired attitude $R_c(t)$ is constructed exploiting the virtual control input $f_d(t)$ given in (5.15) and the desired heading direction $x_{B_d}(t)$ as

$$R_c = [x_{B_c}, y_{B_c}, z_{B_c}], \quad (5.22)$$

$$z_{B_c} = \frac{f_d}{\|f_d\|} \quad y_{B_c} = -\frac{x_{B_d} \times z_{B_c}}{\|x_{B_d} \times z_{B_c}\|} \quad x_{B_c} = y_{B_c} \times z_{B_c}. \quad (5.23)$$

Once we construct a well defined¹ desired attitude matrix $R_c(t)$ as given in (5.22), the desired angular velocity $\Omega_c(t)$ and acceleration $\dot{\Omega}_c(t)$ corresponding to this can be calculated as

$$\hat{\Omega}_c = R_c^T \dot{R}_c, \quad (5.24)$$

$$\dot{\hat{\Omega}}_c = R_c^T \ddot{R}_c - \hat{\Omega}_c^2, \quad (5.25)$$

Note that the above calculations are essential to design an accurate attitude tracking control law for the dynamics given in (5.7) and (5.8) Lee *et al.* (2010); Lee (2012). However, one requires information about the higher derivatives of the virtual input $f_d(t)$ and heading direction $x_{B_d}(t)$ to calculate $\dot{R}_c, \ddot{R}_c$. As providing the information mentioned earlier is not feasible, we tackle this issue by systematically defining an auxiliary filter dynamics for the available signal $R_c(t) \in SO(3)$ to obtain a filtered signal $R_d(t) \in SO(3)$ strictly belonging to the special orthogonal group and the corresponding angular veloc-

¹We assume that $x_{B_d}(t) \nparallel z_{B_c}(t)$. We note that one can circumvent this singularity in differential flatness transformation using the Hopf Fibration on $SO(3)$ proposed in [25]. However, this is out of the scope of the current problem formulation.

ity $\Omega_d(t)$ as

$$\dot{R}_d = R_d \hat{\Omega}_d, \quad R_c(0) = I, \quad (5.26)$$

$$\dot{\Omega}_d = -\frac{1}{\gamma_4} e_{\Omega_{dc}} + e_R(R_d, R_c) + \frac{1}{\gamma_3} E(R_d, R_c) \Omega_d, \quad \Omega_d(0) = 0, \quad (5.27)$$

where $e_R(R_d, R_c)$ is the attitude error between R_c and R_d as described in Appendix A. $\gamma_3, \gamma_4 \in \mathbb{R}_+$ are the constant design parameters. The term $E(R_d, R_c)$ is also described in the subsequent section. The error $e_{\Omega_{dc}} = \Omega_d + \frac{1}{\gamma_3} e_R(R_d, R_c)$ is carefully designed based on the stability analysis in the subsequent section.

5.2.4 Effect of large attitude errors on $f(t)$

Authors in Lee *et al.* (2010) discussed the necessity of designing the thrust control input f to reduce its magnitude whenever there are large attitude errors. They did so by multiplying the magnitude of the virtual control input f_d with the dot product of $z_{B_c} = R_c e_3$ and $z_B = R e_3$, i.e., $f = \|f_d\| e_3^T R_c^T R e_3$. The term $e_3^T R_c^T R e_3$ represents the cosine of the angle between z_{B_c} and z_B . This term can make the control input f negative whenever the angle is greater than 90° , making it infeasible for the quadrotor to attain. To address this, authors in Sharma and Kar (2021) suggested using the magnitude of the virtual control input, i.e., $f = \|f_d\|$ as the thrust control input. However, this may cause instability in the overall system dynamics by generating unnecessarily high thrust for large attitude errors. The careful analysis and design of the thrust control input in equation (5.14) attenuates both of these problems by multiplying the scaling quantity $\sqrt{\frac{1+e_3^T R_c^T R e_3}{2}}$ to $\|f_d\|$ which is nothing but cosine of half of the angle between z_{B_c} and z_B .

Remark 8. Note that the proposed virtual control input (5.15) is always positive for a tunable k_α , which is, theoretically, a necessary condition for the existence of the desired attitude $R_c(t)$. Moreover, the requirement of proportionally reduced thrust magnitude for large attitude error is inherently captured in the designed novel thrust input (5.14) in the whole configuration space (especially angles greater than 90°).

5.2.5 Attitude Tracking Controller

To design the attitude tracking controller on a nonlinear space $SO(3)$, we start with constructing a positive definite configuration error function Ψ . We then choose the configuration error vector and a velocity error vector in the tangent space $T_R SO(3)$ utilizing the derivatives of Ψ . Considering the carefully crafted error system, we design the controller following the Lyapunov direct method. Owing to the superior tracking performance for large initial attitude errors, we consider the following configuration error function Lee (2012) $\Psi : SO(3) \times SO(3) \rightarrow \mathbb{R}$ for a given pair of desired attitude and angular velocity $R_1 \in SO(3), \Omega_1 \in \mathbb{R}^3$, and the current attitude and angular velocity $R_2 \in SO(3), \Omega_2 \in \mathbb{R}^3$

$$\Psi(R_2, R_1) = 2 - \sqrt{1 + \text{tr}[R_1^T R_2]}. \quad (5.28)$$

Based on the configuration error function, we define an attitude error vector $e_R(R_1, R_2) : SO(3) \times SO(3) \rightarrow \mathbb{R}^3$ and an angular velocity error vector $e_\Omega : SO(3) \times \mathbb{R}^3 \times SO(3) \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$ as

$$e_R(R_2, R_1) = \frac{1}{2\sqrt{1 + \text{tr}[R_1^T R_2]}} (R_1^T R_2 - R_2^T R_1)^\vee, \quad (5.29)$$

$$e_\Omega(R_2, \Omega_2, R_1, \Omega_1) = \Omega_2 - R_2^T R_1 \Omega_1. \quad (5.30)$$

Throughout this chapter, we will use the following well-established properties Lee (2012) of these error variables in a sub-level set $\mathcal{L}_2 = \{R_1, R_2 \in SO(3) | \Psi(R_2, R_1) < 2\}$ of Ψ

- Ψ is positive definite about $R_2 = R_1$ and has only one critical point $R_2 = R_1$.
- $\|e_R\|^2 \leq \Psi \leq 2\|e_R\|^2$.
- $\|e_R\| = \sin \frac{\|\theta\|}{2}$, where $\exp(\hat{\theta}) = R_1^T R_2$ i.e. $\|e_R\| < 1$.

Equipped with the auxiliary dynamics (5.26) and (5.26), we now design the attitude control input M Lee (2012) to track the auxiliary reference signals $R_d(t), \Omega_d(t), \dot{\Omega}_d(t)$ as follows

$$M = -k_R e_R(R, R_d) - k_\Omega e_\Omega(R, \Omega, R_d, \Omega_d) + \Omega \times J\Omega - J(\hat{\Omega} R^T R_d \Omega_d - R^T R_d \dot{\Omega}_d), \quad (5.31)$$

where $\{k_R, k_\Omega \in \mathbb{R}_+\}$ are constant control design parameters.

5.3 Theoretical Analysis

In this section, we analyze the stability and error convergence of the complete system, providing the theoretical arguments regarding the proposed controller's efficacy in tracking a partially known trajectory. We first derive the comprehensive closed-loop error dynamics for individual subsystems, followed by the results on the boundedness of these closed-loop errors.

5.3.1 Position Tracking Error System

Using the relations given in (5.12) the time derivative of e_x can be given as

$$\dot{e}_x = e_\alpha - \alpha_x \tanh(e_x) - \tanh(e_f). \quad (5.32)$$

Further, the dynamics for error $e_\alpha(t)$ using auxiliary filter dynamics (5.13), can be given as

$$\begin{aligned} \dot{e}_\alpha &= \ddot{x} - \ddot{x}_d + \alpha_x \text{Sech}^2(e_x) \dot{e}_x + \text{Sech}^2(e_f) \dot{e}_f \\ &= \ddot{x} - \ddot{x}_d + \alpha_x \text{Sech}^2(e_x) \dot{e}_x - k_\alpha e_\alpha + \tanh(e_x) - \alpha_f \tanh(e_f). \end{aligned} \quad (5.33)$$

To analyze the effect of designed thrust input $f(t)$ on position tracking errors, we utilize the filter tracking error multiplied by mass, i.e., $me_\alpha(t)$. The open-loop dynamics of $me_\alpha(t)$ can be obtained by substituting the system dynamics (5.6) into (5.33) as

$$m\dot{e}_\alpha = mge_3 - fRe_3 - m\ddot{x}_d + \chi - mk_\alpha e_\alpha + m\tanh(e_x) \quad (5.34)$$

where $\chi = m\alpha_x \text{Sech}^2(e_x) \dot{e}_x - m\alpha_f \tanh(e_f)$. To better understand the attitude dynamics coupling, the virtual control input $f_d(t)$ is added and subtracted to the right-hand side of (5.34), yielding

$$m\dot{e}_\alpha = mge_3 + \Delta f + \chi - m\ddot{x}_d - mk_\alpha e_\alpha + m\tanh(e_x) - f_d, \quad (5.35)$$

where $\Delta f = f_d - fRe_3$.

Using the expression for virtual control input from (5.15), the closed-loop filter error dynamics can be written as

$$m\dot{e}_\alpha = \Delta f + \chi - m(\ddot{x}_d - g_1) - mk_\alpha e_\alpha - m \tanh(e_x) + mk_\alpha \tanh(e_f). \quad (5.36)$$

To analyze the dynamical behavior of the proposed filter structures for desired position reference derivatives $\frac{d^n x_d(t)}{dt^n}$, we propose the following error system $e_d(t)$ as

$$e_d = \begin{bmatrix} e_{g_x} \\ e_{\ddot{x}_d} \\ e_{g_v} \\ e_{g_1} \end{bmatrix} = \begin{bmatrix} \dot{x}_d - g_{x_d} \\ \ddot{x}_d - \dot{g}_{x_d} \\ \dot{g}_{x_d} - g_{v_d} \\ \ddot{x}_d - g_1 \end{bmatrix}. \quad (5.37)$$

Using the expressions from (5.16), (5.17), (5.19) and (5.21), the closed loop error dynamics for error system (5.37) can be given as

$$\begin{bmatrix} \dot{e}_{g_x} \\ \dot{e}_{\ddot{x}_d} \\ \dot{e}_{g_v} \\ \dot{e}_{g_1} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\gamma_1} e_{g_x} + \ddot{x}_d \\ -\frac{1}{\gamma_1} e_{\ddot{x}_d} + \ddot{\ddot{x}}_d \\ -\frac{1}{\gamma_2} e_{g_v} + \frac{1}{\gamma_1} e_{\ddot{x}_d} \\ -\frac{1}{\gamma_{21}} e_{g_1} + \frac{1}{\gamma_{21}} e_{g_v} + \frac{1}{\gamma_{21}} e_{\ddot{x}_d} + \Phi_{g_1} + \ddot{\ddot{x}}_d \end{bmatrix} \quad (5.38)$$

where

$$\Phi_{g_1} = \frac{\nabla f(g_1)(\nabla f(g_1))^T}{\|\nabla f(g_1)\|^2} \phi f(g_1) \quad (5.39)$$

which appears only if $f(g_1) > 0$ & $\phi^T \nabla f(g_1) > 0$. Moreover, $e_{g_1}^T \Phi_{g_1} \leq 0$ (Please refer to the properties of projection in Khalil (1996)).

5.3.2 Attitude Tracking Error System

The time derivative of the configuration function, attitude error, and angular velocity $\Psi(R_2, R_1)$, $e_R(R_2, R_1)$, $e_\Omega(R_2, \Omega_2, R_1, \Omega_1)$ can be given as

$$\frac{d}{dt}\Psi(R_2, R_1) = e_R(R_2, R_1)^T e_\Omega(R_2, \Omega_2, R_1, \Omega_1), \quad (5.40)$$

$$\frac{d}{dt}e_R(R_2, R_1) = E(R_2, R_1)e_\Omega(R_2, \Omega_2, R_1, \Omega_1), \quad (5.41)$$

$$\frac{d}{dt}J e_\Omega(R_2, \Omega_2, R_1, \Omega_1) = J\dot{\Omega}_2 + J(\hat{\Omega}_2 R_2^T R_1 \Omega_1 - R_2^T R_1 \dot{\Omega}_1), \quad (5.42)$$

where $E(R_2, R_1) \in \mathbb{R}^{3 \times 3}$ is given as

$$E(R_2, R_1) = \frac{1}{2\sqrt{1 + \text{tr}[R_1^T R_2]}} (\text{tr}[R_2^T R_1]I - R_2^T R_1 + 2e_R e_R^T). \quad (5.43)$$

Using the fact that $\|E(R_2, R_1)\| = \frac{1}{2}$ from Lee (2012), we can state that

$$\left\| \frac{d}{dt}e_R(R_2, R_1) \right\| \leq \frac{1}{2} \|e_\Omega(R_2, \Omega_2, R_1, \Omega_1)\|. \quad (5.44)$$

For simplicity in the representation, we will use the following notations for the various attitude and angular velocity errors throughout the chapter.

$$e_R(R, R_d) = e_{R_d}, \quad e_R(R, R_c) = e_{R_c}, \quad e_R(R_d, R_c) = e_{R_{dc}}, \quad e_\Omega(R, \Omega, R_d, \Omega_d) = e_{\Omega_d}. \quad (5.45)$$

Using the control input M , the closed loop error dynamics of angular error e_{Ω_d} premultiplied with inertia matrix J can be given as

$$J\dot{e}_{\Omega_d} = -k_R e_{R_d} - k_\Omega e_{\Omega_d}. \quad (5.46)$$

5.3.3 Stability Analysis

The stability and convergence analysis is systematically presented in this section. We first state the following lemmas, which will be utilized in the subsequent analysis. We then present the theoretical results claiming exponential stability of attitude-tracking

errors. Subsequently, the proof of the convergence of overall system tracking errors for the translational, angular, and auxiliary dynamics to a small ultimate bound is provided.

Lemma 7. *There exist positive constants α_1, α_2 such that*

$$\|f_d\| \leq \alpha_1, \quad \|f_d\| \geq \alpha_2. \quad (5.47)$$

Further, α_2 is strictly positive if

$$k_\alpha < g - h_2 - 2. \quad (5.48)$$

Proof. From equation (5.15) we can write

$$\|f_d\| \leq mg + m\|g_1\| + 2m\|\tanh(e_x)\| + mk_\alpha\|\tanh(e_f)\|. \quad (5.49)$$

From the projection-based differential estimate given in equation (5.21), we have $\|g_1\| \leq h_2$ and the fact that $\|\tanh(v)\| \leq 1 \ \forall v \in (-\infty, \infty)$, we can write

$$\|f_d\| \leq mg + mh_2 + 2\sqrt{3}m + m\sqrt{3}k_\alpha = \alpha_1. \quad (5.50)$$

Also, from the third row of equation (5.15) we get

$$f_{d_3} = mg - mg_{1_3} + 2m\tanh(e_{z_3}) - mk_\alpha\tanh(e_{f_3}). \quad (5.51)$$

Note that the projection operator (5.21) ensures that $\|g_{1_3}\| \leq \|g_1\| \leq h_2$. Hence, the lower bound of the $f_{d_3}(t)$ can be obtained as

$$\|f_{d_3}\| \geq mg - mh_2 - 2m - mk_\alpha = \alpha_2. \quad (5.52)$$

α_2 is strictly positive for the given choice of k_α . □

Lemma 8. *The quantity $\Delta f = f_d - fRe_3$ satisfies the following bound*

$$\|\Delta f\| \leq 2\alpha_1(\|e_{R_{dc}}\| + \|e_{R_d}\|). \quad (5.53)$$

Proof. Substituting the expression of $f(t)$ in the expression of Δf , yields

$$\Delta f = \|f_d\| R_c e_3 - \|f_d\| \sqrt{\frac{1 + e_3^T R_c^T R e_3}{2}} R e_3, \quad (5.54)$$

The upper bound of the Δf can be calculated as

$$\|\Delta f\| \leq \left\| \|f_d\| R_c e_3 - \|f_d\| \sqrt{\frac{1 + e_3^T R_c^T R e_3}{2}} R e_3 \right\|. \quad (5.55)$$

Substituting the bounds of $\|f_d\|$ from lemma 7 to equation (5.55)

$$\|\Delta f\| \leq \alpha_1 \left\| R_c e_3 - \sqrt{\frac{1 + e_3^T R_c^T R e_3}{2}} R e_3 \right\|. \quad (5.56)$$

The quantity $e_3^T R_c^T R e_3$ represents the cosine of the angle between vectors $R_c e_3$ and $R e_3$. Let that angle be θ' , then from equation (5.56), we have

$$\|\Delta f\| \leq \alpha_1 \sqrt{1 - 2\cos\left(\frac{\theta'}{2}\right)\cos(\theta') + \cos^2\left(\frac{\theta'}{2}\right)}. \quad (5.57)$$

Using Rodrigues' formula, let's consider the following

$$R_c^T R = \exp(\hat{\theta}_1), \quad R_c^T R_d = \exp(\hat{\theta}_2) \quad R_d^T R = \exp(\hat{\theta}_3), \quad (5.58)$$

In the sublevel set $\mathcal{L}_2$ where $\theta' \leq \|\theta_1\|$ the following holds

$$\sqrt{1 - 2\cos\left(\frac{\theta'}{2}\right)\cos(\theta') + \cos^2\left(\frac{\theta'}{2}\right)} \leq 2\sin\left(\frac{\|\theta_1\|}{2}\right) = 2\|e_{R_c}\|. \quad (5.59)$$

Also, from (5.58) we have

$$R_c^T R = (R_c^T R_d)(R_d^T R), \quad \exp(\hat{\theta}_1) = \exp(\hat{\theta}_2)\exp(\hat{\theta}_3), \quad \exp(\hat{\theta}_1) = \exp(\hat{\theta}_2 + \hat{\theta}_3). \quad (5.60)$$

From (5.60), the following relation holds

$$\|\theta_1\| \leq \|\theta_2\| + \|\theta_3\|. \quad (5.61)$$

Note that the attitude error is equal to $\sin \frac{\|\theta_i\|}{2}$ and in the sub-level set $\mathcal{L}_2$ i.e. $0 \leq \|\theta_i\| < \pi, \forall i = 1, 2, 3$, it is monotonically increasing. Hence,

$$\|e_{R_c}\| \leq \|e_{R_{dc}}\| + \|e_{R_d}\|. \quad (5.62)$$

From equation (5.62) and (5.59), the bounds of $\|f_d\|$ can be given as

$$\|\Delta f\| \leq 2\alpha_1(\|e_{R_{dc}}\| + \|e_{R_d}\|), \quad (5.63)$$

□

Lemma 9. *The desired angular velocity Ω_c satisfies the following bound*

$$\|\hat{\Omega}_c\| \leq \rho_1(\|e_\alpha\|, \|\tanh(e_x)\|, \|\tanh(e_f)\|) + \rho_{01}, \quad (5.64)$$

where $\rho_1()$ is class κ -function and ρ_{01} is a positive constant.

Proof. From (5.24) we have

$$\|\hat{\Omega}_c\| \leq \|\dot{R}_c\|. \quad (5.65)$$

The time derivative of R_c can be written as

$$\dot{R}_c = [\dot{x}_{B_c}, \dot{y}_{B_c}, \dot{z}_{B_c}], \quad (5.66)$$

where

$$\begin{aligned} \dot{z}_{B_c} &= \frac{\dot{f}_d}{\|f_d\|} - \frac{f_d(f_d^T \dot{f}_d)}{\|f_d\|^3}, \\ \dot{y}_{B_c} &= \frac{\dot{A}}{\|A\|} - \frac{(A^T \dot{A})}{\|A\|^3} A, \quad A = x_{B_d} \times z_{B_c} \\ \dot{x}_{B_c} &= \dot{y}_{B_c} \times z_{B_c} + y_{B_c} \times \dot{z}_{B_c}. \end{aligned} \quad (5.67)$$

We individually calculate the upper bounds of column vectors of $\dot{R}_c$. The bounds on $\dot{z}_{B_c}$ can be given as

$$\|\dot{z}_{B_c}\| \leq 2 \frac{\|\dot{f}_d\|}{\|f_d\|} \quad (5.68)$$

Similarly, the bounds on $\dot{y}_{B_c}$ can be given as

$$\|\dot{y}_{B_c}\| \leq 2 \frac{\|\dot{A}\|}{\|A\|} \quad (5.69)$$

Since $x_{B_d} \nparallel z_{B_c}$, we have

$$\|\dot{y}_{B_c}\| \leq \frac{2}{\delta_A} (\|\dot{x}_{B_d}\| + \|\dot{z}_{B_c}\|), \quad (5.70)$$

where $\delta_A \geq \|x_{B_d} \times z_{B_c}\|$. Using, equation (5.67) and properties of SO(3) we can obtain the upper bound of $\dot{x}_{B_c}$ as

$$\|\dot{x}_{B_c}\| \leq \|\dot{z}_{B_c}\| + \|\dot{y}_{B_c}\| \quad (5.71)$$

Now, the upper bound of $\|\hat{\Omega}_c\|$ can be written as

$$\begin{aligned} \|\hat{\Omega}_c\| &\leq \|\dot{z}_{B_c}\| + \|\dot{y}_{B_c}\| + \|\dot{x}_{B_c}\| \\ &\leq 2\|\dot{z}_{B_c}\| + 2\|\dot{y}_{B_c}\| \\ &\leq \left(4 + \frac{4}{\delta_A}\right) \frac{\|\dot{f}_d\|}{\|f_d\|} + \frac{4}{\delta_A} h_5. \end{aligned} \quad (5.72)$$

Let's evaluate the bound on the time derivatives of f_d . The time derivatives of f_d can be given as

$$\dot{f}_d = -m\dot{g}_1 + 2m\text{Sech}^2(e_x)\dot{e}_x - mk_\alpha\text{Sech}^2(e_f)\dot{e}_f. \quad (5.73)$$

Substituting the expressions from (5.13), (5.21) and (5.32) in the (5.73) will yield

$$\begin{aligned} \dot{f}_d &= -m\phi + m\Phi_{g_1} + 2m\text{Sech}^2(e_x) (e_\alpha - \alpha_x \tanh(e_x) - \tanh(e_f)) \\ &\quad + mk_\alpha^2 e_\alpha - mk_\alpha \tanh(e_x) + mk_\alpha \alpha_f \tanh(e_f). \end{aligned} \quad (5.74)$$

The quantity Φ_{g_1} can be bounded as

$$\begin{aligned}\|\Phi_{g_1}\| &\leq \frac{\|\nabla f(g_1)\|^2}{\|\nabla f(g_1)\|^2} \|\phi\| \|f(g_1)\|, \\ &\leq \|\phi\| \|f(g_1)\|,\end{aligned}\tag{5.75}$$

and Φ_{g_1} appears when $f(g_1) > 0$ (i.e. $h_2 < \|g_1\| < h_2 + \varepsilon_0$), then

$$\|\Phi_{g_1}\| \leq \|\phi\| \frac{\varepsilon_0 + 2h_2}{h_2^2},\tag{5.76}$$

where ϕ can be written in the form of filter errors as

$$\phi = \frac{1}{\gamma_{21}}(e_{g_1} - e_{g_v} - e_{\ddot{x}_d}).\tag{5.77}$$

Then from equation (5.38) and (5.74), we can write

$$\begin{aligned}\|\dot{f}_d\| &\leq \frac{m}{\gamma_{21}} \left(1 + \frac{\varepsilon_0 + 2h_2}{h_2^2}\right) (\|e_{g_1}\| + \|e_{g_v}\| + \|e_{\ddot{x}_d}\|) + 2m(\|e_\alpha\| + \alpha_x \|\tanh(e_x)\|) \\ &\quad + 2m\|\tanh(e_f)\| + mk_\alpha^2\|e_\alpha\| + mk_\alpha\|\tanh(e_x)\| + mk_\alpha\alpha_f\|\tanh(e_f)\|.\end{aligned}\tag{5.78}$$

Using lemma 7, 10 and relation (5.78), the equation (5.72) can be written as

$$\|\hat{\Omega}_c\| \leq \rho_1(\|e_\alpha\|, \|\tanh(e_x)\|, \|\tanh(e_f)\|) + \rho_{01},\tag{5.79}$$

where $\rho_1(\cdot)$ is class κ -function and ρ_{01} is a positive constant. $\square$

Lemma 10. *The filter errors given in equation (5.38) can be upper-bounded as*

$$\|e_{g_x}(t)\| \leq \sqrt{\|e_{g_x}(0)\|^2 \exp\left(-\frac{1}{\gamma_1}t\right) + \gamma_1^2 h_2^2} \leq \max(\|e_{g_x}(0)\|, \gamma_1 h_2) = \alpha_3,\tag{5.80}$$

$$\|e_{\ddot{x}_d}(t)\| \leq \sqrt{\|e_{\ddot{x}_d}(0)\|^2 \exp\left(-\frac{1}{\gamma_1}t\right) + \gamma_1^2 h_3^2} \leq \max(\|e_{\ddot{x}_d}(0)\|, \gamma_1 h_3) = \alpha_4,\tag{5.81}$$

$$\|e_{g_v}(t)\| \leq \sqrt{\|e_{g_v}(0)\|^2 \exp\left(-\frac{1}{\gamma_2}t\right) + \frac{\gamma_2^2}{\gamma_1^2} \|e_{\ddot{x}_d}\|^2} \leq \alpha_5,\tag{5.82}$$

$$\|e_{g_1}(t)\| \leq \max\left(\|e_{g_1}(0)\|, \sqrt{8(\alpha_4^2 + \alpha_5^2) + 2\gamma_{21}^2 h_3^2}\right) = \alpha_6,\tag{5.83}$$

where $\alpha_5 = \max\left(\|e_{g_v}(0)\|, \frac{\gamma_2}{\gamma_1} \max(e_{\ddot{x}_d}(0), \gamma_1 h_3)\right)$.

Proof. For the error e_{g_x} , lets consider the following Lyapunov candidate

$$V_{g_x} = \frac{1}{2} e_{g_x}^T e_{g_x}. \quad (5.84)$$

Taking the time derivative of $V_{g_x}(t)$ and using (5.39) will yield

$$\dot{V}_{g_x} = -\frac{1}{\gamma_1} e_{g_x}^T e_{g_x} + e_{g_x}^T \ddot{x}_d. \quad (5.85)$$

This implies that

$$\|e_{g_x}\| = \sqrt{2V_{g_x}(t)} \leq \sqrt{\|e_{g_x}(0)\|^2 \exp\left(-\frac{1}{\gamma_1} t\right) + \gamma_1^2 h_2^2}. \quad (5.86)$$

Similarly, for error $e_{\ddot{x}_d}(t)$, consider the following Lyapunov candidate

$$V_{\ddot{x}_d} = \frac{1}{2} e_{\ddot{x}_d}^T e_{\ddot{x}_d}. \quad (5.87)$$

Taking the time derivative of $V_{\ddot{x}_d}(t)$ and using (5.39) will yield

$$\dot{V}_{\ddot{x}_d} = -\frac{1}{\gamma_1} e_{\ddot{x}_d}^T e_{\ddot{x}_d} + e_{\ddot{x}_d}^T \ddot{\ddot{x}}_d. \quad (5.88)$$

This implies that

$$\|e_{\ddot{x}_d}\| = \sqrt{2V_{\ddot{x}_d}(t)} \leq \sqrt{\|e_{\ddot{x}_d}(0)\|^2 \exp\left(-\frac{1}{\gamma_1} t\right) + \gamma_1^2 h_3^2}. \quad (5.89)$$

For error e_{g_v} , consider the following Lyapunov candidate

$$V_{g_v} = \frac{1}{2} e_{g_v}^T e_{g_v}. \quad (5.90)$$

Taking the time derivative of $V_{g_v}(t)$ and using (5.39) will yield

$$\dot{V}_{g_v} = -\frac{1}{\gamma_2} e_{g_v}^T e_{g_v} + \frac{1}{\gamma_1} e_{g_v}^T e_{\ddot{x}_d}. \quad (5.91)$$

This implies that

$$\|e_{g_v}\| = \sqrt{2V_{g_v}(t)} \leq \sqrt{\|e_{g_v}(0)\|^2 \exp\left(-\frac{1}{\gamma_2}t\right) + \frac{\gamma_2^2}{\gamma_1^2} \|e_{\ddot{x}_d}\|^2}. \quad (5.92)$$

Finally, we will analyze the dynamical behavior of the error $e_{g_1}(t)$ by choosing the following Lyapunov candidate

$$V_{g_1} = \frac{1}{2} e_{g_1}^T e_{g_1}. \quad (5.93)$$

Taking the time derivative of $V_{g_1}(t)$ and using (5.39) will yield

$$\dot{V}_{g_1} = -\frac{1}{\gamma_{21}} e_{g_1}^T e_{g_1} + \frac{1}{\gamma_{21}} e_{g_1}^T e_{g_v} + \frac{1}{\gamma_{21}} e_{g_1}^T e_{\ddot{x}_d} + e_{g_1}^T \Phi_{g_1} + e_{g_1}^T \ddot{x}_d. \quad (5.94)$$

Using the fact that $e_{g_1}^T \Phi_{g_1} \leq 0$, the bounds on $\dot{V}_{g_1}(t)$ can be given as

$$\dot{V}_{g_1} \leq -\frac{1}{2\gamma_{21}} \|e_{g_1}\|^2 + \frac{2}{\gamma_{21}} \|e_{g_v}\|^2 + \frac{2}{\gamma_{21}} \|e_{\ddot{x}_d}\|^2 + \gamma_{21} h_3^2. \quad (5.95)$$

Solving (5.95) for $V_{g_1}(t)$ using equations (5.81), (5.82)

$$V_{g_1}(t) \leq V_{g_1}(0) \exp\left(-\frac{1}{\gamma_{21}}t\right) + 4\alpha_4^2 + 4\alpha_5^2 + \gamma_{21}^2 h_3^2. \quad (5.96)$$

This implies that

$$\|e_{g_1}(t)\| = \sqrt{2V_{g_1}(t)} \leq \sqrt{\|e_{g_1}(0)\|^2 \exp\left(-\frac{1}{\gamma_{21}}t\right) + \alpha_7}, \quad (5.97)$$

where $\alpha_7 = 8(\alpha_4^2 + \alpha_5^2) + 2\gamma_{21}^2 h_3^2$. □

Next, we derive the stability and convergence properties of the attitude subsystem of the multirotor tracking the auxiliary desired attitude.

Theorem 7. (Adapted from Lee (2012)) The closed loop error dynamics of $e_1(t) = [e_{R_d}(t)^T, e_{\Omega_d}(t)^T]^T$ is exponentially stable with the control input M defined in (5.31) for the smooth desired attitude reference $(R_d(t), \Omega_d(t), \dot{\Omega}_d(t))$. Moreover, an estimate of the region of attraction is given by

$$\Psi(R(0), R_d(0)) < 2, \quad (5.98)$$

$$\|e_{\Omega_d}(0)\|^2 < \frac{2k_R}{\lambda_{\max}(J)}(2 - \Psi(R(0), R_d(0))). \quad (5.99)$$

Proof. We start by proving that given conditions (5.98) and (5.99), the sublevel set $\mathcal{L}_2 = \{R, R_d \in \text{SO}(3) | \Psi(R, R_d) < 2\}$ is positively invariant set. Consider the following Lyapunov candidate

$$V_1 = \frac{1}{2}e_{\Omega_d}^T e_{\Omega_d} + k_R \Psi(R, R_d) \quad (5.100)$$

Taking the time derivative of $V_1(t)$ after substituting the dynamics (5.46) and (5.40) yields

$$\dot{V}_1 = -k_{\Omega} \|e_{\Omega_d}\|^2. \quad (5.101)$$

Equation (5.101) along with (5.98) and (5.99) implies that

$$k_R \Psi(R(t), R_d(t)) \leq V_1(t) \leq V_1(0) < 2k_R \quad \forall t > 0. \quad (5.102)$$

This proves that the sub-level set $\mathcal{L}_2$ is a positively invariant set such that $\Psi(R(t), R_d(t)) < 2$ and hence the attitude errors are well defined.

To prove the exponential stability, let's consider another Lyapunov candidate

$$V_2 = \frac{1}{2}e_{\Omega_d}^T e_{\Omega_d} + k_R \Psi(R, R_d) + c_1 e_{\Omega_d}^T e_{R_d}, \quad (5.103)$$

where c_1 is a positive constant. Using the properties of configuration error function, for $e_1(t) = [e_{R_d}(t)^T, e_{\Omega_d}(t)^T]^T$ we get

$$e_1^T W_{11} e_1 \leq V_1 \leq e_1^T W_{12} e_1, \quad (5.104)$$

where

$$W_{11} = \begin{bmatrix} k_R & \frac{c_2}{2} \\ \frac{c_2}{2} & \frac{\lambda_{\min}(J)}{2} \end{bmatrix}, \quad W_{12} = \begin{bmatrix} 2k_R & \frac{c_2}{2} \\ \frac{c_2}{2} & \frac{\lambda_{\max}(J)}{2} \end{bmatrix}. \quad (5.105)$$

Taking time derivative of $V_2(t)$ and substituting the appropriate error dynamics using (5.46), (5.40) and (5.41) will yield

$$\dot{V}_2 = -k_\Omega e_{\Omega_d}^T e_{\Omega_d} - c_1 k_R e_{R_d}^T J^{-1} e_{R_d} + c_1 k_\Omega e_{R_d}^T J^{-1} e_{\Omega_d} + c_1 e_{\Omega_d}^T E^T(R, R_d) e_{\Omega_d}. \quad (5.106)$$

Further taking bounds using the properties of attitude errors in equation (5.106) will yield

$$\dot{V}_2 \leq - \underbrace{\left[k_\Omega - \frac{c_1}{2} - \frac{c_1 k_\Omega}{2\lambda_{\min}(J)} \right]}_{\bar{\lambda}_1} \|e_{\Omega_d}\|^2 - \underbrace{\left[\frac{c_1 k_R}{\lambda_{\max}(J)} - \frac{c_1 k_\Omega}{2\lambda_{\min}(J)} \right]}_{\bar{\lambda}_2} \|e_{R_d}\|^2. \quad (5.107)$$

Furthermore, we can write

$$\dot{V}_2 \leq -\bar{\beta} V_2, \quad (5.108)$$

where $\bar{\beta} = \frac{\min(\bar{\lambda}_1, \bar{\lambda}_2)}{\lambda_{\max}(W_{12})}$. We choose the parameter so that $\bar{\beta}$ is positive. Then, the relation (5.108) invokes global uniform exponential stability (based on Theorem 4.10 in Khalil (2002)). $\square$

Now, we are ready to state our main results.

Theorem 8. *Consider the multi-rotor dynamics (5.5), (5.6), (5.7) and (5.8), with control inputs (5.14) and (5.31). The closed loop dynamics of overall system errors $e(t) = [e_1(t)^T, e_2(t)^T, e_3(t)^T]^T$ where $e_1(t) = [e_{R_d}(t)^T, e_{\Omega_d}(t)^T]^T$, $e_2(t) = [e_{R_{dc}}^T(t), e_{\Omega_{dc}}^T(t)]^T$ and $e_3(t) = [e_\alpha^T(t), \tanh(e_x(t))^T, \tanh(e_f(t))^T]^T$, is uniformly ultimately bounded, provided*

the following gain conditions are satisfied

$$mk_\alpha - \frac{m\alpha_x(3\alpha_x + 1)}{2} - \frac{m\alpha_f}{2} - \rho_{03} > 0, \quad (5.109)$$

$$m\alpha_x - \frac{m\alpha_x^2}{2} > 0, \quad (5.110)$$

$$\frac{1}{2\gamma_3} - 2\alpha_1^2 > 0, \quad (5.111)$$

$$\frac{m\alpha_f}{2} - \frac{m\alpha_x}{2} > 0, \quad (5.112)$$

$$k_\Omega - \frac{c_1}{2} - \frac{c_1 k_\Omega}{2\lambda_{\min}(J)} > 0, \quad (5.113)$$

$$\frac{c_1 k_R}{\lambda_{\max}(J)} - \frac{c_1 k_\Omega}{2\lambda_{\min}(J)} - 2\alpha_1^2 > 0, \quad (5.114)$$

where the constant ρ_{03} is defined subsequently.

Proof. For the sake of simplicity, we proceed in the following steps to prove the theorem claims.

Step 1

Let's first consider the Lyapunov candidate $V_3(t)$ for the translational error $e_3(t)$ of multi-rotor

$$\begin{aligned} V_3 = & \frac{1}{2} m e_\alpha^T e_\alpha + m \ln(\cosh(e_{x_1})) \\ & + m \ln(\cosh(e_{x_2})) + m \ln(\cosh(e_{x_3})) + \frac{1}{2} m \tanh(e_f)^T \tanh(e_f). \end{aligned} \quad (5.115)$$

Taking the time derivative of $V_3(t)$ and substituting equations (5.13), (5.32) and (5.36) yields

$$\begin{aligned} \dot{V}_3 = & -mk_\alpha e_\alpha^T e_\alpha + e_\alpha^T \Delta f + e_\alpha^T \chi - m e_\alpha^T e_{g_1} - m e_\alpha^T \tanh(e_x) + m k_\alpha e_\alpha^T \tanh(e_f) \\ & + m \tanh^T(e_x) (e_\alpha - \alpha_x \tanh(e_x) - \tanh(e_f)) \\ & + m \tanh(e_f)^T (-k_\alpha e_\alpha + \tanh(e_x) - \alpha_f \tanh(e_f)) \end{aligned} \quad (5.116)$$

Further solving (5.116) and taking bounds on $\dot{V}_2(t)$ yield

$$\begin{aligned}\dot{V}_3 \leq & -mk_\alpha \|e_\alpha\|^2 - m\alpha_x \|\tanh(e_x)\|^2 - m\alpha_f \|\tanh(e_f)\|^2 \\ & + \|e_\alpha\| \|\Delta f\| + \|e_\alpha\| \|\chi\| + m\|e_\alpha\| \|e_{g1}\|. \end{aligned} \quad (5.117)$$

Substituting the dynamics of $\dot{e}_x(t)$ from (5.32) in χ , the following can be deduced

$$\begin{aligned}\chi = & m\alpha_x \text{Sech}^2(e_x) e_\alpha - m\alpha_x^2 \text{Sech}^2(e_x) \tanh(e_x) \\ & - m\alpha_x \text{Sech}^2(e_x) \tanh(e_f) - m\alpha_f \tanh(e_f) \end{aligned} \quad (5.118)$$

Solving equation (5.117) utilising equations (5.118), (5.53) and (5.83) produces

$$\begin{aligned}\dot{V}_3 \leq & -mk_\alpha \|e_\alpha\|^2 - m\alpha_f \|\tanh(e_f)\|^2 + \|e_\alpha\| (m\alpha_x \|\text{Sech}^2(e_x)\| \|\tanh(e_f)\|) \\ & + m\alpha_f \|e_\alpha\| \|\tanh(e_f)\| - m\alpha_x \|\tanh(e_x)\|^2 + m\|e_\alpha\| \alpha_6 \\ & + \|e_\alpha\| \|\text{Sech}^2(e_x)\| (m\alpha_x \|e_\alpha\| + m\alpha_x^2 \|\tanh(e_x)\|) + 2\alpha_1 \|e_\alpha\| (\|e_{R_{dc}}\| + \|e_{R_d}\|), \end{aligned} \quad (5.119)$$

which modifies to,

$$\begin{aligned}\dot{V}_3 \leq & -mk_\alpha \|e_\alpha\|^2 - m\alpha_x \|\tanh(e_x)\|^2 - m\alpha_f \|\tanh(e_f)\|^2 + m\alpha_x \|e_\alpha\|^2 \\ & + m\alpha_x^2 \|e_\alpha\| \|\tanh(e_x)\| + m\alpha_x \|e_\alpha\| \|\tanh(e_f)\| + m\alpha_f \|e_\alpha\| \|\tanh(e_f)\| \\ & + m\|e_\alpha\| \alpha_6 + 2\alpha_1 \|e_\alpha\| (\|e_{R_{dc}}\| + \|e_{R_d}\|), \\ \leq & -mk_\alpha \|e_\alpha\|^2 - m\alpha_x \|\tanh(e_x)\|^2 - m\alpha_f \|\tanh(e_f)\|^2 + m\alpha_x \|e_\alpha\|^2 \\ & + \frac{m\alpha_x^2}{2} \|e_\alpha\|^2 + \frac{m\alpha_x^2}{2} \|\tanh(e_x)\|^2 + \frac{m\alpha_x}{2} \|e_\alpha\|^2 + \frac{m\alpha_x}{2} \|\tanh(e_f)\|^2 + \frac{m\alpha_f}{2} \|e_\alpha\|^2 \\ & + \frac{m\alpha_f}{2} \|\tanh(e_f)\|^2 + \frac{m}{2} \|e_\alpha\|^2 + \frac{m}{2} \alpha_6^2 + \|e_\alpha\|^2 + 2\alpha_1^2 \|e_{R_{dc}}\|^2 + 2\alpha_1^2 \|e_{R_d}\|^2. \end{aligned} \quad (5.120)$$

Further manipulation yield

$$\begin{aligned}\dot{V}_3 \leq & - \left(mk_\alpha - m\alpha_x - \frac{m\alpha_x^2}{2} - \frac{m\alpha_x}{2} - \frac{m\alpha_f}{2} - \frac{m}{2} - 1 \right) \|e_\alpha\|^2 \\ & - \left(m\alpha_x - \frac{m\alpha_x^2}{2} \right) \|\tanh(e_x)\|^2 - \left(m\alpha_f - \frac{m\alpha_x}{2} - \frac{m\alpha_f}{2} \right) \|\tanh(e_f)\|^2 \\ & + 2\alpha_1^2 \|e_{R_{dc}}\|^2 + 2\alpha_1^2 \|e_{R_d}\|^2 + \frac{m}{2} \alpha_6^2.\end{aligned}\quad (5.121)$$

Step 2

Now, let's consider another Lyapunov candidate $V_4(t)$ for auxiliary attitude errors $e_2(t)$ as

$$V_4(t) = \Psi(R_d, R_c) + \frac{1}{2} e_{\Omega_{dc}}^T e_{\Omega_{dc}}, \quad (5.122)$$

Taking the time derivative and using equations (5.26), (5.27), (5.40), (5.41) and (5.42) yields

$$\dot{V}_4(t) = -\frac{1}{\gamma_3} e_{R_{dc}}^T e_{R_{dc}} - \frac{1}{\gamma_4} e_{\Omega_{dc}}^T e_{\Omega_{dc}} - e_{R_{dc}}^T R_d^T R_c \Omega_c + \frac{1}{\gamma_3} e_{\Omega_{dc}}^T E(R_d, R_c) R_d^T R_c \Omega_c \quad (5.123)$$

Taking bounds on $\dot{V}_4(t)$ yields

$$\dot{V}_4(t) \leq -\frac{1}{2\gamma_3} \|e_{R_{dc}}\|^2 - \frac{1}{2\gamma_4} \|e_{\Omega_{dc}}\|^2 + \left(\frac{\gamma_3}{2} + \frac{\gamma_4}{8\gamma_3^2} \right) \|\Omega_c\|^2 \quad (5.124)$$

Step 3

To prove the stability of dynamics of error $e(t)$, let's consider the Lyapunov candidate as

$$V = V_2(t) + V_3(t) + V_4(t) \quad (5.125)$$

Taking the derivative of $V(t)$ using (5.124), (5.121) and (5.107) yields

$$\begin{aligned}
\dot{V} \leq & - \left(mk_\alpha - m\alpha_x - \frac{m\alpha_x^2}{2} - \frac{m\alpha_x}{2} - \frac{m\alpha_f}{2} - \frac{m}{2} - 1 \right) \|e_\alpha\|^2 \\
& - \left(m\alpha_x - \frac{m\alpha_x^2}{2} \right) \|\tanh(e_x)\|^2 - \left(m\alpha_f - \frac{m\alpha_x}{2} - \frac{m\alpha_f}{2} \right) \|\tanh(e_f)\|^2 + \frac{m}{2} \alpha_6^2 \\
& - \left(\frac{1}{2\gamma_3} - 2\alpha_1^2 \right) \|e_{R_{dc}}\|^2 + \left(\frac{\gamma_3}{2} + \frac{\gamma_4}{8\gamma_3^2} \right) \|\Omega_c\|^2 - \left(k_\Omega - \frac{c_1}{2} - \frac{c_1 k_\Omega}{2\lambda_{\min}(J)} \right) \|e_{\Omega_d}\|^2 \\
& - \frac{1}{2\gamma_4} \|e_{\Omega_{dc}}\|^2 - \left(\frac{c_1 k_R}{\lambda_{\max}(J)} - \frac{c_1 k_\Omega}{2\lambda_{\min}(J)} - 2\alpha_1^2 \right) \|e_{R_d}\|^2
\end{aligned} \tag{5.126}$$

From lemma (9) and using the properties of hyperbolic functions, we can state that for some $\|e_\alpha\| < \bar{e}_\alpha$, the function $\rho_1()$ is locally Lipschitz. Thus,

$$\|\hat{\Omega}_c\|^2 \leq L_1^2 \|e_\alpha\|^2 + \rho_{02}^2, \tag{5.127}$$

where L_1 , a function of $\bar{e}_\alpha$ is the Lipschitz constant and ρ_{02} is a positive constant. Substituting the (5.127) in (5.126) yields

$$\begin{aligned}
\dot{V} \leq & - \underbrace{\left[mk_\alpha - \frac{m\alpha_x(3\alpha_x+1)}{2} - \frac{m\alpha_f}{2} - \rho_{03} \right]}_{\lambda_1} \|e_\alpha\|^2 - \underbrace{\left[m\alpha_x - \frac{m\alpha_x^2}{2} \right]}_{\lambda_2} \|\tanh(e_x)\|^2 \\
& - \underbrace{\left[\frac{m\alpha_f}{2} - \frac{m\alpha_x}{2} \right]}_{\lambda_3} \|\tanh(e_f)\|^2 - \underbrace{\left[k_\Omega - \frac{c_1}{2} - \frac{c_1 k_\Omega}{2\lambda_{\min}(J)} \right]}_{\lambda_6} \|e_{\Omega_d}\|^2 \\
& - \underbrace{\left[\frac{c_1 k_R}{\lambda_{\max}(J)} - \frac{c_1 k_\Omega}{2\lambda_{\min}(J)} - 2\alpha_1^2 \right]}_{\lambda_7} \|e_{R_d}\|^2 - \underbrace{\left[\frac{1}{2\gamma_3} - 2\alpha_1^2 \right]}_{\lambda_4} \|e_{R_{dc}}\|^2 - \underbrace{\left[\frac{1}{2\gamma_4} \right]}_{\lambda_5} \|e_{\Omega_{dc}}\|^2 \\
& + \underbrace{\left[\left(\frac{\gamma_3}{2} + \frac{\gamma_4}{8\gamma_3^2} \right) \rho_{02}^2 + \frac{m}{2} \alpha_6^2 \right]}_D,
\end{aligned} \tag{5.128}$$

where $\rho_{03} = \frac{m}{2} + 1 + \frac{4\gamma_3^2 + \gamma_4}{8\gamma_3^2} L_1^2$. Equation (5.128) implies

$$\dot{V} \leq - \underbrace{\min(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7)}_{\lambda_v} \|e\|^2 + D \tag{5.129}$$

The relation (5.129) invokes uniformly ultimately bounded (UUB) stability since for $\|e(t)\|^2 > \frac{D}{\lambda_v}$, $\dot{V}(t)$ is negative-definite. However, note that (5.129) is derived based on (5.127), which demands $\|e_\alpha\| \leq \bar{e}_\alpha$. Hence, $\bar{e}_\alpha$ should be chosen such that $\bar{e}_\alpha \geq$

$\sqrt{2 \frac{\max(\bar{V}(0), \frac{D}{\lambda_V})}{m}}$, where $\bar{V}(0)^2$ is a known upper-bound of $V(0)$, and accordingly the gain condition related to λ_1 has to be satisfied. This would ensure $\|e_\alpha(t)\| \leq \bar{e}_\alpha, \forall t \geq 0$, and hence, the UUB result would follow. □

Remark 9. *The UUB results from (5.129) follows only when $\|e_\alpha\| \leq \bar{e}_\alpha$ and $\bar{e}_\alpha \geq \sqrt{2 \frac{\max(\bar{V}(0), \frac{D}{\lambda_V})}{m}}$. To select the gain, choose a $\bar{e}_\alpha$ according to the aforementioned conditions at $t = 0$. It's worth noting that D can be made arbitrarily small by choosing the filter gains $\gamma_1, \gamma_2, \gamma_{21}, \gamma_3, \gamma_4$ appropriately. This relaxes the unnecessary burden on the controller gains for such formulations (i.e., no need to make λ_V very high). Further, one should select the controller gains according to relations (5.109)-(5.114) based on $\bar{e}_\alpha$ assuming a Lipschitz constant L_1 . Moreover, if k_α is chosen such that the condition (5.48) is also satisfied then the desired attitude $R_c(t)$ always exists as $\|f_d(t)\| \geq \alpha_2 > 0$ is guaranteed $\forall t \geq 0$.*

5.4 Results

The efficacy of the proposed controller is tested using two numerical examples simulated in a MATLAB (2021) environment using a quadrotor model. We assume that at time t , the quadrotor has access to only desired position $x_d(t)$ and heading $x_{B_d}(t)$. For the simulations, we consider the quadrotor model with the following mass $m = 4.34$ kg, inertia

$$J = \begin{bmatrix} 0.0820 & 0 & 0 \\ 0 & 0.0845 & 0 \\ 0 & 0 & 0.1377 \end{bmatrix} \text{ kg/m}^2,$$

and the following initial conditions

$$x(0) = [5, 5, -5]^T, \quad v(0) = [0, 0, 0]^T, \quad R(0) = \exp(\frac{\pi}{5} \hat{e}_2), \quad \Omega(0) = [0, 0, 0]^T,$$

²Since some of the error variables used in $V(t)$ are not obtainable from the measurements, a conservative upper-bound has to be used for $V(0)$

where $e_2 = [0, 1, 0]^T$. For both examples, the following controller parameters are selected and found to be performing well in our simulations

$$k_\alpha = 5, \alpha_x = 1, \alpha_f = 2, k_R = 8.81, k_\Omega = 2.54, h_2 = 2.5, \gamma_1 = 10^{-2}, \gamma_2 = 10^{-3}, \\ \gamma_3 = 10^{-2}, \gamma_4 = 10^{-3}, \gamma_{21} = 10^{-2.5}, \varepsilon_0 = 0.1.$$

Further, we compare our results with the classical geometric controller given in Lee *et al.* (2010) with access to the desired trajectory derivatives. The controller parameters are chosen similarly to those in Lee *et al.* (2010). This comparative analysis aims to demonstrate that our proposed method attains comparable performance levels without access to higher derivatives of the desired trajectory.

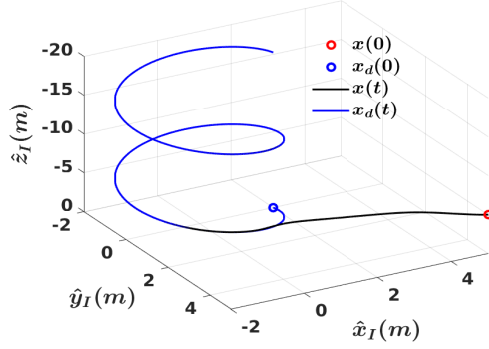
5.4.1 Example 1

For this example, we consider the following desired trajectory

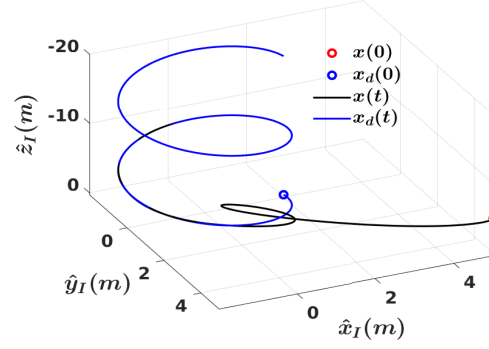
$$x_d(t) = \left[2 \cos\left(\frac{\pi t}{10}\right), 2 \sin\left(\frac{\pi t}{10}\right), -\frac{t}{2} \right]^T, \quad x_{B_d}(t) = \frac{\dot{x}_d(t)}{\|\dot{x}_d(t)\|}.$$

Note that the expression $\dot{x}_d(t)$ is solely used to represent the structure of $x_{B_d}(t)$. The results are compiled in Figure 5.4. Figure 5.4a represents the quadrotor trajectory plot in the 3-D plane with the proposed controller, while the trajectory plot with the classical geometric controller Lee *et al.* (2010) is shown in Figure 5.4b. Figure 5.4d shows the 2-norm of the positional tracking error (i.e., $\|e_x(t)\|$) with the proposed controller and that of the classical geometric controller. The plots show that the quadrotor can accurately track the desired trajectory with the proposed controller despite only supplying position and heading. Further from Figure 5.4d, one can observe that the proposed controller makes the tracking error converge faster with an exponential decay while the convergence is asymptotic and slow with the classical geometric controller. Figures 5.4e and 5.4f represent the attitude errors $e_{R_d}(t)$ and angular velocity error $e_{\Omega_d}(t)$, respectively for the reference generated by auxiliary attitude dynamics (5.26) and (5.27). Figure 5.4g shows that the attitude error $e_{R_c}(t)$ converges to a very small ultimate bound around the origin verifying quadrotor achieving the desired attitude as per the claims in Theorem 8. Further, the error $e_{g_1}(t)$ between the second-time derivative of the desired trajectory and its estimate is represented in Figure 5.4h, showing the convergence to an

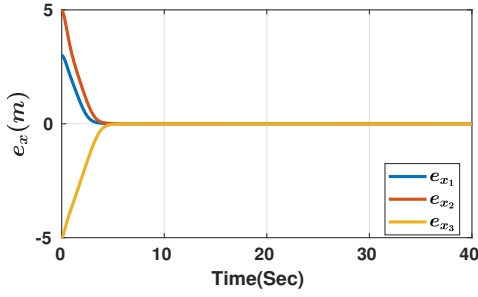
ultimate bound within the origin as claimed in Lemma 10.



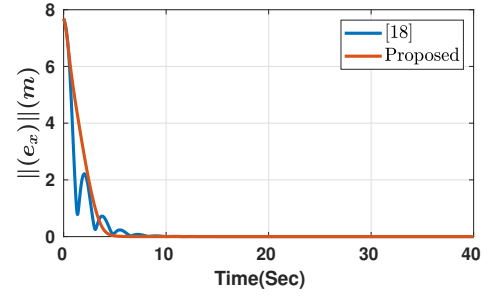
(a) Trajectory plot with proposed controller



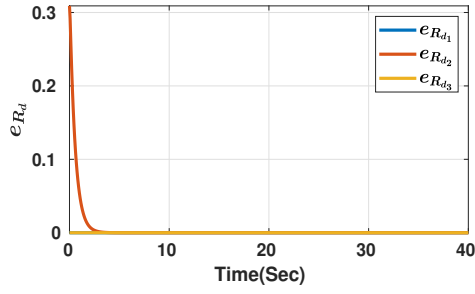
(b) Trajectory plot with controller in Lee *et al.* (2010)



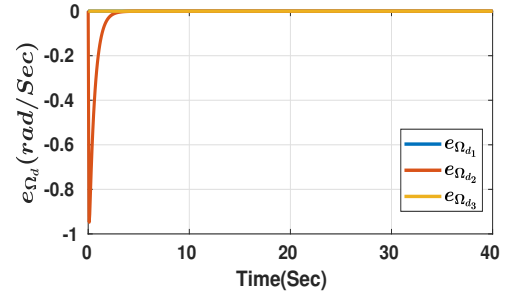
(c) Position error



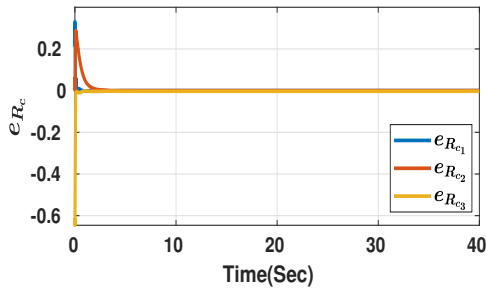
(d) Position error norm comparison plot



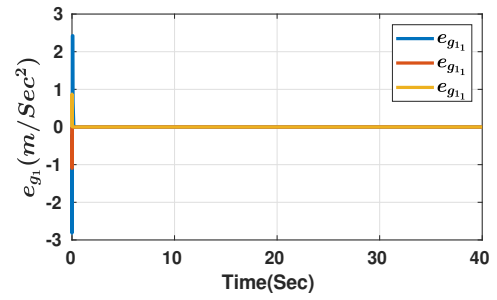
(e) $e_{R_d}(t)$



(f) $e_{\Omega_d}(t)$



(g) $e_{R_c}(t)$



(h) $e_{g_1}(t)$

Figure 5.4: Simulation results for Example 1. [18] in there error norm plot represents plot with controller in Lee *et al.* (2010)

5.4.2 Example 2

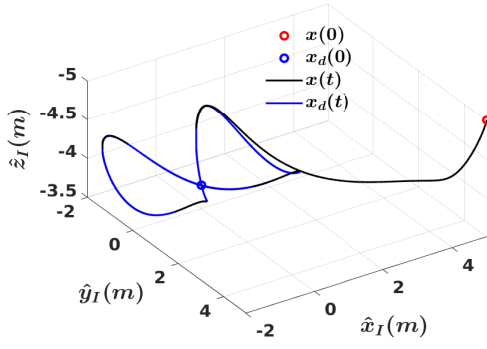
To test the performance of the proposed controller with more aggressive trajectories, we consider a Lissajous desired trajectory as follows

$$x_d(t) = [2 \sin(0.5t), 2 \sin(t), -4 + 0.2 \cos(2t)]^T, \quad x_{B_d}(t) = [\cos(\frac{\pi}{5}t), \sin(\frac{\pi}{5}t), 0]^T.$$

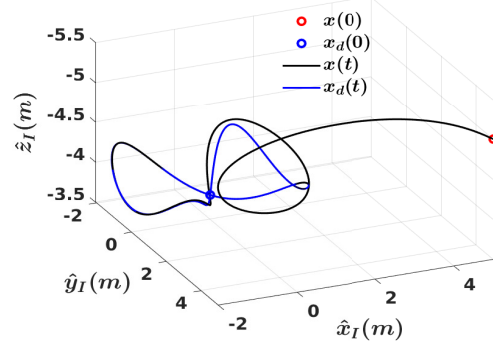
The findings for this example are consolidated in Figure 5.5. Figure 5.5a illustrates the trajectory plot of the quadrotor in the 3-D plane employing the proposed controller. In contrast, Figure 5.5b showcases the trajectory plot utilizing the classical geometric controller Lee *et al.* (2010), where all the necessary derivatives of the desired trajectory are supplied to the controller. Figure 5.5d displays the 2-norm of the positional tracking error with both controllers. Similar to Example 1, it's evident that the proposed controller facilitates faster convergence of the tracking error with exponential decay in positional tracking errors compared to the classical geometric controller. Figures 5.5e and 5.5f depict the attitude errors $e_{R_d}(t)$ and angular velocity error $e_{\Omega_d}(t)$, respectively. One can observe that the errors $e_{R_d}(t)$ and $e_{\Omega_d}(t)$ are converging perfectly to the origin as claimed in Theorem 7. Figure 5.5g illustrates that the attitude error $e_{R_c}(t)$ converges to a very small ultimate bound (± 0.005) around the origin. Moreover, Figure 5.5h represents the error $e_{g_1}(t)$ between the second-time derivative of the desired trajectory and its estimate, demonstrating convergence to an ultimate bound (± 0.05) within the origin.

5.5 Conclusion

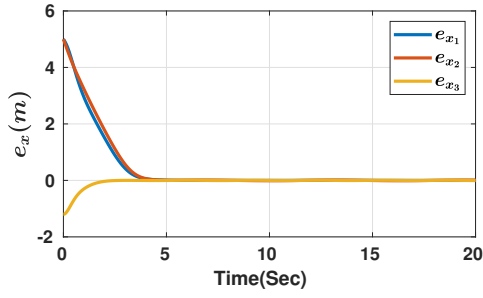
Chapter 5 presented a trajectory-tracking controller for multi-rotor UAVs with limited knowledge of the desired trajectory. By incorporating auxiliary filter dynamics, the controller effectively addresses the lack of higher derivatives of the desired trajectory, allowing for the hierarchical design of attitude and translational motion controllers. Theoretical analysis demonstrated that the thrust is always positive and saturated, ensuring the avoidance of singularities in the desired attitude construction. Furthermore, the controller was proven to be ultimately bounded in the extended error space, considering both attitude and position error dynamics. This work contributes a robust solution



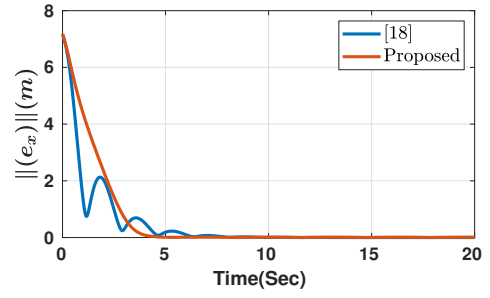
(a) Trajectory plot with proposed controller



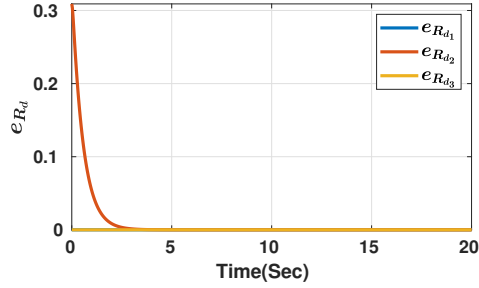
(b) Trajectory plot with controller in Lee *et al.* (2010)



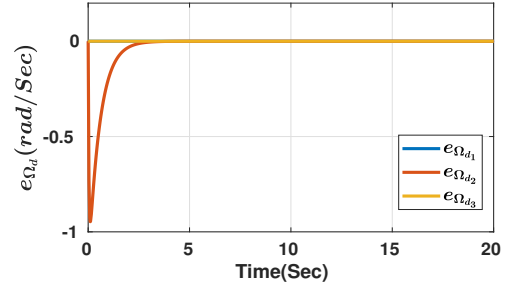
(c) Position error



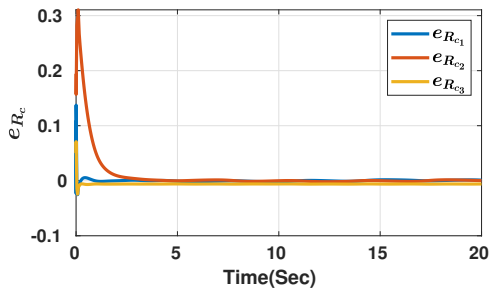
(d) Position error norm comparison plot



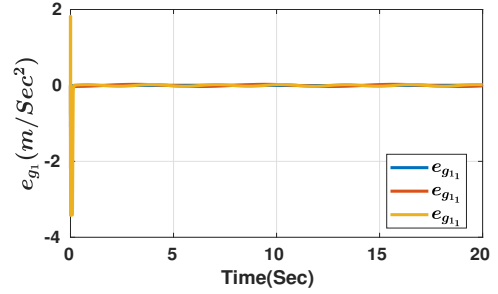
(e) $e_{R_d}(t)$



(f) $e_{\Omega_d}(t)$



(g) $e_{R_c}(t)$



(h) $e_{g1}(t)$

Figure 5.5: Simulation results for Example 2. [18] in there error norm plot represents plot with controller in Lee *et al.* (2010).

for trajectory tracking in UAVs with limited trajectory information, with rigorous guarantees on stability and performance.

Chapter 6

Asynchronous DNN-based Control for Systems with State and Control Input Dependent Uncertainties

This chapter presents a novel real-time learning control framework utilizing deep neural networks (DNNs) to address systems affected by state- and control input-dependent uncertainties. The proposed approach decomposes the DNN-based uncertainty approximation into two components: the outer layer weights, updated online at the control loop rate, and deep hidden layers of arbitrary depth, acting as basis functions. Using the inferred uncertainty from the DNN, we design a control input and introduce techniques to extract the control signal from non-affine control expressions.

Inspired by filtering techniques such as those in Roy *et al.* (2018), we approximate the actual uncertainty using a low-pass filter and develop a composite adaptation-based update law for the outer layer weights. To enhance learning, the deep basis functions are periodically trained using online stored filtered uncertainty data, and we compare this method with the generative approach in Joshi and Chowdhary (2019). The stability and convergence of the closed-loop system are rigorously analyzed through Lyapunov-based methods, ensuring uniform ultimate boundedness (UUB).

The efficacy of the proposed framework is validated through simulations and experiments on a quadrotor UAV. These results highlight the framework's ability to tackle coupled state- and control input-based uncertainties, such as the ground effect, demonstrating its practical utility.

6.1 Problem Formulation

Consider the following system for its relative simplicity, and its sufficiency in illustrating our general approach for DNN-based control design

$$\dot{x} = \Delta(x, u) + u, \tag{6.1}$$

where $x(t) \in \mathbb{X} \subset \mathbb{R}^n$, and $u(t) \in \mathbb{U} \subset \mathbb{R}^n$ denotes the system states and the control input. The state and input-dependent uncertainty is represented by the unknown and differentiable function $\Delta : \mathbb{X} \times \mathbb{U} \rightarrow \mathbb{R}^n$. We define the tracking error $e(t)$ as

$$e = x - x_d, \quad (6.2)$$

where $x_d(t)$ is the continuously differentiable reference trajectory. Given the uncertain dynamics (6.1), the control objective is to design a control input $u(t)$, which regulates the tracking error $e(t)$ to the origin in the presence of uncertainty $\Delta(x, u)$.

6.1.1 DNN Representation of Uncertainty

Since the uncertainty is structurally unknown, we leverage the universal approximation property of the deep neural networks Kidger and Lyons (2020) and denote $\Delta(x, u)$ using a L -layer ($L > 2$) DNN on the compact set $\mathbb{X} \times \mathbb{U}$ such that

$$\Delta(x, u) = W_L^T \Phi_L(x, u) + \varepsilon(x, u), \quad (6.3)$$

where $W_L \in \mathbb{R}^{n_L+1 \times n}$ denotes the ideal weight matrix of the output layer of the DNN with n_L neurons. $\Phi_L = \phi_L(W_{L-1}, \phi_{L-1}(\dots(W_1, \phi_1(W_0, [x^T, u^T]^T)))) \in \mathbb{R}^{n_L+1 \times 1}$ is the feature vector (or basis function) with last element being 1 corresponding to the bias term in the W_L Joshi and Chowdhary (2019). ϕ_i and W_{i-1} for $i = 1, 2, \dots, L$ denote the activation functions and ideal weights for the deep layers. $\varepsilon(x, u)$ denotes the function approximation error such that $\sup_{(x, u) \in \mathbb{X} \times \mathbb{U}} \|\varepsilon(x, u)\| \leq \bar{\varepsilon}$.

Real-time training of a DNN such as (6.3) to find the ideal weights is a challenging task for control applications as it is computationally demanding. Thus, we follow a semi-online training approach similar to Joshi and Chowdhary (2019) with suitable modification. We use two DNN structures to facilitate this. The first one, the main DNN, is used for real-time uncertainty inference and training outer-layer weights while keeping the inner-layer weights fixed. In contrast, the second one is used to perform offline training remotely (or in a separate thread from the main process) of inner-layer weights using a suitable data buffer, where the weights of the outer layer are fixed to those of the main DNN at the start of each training instance. Once the offline training

is completed, the inner-layer weights of the main DNN are updated with those of the secondary DNN.

To formally define this switching feature vector-based approach, let there be $j \in \mathbb{N}$ such instances where the weights of the basis function (inner layers) are updated with those of the secondary DNN upon the training completion. Let t_{j-1} be the time when the inner-layer weights of the main DNN are updated; we can write the actual uncertainty at any time $t \in [t_{j-1}, t_j)$ using a structurally same DNN that of given in (6.3) as follows

$$\Delta(x, u) = W_L^T \Phi_{Lj}(x, u, \Theta_j) + \varepsilon_j(x, u), \quad (6.4)$$

where $\Phi_{Lj}(x, u, \Theta_j) = \phi_L(\hat{W}_{L-1j-1}, \phi_{L-1}(\dots(\hat{W}_{1j-1}, \phi_1(\hat{W}_{0j-1}, [x^T, u^T]^T))))$ represents the basis function for the j^{th} switching instance with $\Theta_j = [\hat{W}_{0j-1}; \hat{W}_{1j-1}; \dots; \hat{W}_{L-1j-1}]$ being the learned weights of the basis function from the secondary DNN. For the sake of simplicity in the representation, we will only use Θ_j explicitly wherever deemed necessary in the expression of $\Phi_{Lj}(x, u, \Theta_j)$. Further, $\varepsilon_j(x, u) = W_L^T (\Phi_L(x, u) - \Phi_{Lj}(x, u)) + \varepsilon(x, u)$.

Assumption 8. *There exist positive constants $\bar{W}_k$ for $k = 1, \dots, n$ such that the columns $W_{Lk} \in \mathbb{R}^{n_L+1}$ of weight matrix W_L are bounded i.e. $\|W_{Lk}\| \leq \bar{W}_k$. Further, $\|\Phi_{Lj}(x, u)\| \leq \sigma, \forall j$ within the set $\mathbb{X} \times \mathbb{U}$.*

Property 1. *Given the compact set $\mathbb{X} \times \mathbb{U}$ there exist $\bar{\varepsilon}' > 0$ such that $\|\varepsilon_j(x, u)\| \leq \bar{\varepsilon}' \forall j$.*

Assumption 8 is standard in literature Gasparino *et al.* (2023); Joshi and Chowdhary (2019), and Property 1 can be derived from it, considering a bounded basis vector due to the bounded nature of activation functions. Note that the ideal weights may not be unique; however, to design the neural network-based controller, only the existence of the ideal weights is assumed, which is always guaranteed when $\Delta(x, u)$ is a continuous function within some compact set (Lewis *et al.*, 1996, Assumption 1). Next, we present the control design and a training routine for the DNN to effectively regulate the tracking error toward the origin.

6.2 Control Design

This section defines the control design and DNN training routine, including the outer layer weight matrix for the time $t \in [t_{j-1}, t_j)$, $\forall j$. An illustration of the control architecture is given in Figure 6.1. The proposed scheme is decomposed into two processes, namely, the control loop and training thread. The control loop implements the controller along with the main DNN-based uncertainty approximation and the outer-layer weight update laws. As discussed in Section 6.1.1, the basis function $\Phi_{Lj}(x, u)$ remains fixed for the time $t \in [t_{j-1}, t_j)$, $\forall j$ and updated only at time t_j when the training of the secondary DNN is completed. The training thread is responsible for training the secondary DNN based on the data buffer creation, which is discussed subsequently. Secondary DNN outer-layer weights are updated at every t_{j-1} before the start of the training and remain fixed throughout the training interval.

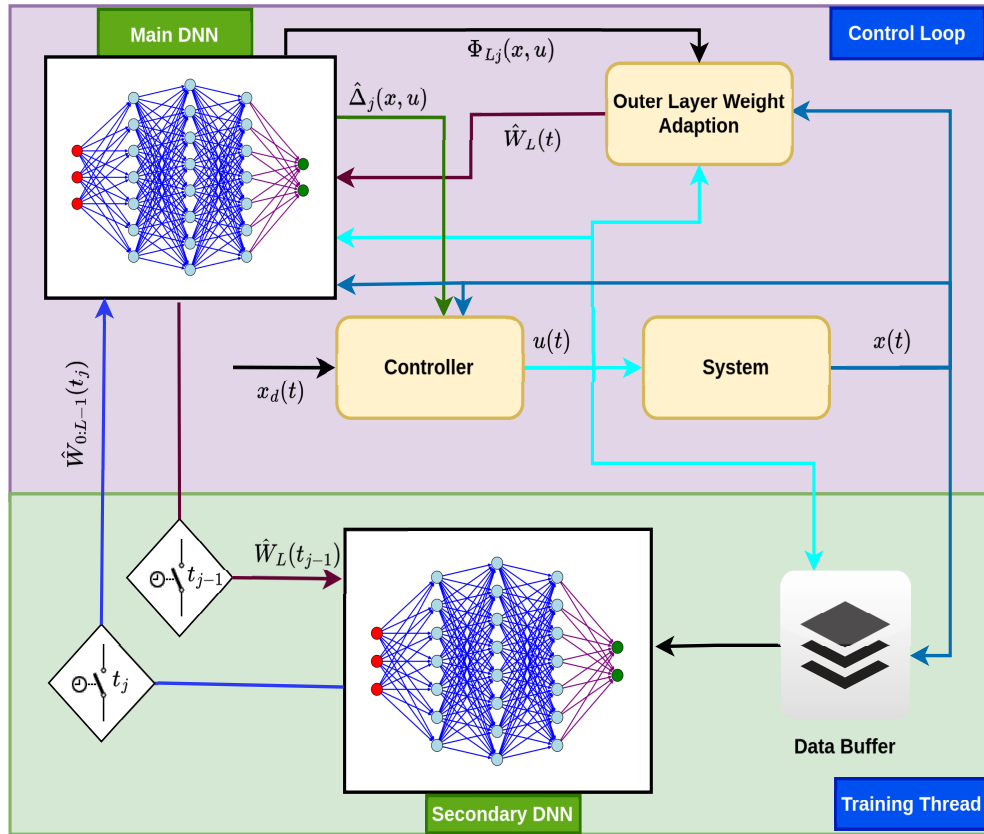


Figure 6.1: Illustration of control architecture.

6.2.1 Non-Affine Control Law

We design the control input $u(t)$ to be applied to the system at any time $t \in [t_{j-1}, t_j)$ as follows

$$u = -k_1 e - \hat{\Delta}_j(x, u) + \dot{x}_d, \quad (6.5)$$

where $k_1 > 0$ is the control parameter and $\hat{\Delta}_j(x, u)$ is the inferred uncertainty as shown in Figure 6.2

$$\hat{\Delta}_j(x, u) = \hat{W}_L^T \Phi_{Lj}(x, u). \quad (6.6)$$

Here $\hat{W}_L(t)$ is the estimate of ideal weight matrix W_L . Note that $\hat{W}_L(t)$ is updated in real-time while Φ_{Lj} is updated at the t_{j-1} upon the completion of training on the secondary DNN. At $t_0 = 0$, the weights of both DNNs are the same and randomly initialized, respecting the bounded weight assumption on the outer layer weights.

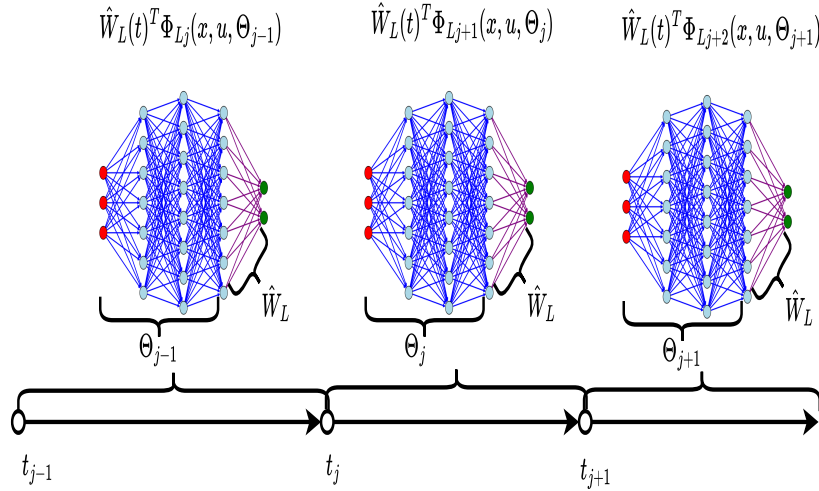


Figure 6.2: Illustration of switching of Basis function weights over t_j and uncertainty inference for any time $t \in [t_{j-1}, t_j)$.

Note that since $\hat{\Delta}_j(x, u)$ is a function of $u(t)$, it makes the control law $u(t)$ non-affine. This is a challenging problem from the implementation perspective. A detailed discussion on synthesizing the control input from this non-affine control law is presented at the start of the Results section. We next design the composite adaption-based update law for $\hat{W}_L(t)$.

6.2.2 Outer Layer Weight Update Law

Let's consider the open loop error dynamics of $e(t)$ using (6.2) and (6.4) for any time $t \in [t_{j-1}, t_j)$

$$\dot{e} = W_L^T \Phi_{Lj}(x, u) + \varepsilon_j(x, u) + u - \dot{x}_d. \quad (6.7)$$

To design a composite update law for outer layer weights $\hat{W}_L$ of the DNN, we consider the filter structures similar to Roy *et al.* (2018), for obtaining the tracking error derivative $\dot{e}(t)$. For the measurable signals $e(t)$, $u(t) - \dot{x}_d(t)$ and $\Phi_{Lj}(x, u)$, let's consider the following filter structures with filter design parameter $\gamma_1 > 0$

$$\gamma_1 \dot{e}_f + e_f = e, \quad e_f(t_{j-1}) = 0, \quad (6.8)$$

$$\gamma_1 \dot{u}_f + u_f = u - \dot{x}_d, \quad u_f(t_{j-1}) = 0, \quad (6.9)$$

$$\gamma_1 \dot{\Phi}_f + \Phi_f = \Phi_{Lj}(x, u), \quad \Phi_f(t_{j-1}) = 0. \quad (6.10)$$

Further consider the same filters for unknown signals $\Delta(x, u)$ and $\dot{e}(t)$ as

$$\gamma_1 \dot{g}_f + g_f = \dot{e}, \quad g_f(t_{j-1}) = 0, \quad (6.11)$$

$$\gamma_1 \dot{h}_f + h_f = W_L^T \Phi_{Lj}(x, u) + \varepsilon_j(x, u), \quad h_f(t_{j-1}) = 0. \quad (6.12)$$

We solve equation (6.11) using integration by parts to get

$$g_f(t) = \frac{1}{\gamma_1} \left(e(t) - e_f(t) - \exp\left(-\frac{1}{\gamma_1}t\right) e_f(t_{j-1}) \right). \quad (6.13)$$

Further substituting the (6.7) in (6.11) would yield

$$h_f = W_L^T \Phi_f + \varepsilon_f = g_f - u_f, \quad (6.14)$$

where ε_f is the filtered output of $\varepsilon_j(x, u)$ via

$$\gamma_1 \dot{\varepsilon}_f + \varepsilon_f = \varepsilon_j(x, u), \quad \varepsilon_f(t_{j-1}) = 0. \quad (6.15)$$

The relation (6.14) provides a measurable signal corresponding to filtered uncertainty h_f . We now design a prediction error for the composite adaption in the outer layer

weight update law. Let's define the following uncertainty predictor model $\hat{h}_f = \hat{W}_L^T \Phi_f$ and the corresponding prediction error

$$\zeta_f = h_f - \hat{h}_f = \tilde{W}^T \Phi_f + \varepsilon_f, \quad (6.16)$$

where $\tilde{W} = W_L - \hat{W}_L$. Equipped with an expression involving a measure of weight matrix error $\tilde{W}$, the outer layer weight matrix update law for any time $t \in [t_{j-1}, t_j)$, $\forall j \in \mathbb{N}$ is designed as follows

$$\dot{\hat{W}}_L = \text{proj}(\hat{W}_L, \Gamma(\Phi_{Lj}(x, u)e^T + k_2 \Phi_f \zeta_f^T)), \quad (6.17)$$

where $\text{proj}(\cdot)$ is the projection operator Landau *et al.* (2011) to ensure the column bounds similar to Assumption 8 and $k_2, \Gamma > 0$ are the design parameters.

6.2.3 Basis Function Training

The main objective of basis function training is to make $\Phi_{Lj}(x, u) \rightarrow \Phi_L(x, u)$ as j progresses, making the $\|\varepsilon_j\|$ smaller and thus yielding a smaller ultimate bound for the tracking error. Analogous to what is established in Joshi and Chowdhary (2019), given an input vector $[x^T, u^T]^T$, $\hat{\Delta}_j(x, u)$ serves as a decent label for training the feature vector or the inner layers of the DNN. Moreover, for a very small γ_1 , $h_f(t)$ can be made arbitrarily close to the uncertainty $\Delta(x, u)$ and hence can be a better choice for training the feature vector. In this chapter, we use both $\hat{\Delta}_j(x, u)$ and $h_f(t)$ as labels corresponding to the input vector $[x^T, u^T]^T$ and report the results.

To train the inner layers of the DNN, we create a buffer $Z = \{X_k, Y_k\}_{k=1}^{p_{max}}$ that can store p_{max} training samples of the input vector $X = [x^T, u^T]^T$ and label $Y = \hat{\Delta}_j(x, u)$ or $h_f(t)$. Once the buffer is full, new data samples are added following the standard techniques given in the literature Joshi and Chowdhary (2019); Hewing *et al.* (2019); Scholkopf *et al.* (1999). Once the buffer is full and a user-defined condition of when to start the training is met, we take a set Z_{train} with $p_{train}, p_{train} \leq p_{max}$ number of samples out of the whole buffer for training the DNN and use the following empirical loss function to

train the DNN

$$\mathcal{L}(Z_{train}, \hat{W}_{0j:L-1j}) = \frac{1}{p_{train}} \sum_{i=1}^{p_{train}} \|Y_i - \hat{Y}_i\|^2, \quad (6.18)$$

where $\hat{Y}_i = \hat{W}_L^T(t_{j-1})\Phi_{Lj}(x_i, u_i)$. Note that the outer layer weight matrix $\hat{W}_L^T(t_{j-1})$ remains fixed for the whole training period and copied from the main DNN at the start of the training. We refer the reader to (Joshi and Chowdhary, 2019, Section IV) for buffer creation and training details.

6.2.4 Lyapunov Analysis

We now show that the tracking error $e(t)$ remains uniformly ultimately bounded.

Theorem 9. *Consider the system dynamics (6.1) with control and outer weight update laws (6.5) and (6.17). The tracking error $e(t)$ is uniformly ultimately bounded.*

Proof. Let's consider the following Lyapunov candidate

$$V(e, \tilde{W}) = \frac{1}{2}e^T e + \frac{1}{2}tr(\tilde{W}^T \Gamma^{-1} \tilde{W}). \quad (6.19)$$

Taking time-derivative of $V(e, \tilde{W})$ using (6.7), (6.5), and (6.17) yields

$$\begin{aligned} \dot{V} = & e^T (-k_1 e + \tilde{W}^T \Phi_{Lj}(x, u) + \epsilon_j(x, u)) \\ & + tr(\tilde{W}^T proj(\hat{W}_L, \Gamma(\Phi_{Lj}(x, u)e^T + k_2 \Phi_f \zeta_f^T))), \\ & t \in [t_{j-1}, t_j), \forall j \in \mathbb{N}. \end{aligned} \quad (6.20)$$

Further, using (6.16) and projection theory Landau *et al.* (2011) yields

$$\begin{aligned} \dot{V} = & -k_1 \|e\|^2 - k_2 tr(\zeta_f \zeta_f^T - \zeta_f \epsilon_f^T) + e^T \epsilon_j(x, u), \\ & t \in [t_{j-1}, t_j), \forall j \in \mathbb{N}. \end{aligned} \quad (6.21)$$

Further using the fact that for any vectors a, b , $tr(aa^T) = \|a\|^2$ and $tr(ab^T) = a^T b$,

we get

$$\begin{aligned}\dot{V} &= -k_1\|e\|^2 - k_2\|\zeta_f\|^2 + k_2\zeta_f^T \varepsilon_f + e^T \varepsilon_j(x, u), \\ t &\in [t_{j-1}, t_j), \forall j \in \mathbb{N}.\end{aligned}\tag{6.22}$$

With Property 1, taking bounds using Cauchy inequality in equation (6.22) yields

$$\dot{V} \leq -\frac{k_1}{2}\|e\|^2 - \frac{k_2}{2}\|\zeta_f\|^2 + \frac{1}{2k_1}\bar{\varepsilon}'^2 + \frac{k_2}{2}\|\varepsilon_f\|^2, \quad t \in [t_0, \infty)\tag{6.23}$$

From filter structure (6.15), for a very small γ_1 and $\varepsilon_f(0) = 0$, we have $\|\varepsilon_f\| \leq \varepsilon'$. Thus,

$$\dot{V} \leq -\frac{k_1}{2}\|e\|^2 - \frac{k_2}{2}\|\zeta_f\|^2 + \left(\frac{1}{2k_1} + \frac{k_2}{2}\right)\bar{\varepsilon}'^2.\tag{6.24}$$

Using the fact that $\frac{1}{2}\|e\|^2 \geq V - \lambda_{\max}(\Gamma^{-1})\|\tilde{W}\|^2$ in (6.24) and further solving yields

$$\dot{V} \leq -k_1 V + \beta,\tag{6.25}$$

where $\beta = k_1 \lambda_{\max}(\Gamma^{-1})\|\tilde{W}\|^2 + \left(\frac{1}{2k_1} + \frac{k_2}{2}\right)\bar{\varepsilon}'^2$. Using the fact that $\|\tilde{W}\|$ is bounded due to Assumption 8 and differential weight update law with projection (6.17), relation (6.25) proves that $e(t)$ uniformly ultimately bounded Khalil (2002). $\square$

6.3 Results

To demonstrate the efficacy of the proposed scheme, we present simulation results that include highly nonlinear state and input-based uncertainties. Additionally, we provide experimental results of the proposed scheme applied to a quadrotor system in the Vi-con Arena¹. These results illustrate the robustness and practical applicability of our composite adaption-based DNN control method in handling complex uncertainties and environmental interactions.

¹The experiments were conducted in the Center for Autonomy Robotics Laboratories at the University of Illinois Urbana-Champaign.

6.3.1 Control Input Synthesis

To synthesize the control input from the non-affine control law (6.5), one can apply Newton's root-finding method for a given $x(t)$ to find $u(t)$ such that

$$f(x, u) = u + k_1 e + \hat{\Delta}_j(x, u) - \dot{x}_d = 0. \quad (6.26)$$

However, the root-finding method may exhibit slow convergence with stringent convergence guarantee conditions. An alternative approach is to use u_{k-1} in calculating u_k in a sampled data sense Shi *et al.* (2019) assuming $f(x, u) - u$ to be a contraction mapping. Further, a continuous time solution method to define a differential control law based on an auxiliary control input is as follows

$$\begin{aligned} u_a &= -k_1 e - \hat{\Delta}_j(x, u) + \dot{x}_d, \\ \dot{u} &= \frac{1}{\gamma_2} (u_a - u). \end{aligned} \quad (6.27)$$

The essence of the above differential law is that, by setting $\gamma_2 > 0$ to a very small value, u can be brought sufficiently close to u_a . With this, we define three methods to obtain the solution of (6.5) as follows

- **Method 1** - In this method, the input from the last time step is used as an input to (6.5) on the right-hand side to compute the $u(t)$ at the current time step Shi *et al.* (2019).
- **Method 2** - We solve differential control law (6.27) to obtain the control input.
- **Method 3** - For the real-time implementation of Newton's method, a large number of iterations may introduce delays in control input, and sometimes convergence is not achieved. Therefore, Method 3 prioritizes calculation time by limiting the number of iterations and switching to Method 1 if convergence is not achieved within these iterations.

A comprehensive analysis of these methods is beyond the scope of this chapter. Instead, we used simulations as a surrogate to verify their performance, which demonstrated that all three methods provide comparable tracking results. The choice of approach is left to the user's preference, with a detailed theoretical analysis reserved for future research.

6.3.2 Simulation Results

First, we present the simulation results obtained from a synthetic example implemented in Python using Google Colab. We consider a system with $n = 2$ i.e. $x(t) = [x_1, x_2]^T \in \mathbb{R}^2$, $u(t) = [u_1, u_2] \in \mathbb{R}^2$ with $f(x, u) = -0.1u + [x_1x_2 \tanh(x_2) + \text{sech}(u_1)u_2, u_1 \text{sech}^2(x_2) + \text{sech}^2(u_1 + u_2)]^T$. Note that the second term of the given $f(x, u)$ represents a non-affine control input and state-based uncertainty, while the first term models actuator inefficiency, i.e., 90% controller effectiveness. The DNN consists of four hidden layers with $n_1 = 20$ and $n_{2,3,4} = 50$ neurons, respectively, employing hyperbolic tangent activation functions at each layer. The simulation parameters are set as follows: $k_1 = 2$, $\Gamma = 0.7I_{2 \times 2}$, $k_2 = 0.5$, $\gamma_1 = 0.01$, $\gamma_2 = 0.01$. The buffer and batch sizes are $p_{max} = p_{train} = 200$, with new secondary DNN training starting once 50 new data samples are added to the buffer once it's full. We store the data in the buffer using the singular value maximization approach Chowdhary and Johnson (2011). The learning rate is kept at 0.01 with 150 epochs to train the secondary DNN. Further, the simulation results are obtained using the RK4 method with a sampling time of 0.01 sec. The reference trajectory is set as $x_d(t) = [\sin(2t), -\cos(t)]^T$. We compile the simulation results with all three methods to solve the non-affine equation (6.5). For each case, we present the results with for composite adaptive outer-layer weight update law (6.17) with both types of inner-loop training dataset labels $\hat{\Delta}_j(x, u)$ and $h_f(t)$.

The results using Method 1 with different DNN weight update rules are shown in Figure 6.3. We briefly clarify the data representation in the figures. For the plot showing the tracking error, i.e., Figure 6.3a, the blue and red curves represent the tracking errors for the cases when update law (6.17) is used with inner-layer training labels $Y = \hat{\Delta}_j(x, u)$ and $Y = h_f(t)$, respectively. Further, we compare these results with an outer-layer weight update law without composite adaption, i.e., $\dot{\hat{W}}_L = \text{proj}(\hat{W}_L, \Gamma \Phi_{Lj}(x, u)e^T)$ and inner-loop training labels $Y = \hat{\Delta}_j(x, u)$, by plotting the norm of tracking errors in Figure 6.3b. Similarly, the results using Methods 2 and 3 are shown in Figures 6.4 and 6.5. Further, Table 6.1 shows the root mean square of the tracking error norms under different outer-layer weight updates and the inner-layer training labels for all the control synthesis methods. The plots show that the tracking error norm for cases where composite adaptation is used is significantly less (also evident from Table 6.1) than in cases without composite adaptation, demonstrating a notably better tracking performance. The

results suggest that composite adaptation improves the ultimate bound of the tracking error and enhances the system's transient and steady-state response.

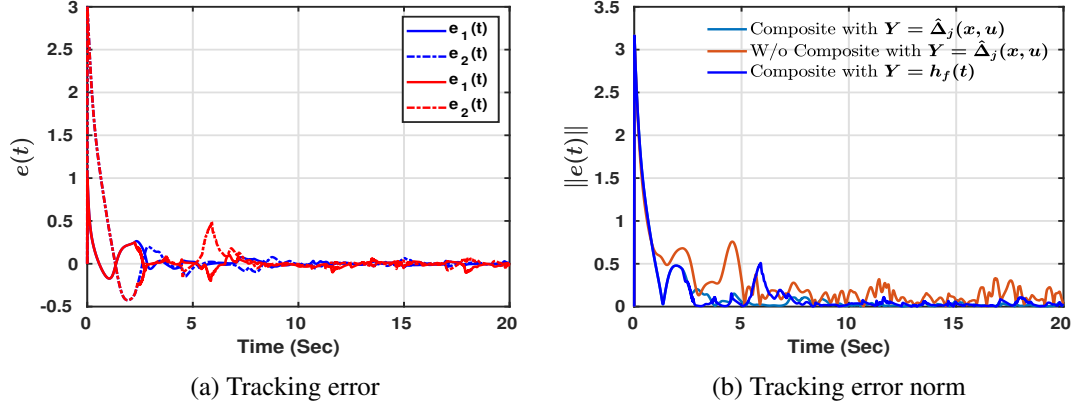


Figure 6.3: Simulation results using Method 1

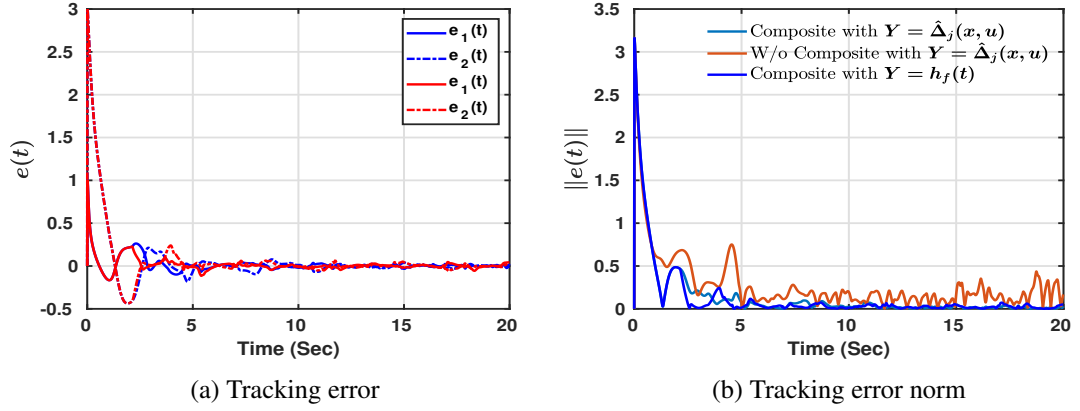


Figure 6.4: Simulation results using Method 2

Method	Composite with $Y = \hat{\Delta}_j(x, u)$	Composite with $Y = h_f(t)$	Without Composite with $Y = \hat{\Delta}_j(x, u)$
1	0.3743	0.3803	0.4354
2	0.3786	0.3747	0.4401
3	0.3922	0.3807	0.4584

Table 6.1: Root mean square of tracking error norm for Methods 1, 2, and 3.

Furthermore, the results in Table 6.1 show that with Method 2 and 3 and the basis function, training labels $Y = h_f(t)$ one gets slightly superior performance ($\approx 1\%$ improvement) compared to training with labels $Y = \hat{\Delta}_j(x, u)$ whereas for Method 1 its opposite. Although the performance improvement is insignificant, this comparison strongly supports the empirical feasibility of $\hat{\Delta}_j(x, u)$ being a good choice for basis function training. Further, One can observe that Methods 2 and 3 yield better results than

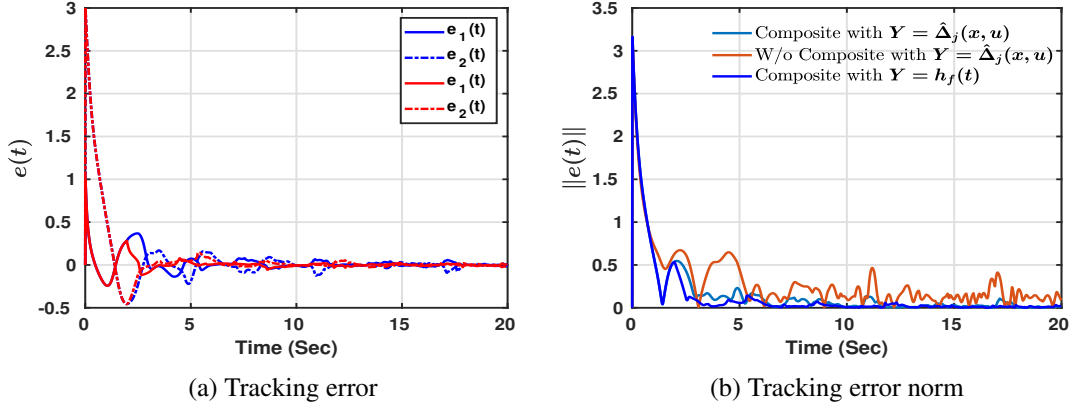


Figure 6.5: Simulation results using Method 3

Method 1 in the steady state. This highlights both Method 2 and 3's effectiveness in achieving more precise solutions of non-affine control law.

Effect of Basis Function Learning

To demonstrate the performance improvements achieved through the proposed offline basis function learning, we present a comparison of three scenarios based on the use of basis learning for the given example. In each case, the training labels are defined as $Y = \hat{\Delta}_j(x, u)$, and the control synthesis follows non-affine Method 3.

Figure 6.6 depicts the tracking error norm for the three scenarios: (1) continuous basis learning, (2) single-instance basis learning, and (3) no basis learning. The empirical results clearly highlight the significant reduction in tracking error when basis learning remains active throughout, demonstrating its effectiveness in enhancing control performance.

6.3.3 Experimental Results

We perform experiments as proof of concept of the proposed scheme using the Bitcraze Crazyflie nano quadcopter platform Giernacki *et al.* (2017) using the CrazySim Llanes *et al.* (2024) and ROS2. The experimental setup within a Vicon environment is shown in Figure 6.7. As dynamics level attitude control is not possible with the Crazyflie, we consider two examples with different uncertainties in the height dynamics of the quadrotor to show the efficacy of the proposed scheme. The translational dynamics of

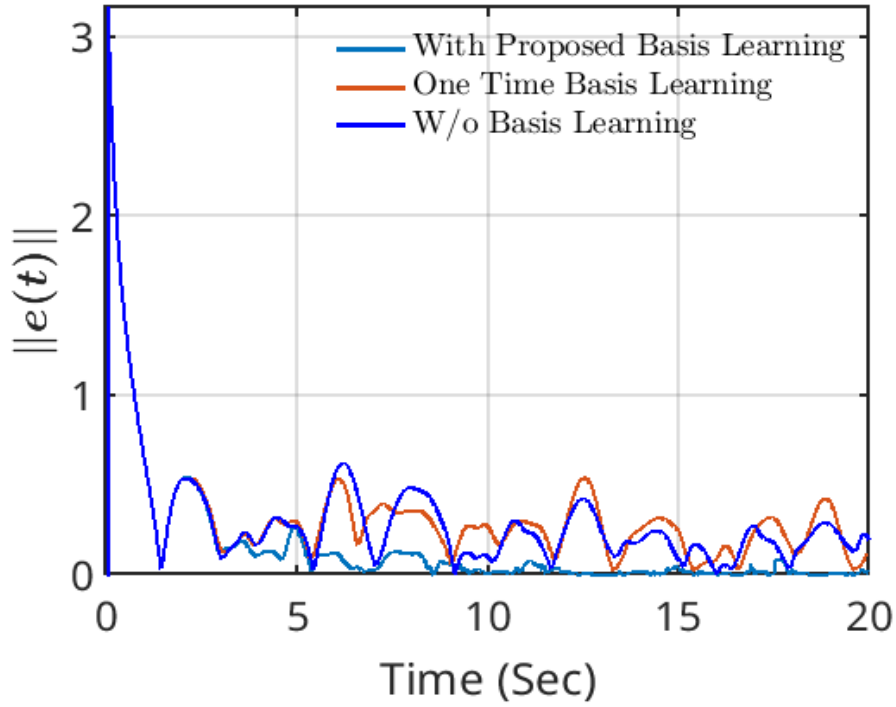


Figure 6.6: Tracking error norms with different basis function learning settings.

the quadrotor can be given as

$$\dot{p} = v \quad x = [p^T, v^T]^T \quad (6.28)$$

$$\dot{v} = uRe_3 - mge_3 + \Delta(x, u), \quad (6.29)$$

where $p(t) = [p_x(t), p_y(t), p_z(t)]^T \in \mathbb{R}^3$ is the position and $v(t) \in \mathbb{R}^3$ is the linear velocity of the quadrotor. The thrust $u(t)$ acts as the control input to the translational dynamics. Further, $\Delta(x, u)$ represents the input and state-based uncertainties, and m denotes the mass. g denotes the gravitational acceleration, $R \in SO(3)$ is the orientation matrix and $e_3 = [0, 0, 1]^T$. For a given desired position reference $p_d(t)$, we formulate an error variable $s = e_v + K_p e_p$, where $e_v = (v - \dot{p}_d)$ is the velocity tracking error and $e_p = (p - p_d)$ is the position tracking error. Following the velocity dynamics given in (6.29), the velocity tracking error dynamics multiplied with mass take the following form

$$m\dot{e}_v = uRe_3 - mg + \Delta(x, u) - m\ddot{p}_d. \quad (6.30)$$

Following the similar step as Section 6.2.2, we first get $h_f(t)$ by filtering the error

dynamics (6.30). The control input $u(t)$ and the differential outer-layer weight update law is then designed as follows

$$u = u_d.Re_3, \quad u_d = -K_s s + mge_3 + m\ddot{p}_d - \hat{\Delta}(x, u). \quad (6.31)$$

$$\dot{\hat{W}}_L = \text{proj}(\hat{W}_L, \Gamma(\Phi_{Lj}(x, u)s^T + k_2\Phi_f\zeta_f^T)), \quad (6.32)$$

We further derive the desired attitude $R_d(t) \in SO(3)$ utilizing the direction of $u_d(t)$ as the desired z-body fixed axis direction (third column of $R_d(t)$) of the quadrotor to achieve desired position tracking. The rest of the R_d columns are derived using standard unit vector operations given a desired yaw vector. We refer the reader to (O’Connell *et al.*, 2022, Robust Adaptive Controller Design) for the details on designing a neural network-based controller. The Crazyflie platform is controlled by sending the thrust input and desired attitude setpoint commands through `/cmd_attitude_setpoint` ROS2 topic at 50 Hz sampling rate.

We consider a DNN with 3 hidden layers with $n_{1,2} = 30$ and $n_3 = 30$ neurons and tan hyperbolic activation function at each layer. The following parameters are considered for implementing outer layer weight update law $\Gamma = 0.08I_{3 \times 3}$, $k_2 = 0.05$, and $\gamma_1 = 0.05$. For the inner layer training, the learning rate is 0.001 with 300 epochs, and the buffer and batch sizes are $p_{max} = p_{train} = 200$, with new secondary DNN training starting once 100 new data samples are added to the buffer once it’s full. We use a cyclic update law for the buffer update and Method 1 to obtain the solution for the non-affine control law. A column-wise discrete projection Gasparino *et al.* (2023) is applied with the following bounds $[0.001, 0.001, 0.035]$. Note that the very small bound values for weights responsible for the first two elements of the $\hat{\Delta}_j(x, u)$ (x and y linear motion of the quadrotor) are used, assuming that uncertainties in both examples affect only the height dynamics of the quadrotor. To test the efficacy of the proposed DNN-based controller, we design the following desired trajectory

$$p_d(t) = \begin{cases} [-0.1, -0.1, p_{z_d}]^T, & \text{if } 0 < t \leq 15, \\ [\sin(0.2t), -1 + \cos(0.2t), p_{z_d}]^T, & \text{otherwise} \end{cases}$$

where p_{z_d} is the constant desired height specified differently for each example. We also use a baseline controller without $\hat{\Delta}_j(x, u)$ term to compare our results. Also, for both

examples, the baseline controller is active in the first segment, i.e., $t \leq 15 \text{ sec}$. The proposed controller is then switched on for the rest of the duration of the experiment. A video containing the experimental results can be found [here](#)-Video.

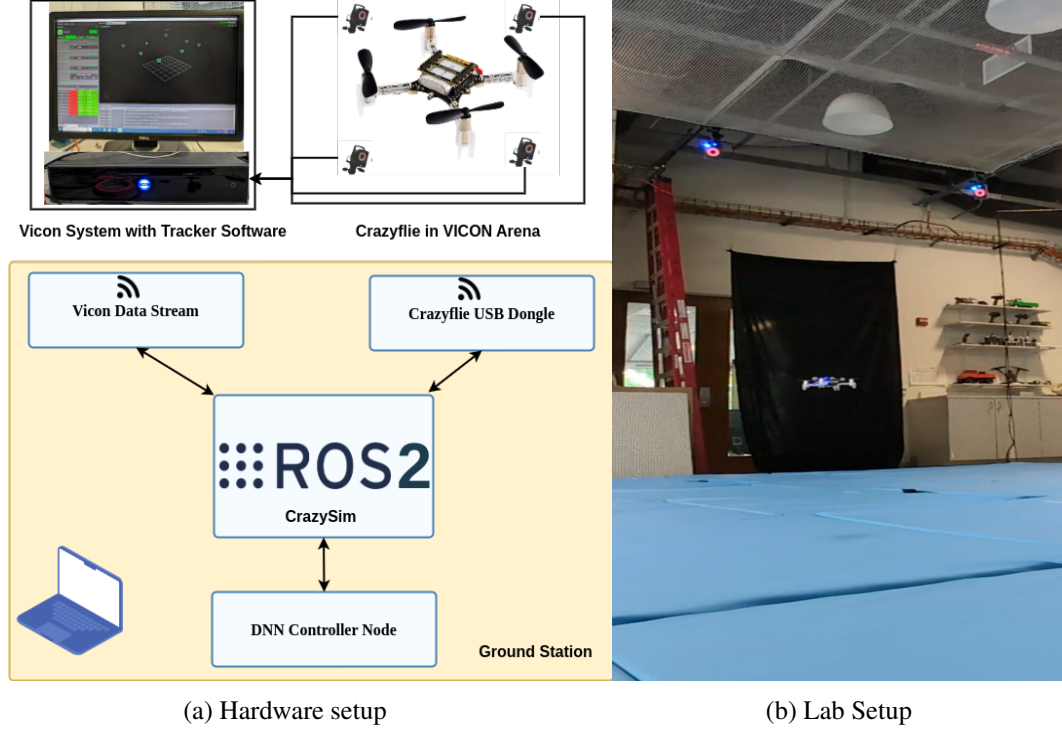
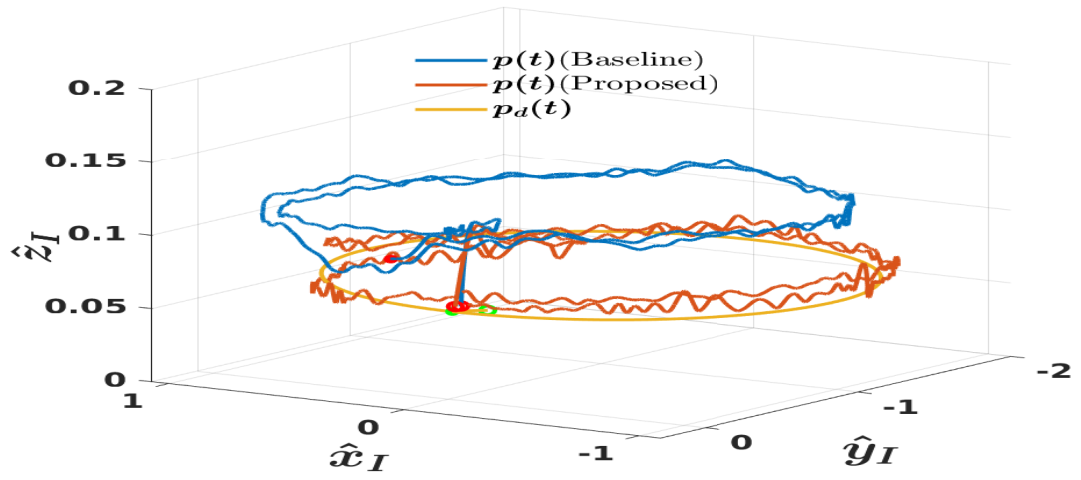


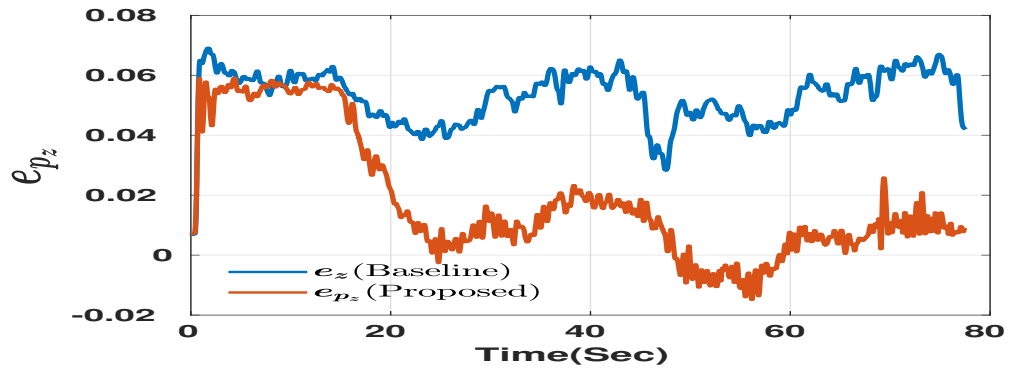
Figure 6.7: Experimental setup

Case Study 1: Ground Effect Compensation

The first example is motivated by the existence of ground effect phenomena when the quadrotor is flying near the ground. For this example, we consider p_{z_d} to be 0.06 m and run the experiments with both the baseline and the proposed scheme. The results are shown in Figure 6.8 in the form of the trajectory plot and the height tracking error of the quadrotor. As shown in the trajectory plot 6.8a, the proposed controller enables the quadrotor to maintain flight close to the desired height, whereas the baseline controller fails to achieve this level of accuracy. This demonstrates the effectiveness of the proposed scheme in compensating for the ground effect and achieving better height-tracking performance. We want to emphasize that the low magnitude of the ground effect and, hence, small deviation in height error is due to the small size of the rotor on the Crazyflie, which measures just 45 mm.



(a) Trajectory Plots

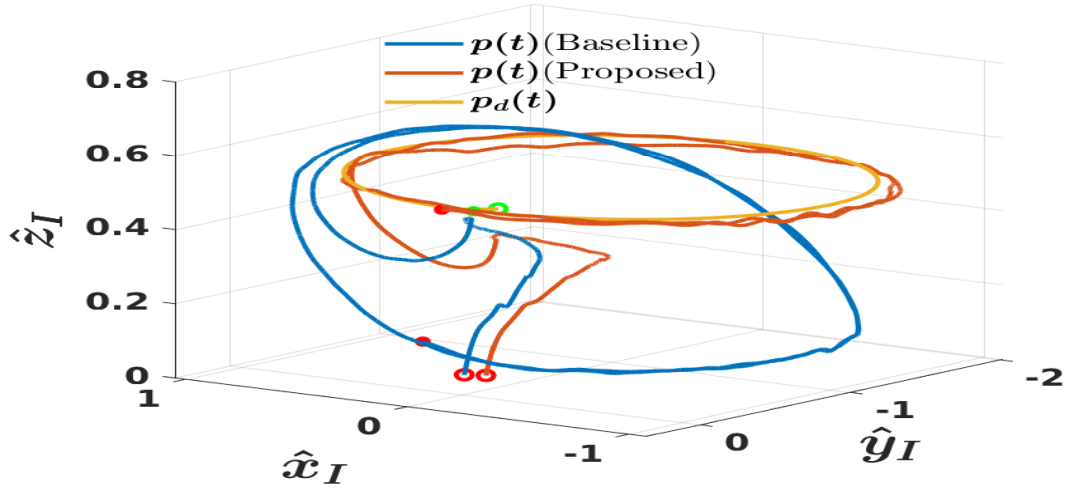


(b) Height error($e_{p_z} = p_z - p_{z_d}$)

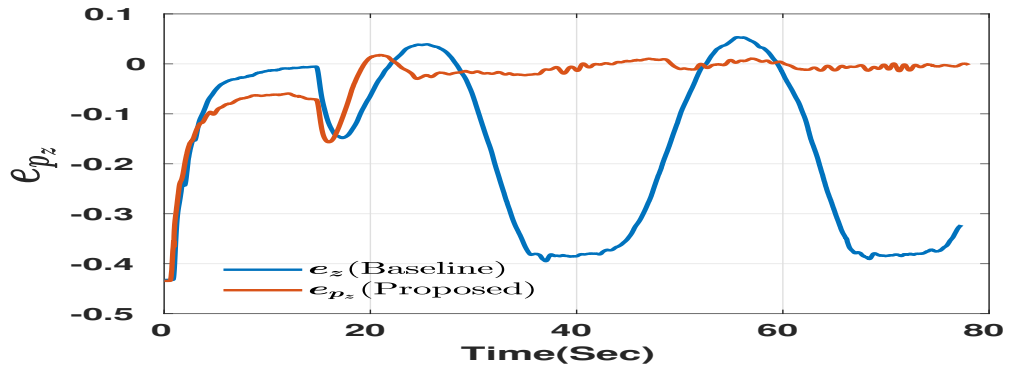
Figure 6.8: Example 1 results

Case Study 2: Artificial Uncertainty Compensation

To test the performance of the proposed controller rigorously, we inject an artificial uncertainty of $[0, 0, 0.05\sin(p_x(t))]^T$ in the calculation of $u(t)$ along with an assumption that only the 90 percent of the thrust input $u(t)$ is delivered to the quadrotor. For this experiment, we consider p_{zd} to be 0.5 m. Figure 6.9 presents the results for this example. From Figure 6.9b, it is evident that with the given uncertainty, the magnitude of the height error fluctuates up to 0.4 m with the baseline controller. In contrast, the height error remains within an ultimate bound of ± 0.02 m with the proposed scheme. This illustrates the robustness and effectiveness of the proposed controller in mitigating uncertainties and achieving precise height control.



(a) Trajectory Plots



(b) Height error ($e_{p_z} = p_z - p_{zd}$)

Figure 6.9: Example 2 results

6.4 Conclusion

This chapter introduced a real-time DNN-based control framework designed to handle both state and control input-based uncertainties. The control problem was approached using a neuro-adaptive design, where uncertain parameters, such as outer-layer weights and inner-layers of the DNN, were treated as the basis functions. A non-affine control law utilizing DNN as an uncertainty approximator, combined with a semi-online DNN training methodology, was developed. Rigorous Lyapunov analysis was employed to ensure uniform ultimate boundedness (UUB) of the closed-loop error dynamics, guaranteeing stability and convergence. The approach's effectiveness was validated through extensive simulations and proof-of-concept experiments on a quadrotor platform, showing significant improvements in both transient and steady-state tracking performance. This work demonstrates the potential of DNN-based control strategies for managing uncertainties in real-time applications.

Chapter 7

Vision-Based Dynamic Control for Quadrotor Tracking and Landing on Moving Targets

This chapter presents an IBVS-based control design for a quadrotor tasked with tracking and landing on a moving planar target exhibiting arbitrary and unknown motion in the presence of uncertainties. Building on the developments from Chapter 3, we first frame the landing problem as an IBVS problem. The control design process begins with the translational dynamics, incorporating target motion estimation, uncertainty mitigation, and the derivation of the scalar thrust input required for stable tracking and landing.

Next, the desired heading is derived by reconstructing the target's heading, following the methodologies outlined in Chapter 4. This is used to construct the desired orientation matrix, the reference for the attitude-tracking controller. The attitude controller design follows the geometric control principles discussed in Chapter 5. Assuming sufficiently fast attitude tracking, we provide a convergence analysis for the translational dynamics, demonstrating ultimately bounded tracking and estimation errors. Finally, simulation results validate the proposed control framework's effectiveness and highlight its practical utility in real-world scenarios involving dynamic target tracking and landing.

7.1 Problem Formulation

7.1.1 Quadrotor Dynamics

Let $\{I\}$, $\{B\}$, and $\{C\}$ be inertial, the quadrotor body frames, and the camera frame, as illustrated in Figure 7.1. For a quadrotor configuration defined by the inertial frame position $x(t) = [x_1(t), x_2(t), x_3(t)]^T \in \mathbb{R}^3$ of the center of mass and attitude $R(t) = [\hat{x}_B, \hat{y}_B, \hat{z}_B] \in \text{SO}(3)$ expressed in $\{I\}$, the dynamics of the quadrotor can be described

by the following equations

$$\dot{x} = v, \quad (7.1)$$

$$m\dot{v} = mge_3 - fRe_3 + \Delta, \quad (7.2)$$

$$\dot{R} = R\hat{\Omega}, \quad (7.3)$$

$$J\dot{\Omega} = -\Omega \times J\Omega + M, \quad (7.4)$$

where m denote the mass of the quadrotor, J denote the inertia matrix expressed in $\{B\}$, and g denote the gravitational acceleration. Along with the configuration states $x(t)$ and $R(t)$, inertial frame velocity $v(t) \in \mathbb{R}^3$, and the body-fixed angular velocity $\Omega(t) \in \mathbb{R}^3$ represent the multi-rotor states. Further, $e_3 = [0, 0, 1]^T$ and the term $\Delta \in \mathbb{R}^3$ represent the unknown disturbances arising due to the complex aerodynamic interactions of the quadrotor UAV with the target and the environment. The total thrust $f \in \mathbb{R}$ and moment $M = [M_1, M_2, M_3]^T \in \mathbb{R}^3$ expressed in the body frame are considered as the control inputs to the multi-rotor model. Given a pair of total thrust and moment, one can obtain the individual rotor speeds to design the motors' low-level pulse width modulation (PWM) control signals. For a comprehensive understanding of quadrotor dynamics, readers should delve into the detailed discussions provided in references such as Lee *et al.* (2010); Mahony *et al.* (2012).

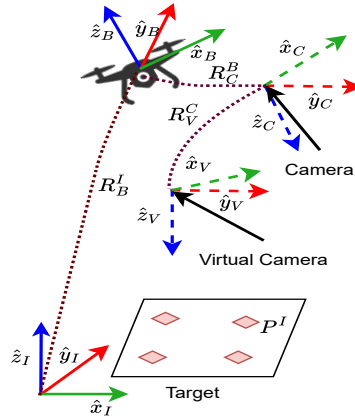


Figure 7.1: Coordinate frames illustration. The x, y, z axes of a coordinate frame $\{A\}$ are denoted by $\hat{x}_A, \hat{y}_A$ and $\hat{z}_A$, respectively.

To pose the quadrotor landing on the moving target task as an IBVS problem, we

next define the image features that yield dynamically feasible kinematics and the control inputs.

7.1.2 IBVS Preliminaries for Quadrotor

We utilize the virtual image plane invariant to quadrotor roll (ϕ) and pitch (θ) angles and define image moment-based features to design the controller. The procedures for constructing the virtual image plane and determining the image moment-based features are elaborated upon in the previous Chapter 3. Here, we will present only the essential expressions. Let $\{V\}$ be the virtual camera frame corresponding to the virtual image plane as represented in Fig. 7.1. Note that the origin of the camera frame $\{C\}$ and the virtual camera frame $\{V\}$ coincide. We reconstruct the virtual image plane coordinates of a 3-D point P^I , expressed in the inertial frame with its image plane point feature coordinates $s = [u, v]^T$, for a body fixed camera as follows

$$s_v = \begin{bmatrix} u_v \\ v_v \end{bmatrix} = \frac{f'}{\bar{R}_3 \bar{p}} \begin{bmatrix} \bar{R}_1 \bar{p} \\ \bar{R}_2 \bar{p} \end{bmatrix}, \quad (7.5)$$

where $\bar{p} = [u - c_x, v - c_y, f']^T$ and $\bar{R}_i, i = 1, 2, 3$ denotes the i^{th} row of the rotation matrix $\bar{R} = R_C^{B^T} R_y(\theta) R_x(\phi) R_C^B$ which represents the rotation from $\{C\}$ to $\{V\}$. (c_x, c_y) is the principal point offset of the camera and f' is the focal length of the camera in pixels. The matrix R_C^B represents the rotation from $\{C\}$ to $\{B\}$, which remains constant as the camera has a fixed pose with respect to the quadrotor body frame. One can compute this matrix by analyzing the fixed camera attachment with the quadrotor body frame.

The landing target is assumed to be planar with N observable virtual image plane visual points $s_{v_k} = [u_{v_k}, v_{v_k}]^T$, $k = 1, 2, \dots, N$. We define an image features set $s'(t) = [x_n(t), y_n(t), a_n(t)] \in \mathbb{R}^3$, and a scalar feature $\alpha(t)$ for controlling the translational and heading motions of the multirotor, respectively as follows

$$x_n = \frac{a_n}{f'} x_g, \quad y_n = \frac{a_n}{f'} y_g, \quad a_n = h_d \sqrt{\frac{a_d}{a}}, \quad (7.6)$$

$$\alpha = \frac{1}{2} \arctan \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right), \quad (7.7)$$

where $[x_g, y_g]^T = [\frac{1}{N} \sum_{k=1}^N u_{v_k}, \frac{1}{N} \sum_{k=1}^N v_{v_k}]^T$ and $a = \sum_{k=1}^N ((u_{v_k} - x_g)^2 + (v_{v_k} - y_g)^2)$.

$\mu_{ij} = \sum_{k=1}^N ((u_{v_k} - x_g)^i (v_{v_k} - y_g)^j)$ is the centered moment of order $i + j$. a_d is the desired value of a , and h_d is the desired height of the virtual camera frame from the planar target.

For a target, moving with translational velocity $\bar{v}_p(t) \in \mathbb{R}^3$ and angular rate $[0, 0, \dot{\psi}_p(t)]^T \in \mathbb{R}^3$, expressed in $\{V\}$, the feature dynamics is given as

$$\dot{s}' = -S_k([0, 0, \dot{\psi}]^T) s' - (\bar{v} - \bar{v}_p), \quad (7.8)$$

$$\dot{\alpha} = -\dot{\psi} + \dot{\psi}_p, \quad (7.9)$$

where $[0, 0, \dot{\psi}(t)]^T \in \mathbb{R}^3$ and $\bar{v}(t) \in \mathbb{R}^3$, are the angular rate and the linear velocity of the virtual camera $\{V\}$, expressed in the $\{V\}$.

Remark 10. *This work considers an appropriate size marker, visible even after the quadrotor's landing on the target. In practice, these would be small and might not be detectable by the state-of-the-art detection algorithms from high altitudes. Using an embedded marker method, the solution to enabling the quadrotor to land on the moving target from higher altitudes is discussed in Yang et al. (2024). This solution treats the bigger marker as the main marker for height descent until the smaller marker is visible. Afterward, the smaller marker is considered the main marker, and the landing task is achieved. Note that the IBVS scheme is kept the same during the whole process. Following the same, we want to emphasize to the practitioners that a similar switching strategy can be used with straightforward adaption of our proposed IBVS scheme for high-altitude difference landings.*

7.1.3 Landing as an IBVS problem

Let's consider that the engagement of the quadrotor and target is such that upon landing, the target centroid is exactly at the center of the image plane, i.e., $x_n^* = 0, y_n^* = 0$. Here x_n^* and y_n^* are the desired for the image features x_n and y_n . Hence, we formally define the $s'^* = [0, 0, h_d]$ as a desired of s' , where h_d is the height of the camera frame (or virtual camera frame) with respect to the target upon landing. We consider such an arrangement where the quadrotor must maintain a relative heading with the target. The formulation formally introduces this by defining a desired α^* for heading feature α . One can calculate the desired α^* based on the given desired feature points (obtained

using the desired image) in the virtual image plane using the relation (7.7). Let's define the feature-tracking errors as

$$e_s(t) = s'(t) - s'^*, \quad (7.10)$$

$$e_\alpha(t) = \alpha(t) - \alpha^*. \quad (7.11)$$

The corresponding feature dynamics is given as

$$\dot{e}_s = -S_k([0, 0, \dot{\psi}]^T)e_s - (\bar{v} - \bar{v}_p), \quad (7.12)$$

$$\dot{e}_\alpha = -(\dot{\psi} - \dot{\psi}_p). \quad (7.13)$$

Control Objective: Given the feature errors dynamics (7.12), (7.13) and the multirotor dynamics (7.2)-(7.4), the control objective is to design the control input pair (f, M) such that $e_s(t), e_\alpha(t)$ converges towards origin i.e. achieving the landing task, despite the unknown and arbitrary target motion and the disturbances in the translational dynamics.

7.1.4 Translational Dynamics Revisited

Now that we will work with the visual features s' , let's revisit the translational motion dynamics. The dynamics can then be given as

$$\dot{s}' = -S_k([0, 0, \dot{\psi}]^T)s' - (\bar{v} - \bar{v}_p), \quad (7.14)$$

$$m\dot{v} = mge_3 - fRe_3 + \Delta. \quad (7.15)$$

Consider the states $X = [s'^T, v^T]^T \in \mathbb{X} \subset \mathbb{R}^6$ and control input $U = f \in \mathbb{U} \subset \mathbb{R}$. We treat Δ as an uncertainty dependent on both the state and control input, denoted henceforth as $\Delta(X, U)$ Shi *et al.* (2019). This representation encompasses various scenarios: saturation, dead zones, hysteresis, actuator ineffectiveness, and environmental variations. A notable example is the ground effect in quadrotors while landing Shi *et al.* (2019).

DNN Representation of Uncertainty

Since the uncertainty is structurally unknown, we leverage the universal approximation property of the deep neural networks Kidger and Lyons (2020) and denote $\Delta(X, U)$ using a L -layer ($L > 2$) DNN on the compact set $\mathbb{X} \times \mathbb{U}$ such that

$$\Delta(X, U) = W_L^T \Phi_L(X, U) + \varepsilon(X, U), \quad (7.16)$$

where $W_L \in \mathbb{R}^{n_L+1 \times 6}$ denotes the ideal weight matrix of the output layer of the DNN with n_L neurons. $\Phi_L = \phi_L(W_{L-1}, \phi_{L-1}(\dots(W_1, \phi_1(W_0, [X^T, U^T]^T)))) \in \mathbb{R}^{n_L+1 \times 1}$ is the feature vector (or basis function) with last element being 1 corresponding to the bias term in the W_L Joshi and Chowdhary (2019). ϕ_i and W_{i-1} for $i = 1, 2, \dots, L$ denote the activation functions and ideal weights for the deep layers. $\varepsilon(X, U)$ denotes the function approximation error such that $\sup_{(X, U) \in \mathbb{X} \times \mathbb{U}} \|\varepsilon(X, U)\| \leq \bar{\varepsilon}$.

Real-time training of a DNN such as (7.16) to find the ideal weights is a challenging task for control applications as it is computationally demanding. Thus, we follow a semi-online training approach similar to Joshi and Chowdhary (2019) with suitable modification. We use two DNN structures to facilitate this. The first one, the main DNN, is used for real-time uncertainty inference and training outer-layer weights while keeping the inner-layer weights fixed. In contrast, the second one is used to perform offline training remotely (or in a separate thread from the main process) of inner-layer weights using a suitable data buffer, where the weights of the outer layer are fixed to those of the main DNN at the start of each training instance. Once the offline training is completed, the inner-layer weights of the main DNN are updated with those of the secondary DNN.

To formally define this switching feature vector-based approach, let there be $j \in \mathbb{N}$ such instances where the weights of the basis function (inner layers) are updated with those of the secondary DNN upon the training completion. Let t_{j-1} be the time when the inner-layer weights of the main DNN are updated; we can write the actual uncertainty at any time $t \in [t_{j-1}, t_j)$ using a structurally same DNN that of given in (7.16) as follows

$$\Delta(X, U) = W_L^T \Phi_{Lj}(X, U, \Theta_j) + \varepsilon_j(X, U), \quad (7.17)$$

where $\Phi_{Lj}(X, U, \Theta_j) = \phi_L(\hat{W}_{L-1j-1}, \phi_{L-1}(\dots(\hat{W}_{1j-1}, \phi_1(\hat{W}_{0j-1}, [X^T, U^T]^T))))$ represents the basis function for the j^{th} switching instance with $\Theta_j = [\hat{W}_{0j-1}; \hat{W}_{1j-1}; \dots; \hat{W}_{L-1j-1}]$ being the learned weights of the basis function from the secondary DNN. For the sake of simplicity in the representation, we will only use Θ_j explicitly wherever deemed necessary in the expression of $\Phi_{Lj}(X, U, \Theta_j)$. Further, $\epsilon_j(X, U) = W_L^T (\Phi_L(X, U) - \Phi_{Lj}(X, U)) + \epsilon(X, U)$.

Assumption 9. *There exist positive constants $\bar{W}_k$ for $k = 1, \dots, n$ such that the columns $W_{Lk} \in \mathbb{R}^{n_L+1}$ of weight matrix W_L are bounded i.e. $\|W_{Lk}\| \leq \bar{W}_k$. Further, $\|\Phi_{Lj}(X, U)\| \leq \sigma, \forall j$ within the set $\mathbb{X} \times \mathbb{U}$.*

Assumption 10. *The target translational velocity $v_p(t)$ and angular velocity $\psi_p(t)$ along the z -axis, both expressed in the inertial frame, are bounded such that*

$$\sup_{t \geq 0} \|v_p(t)\| = h_1, \quad \sup_{t \geq 0} \|\dot{v}_p(t)\| = h_2, \quad (7.18)$$

$$\sup_{t \geq 0} \|\dot{\psi}_p(t)\| = h_3, \quad \sup_{t \geq 0} \|\ddot{\psi}_p(t)\| = h_4, \quad (7.19)$$

where $h_1, h_2, h_3, h_4 \in \mathbb{R}_+$ are finite constants.

Based on assumption 10 and the fact that matrix R_B^C is constant, we can derive the following upper bounds on the target motion expressed in $\{v\}$

$$\sup_{t \geq 0} \|\bar{v}_p(t)\| = \sup_{t \geq 0} \|R_C^{BT} R_z(\psi)^T v_p(t)\| \leq h_1. \quad (7.20)$$

$$\sup_{t \geq 0} \|\ddot{\psi}_p(t)\| \leq \|\ddot{\psi}_p(t)\| \leq h_4. \quad (7.21)$$

Property 2. *Given the compact set $\mathbb{X} \times \mathbb{U}$ there exist $\bar{\epsilon}' > 0$ such that $\|\epsilon_j(X, U)\| \leq \bar{\epsilon}' \forall j$.*

7.2 Control Design

The control design is done hierarchically as shown in Figure 7.2. We first design the thrust input $f(t)$ to control the quadrotor translational motion based on feature and quadrotor velocity dynamics (7.14) and (7.15) along with a desired rotation matrix $R_c \in$

SO(3). We then briefly discuss the control design process for attitude controller such that the $R(t)$ should track $R_c(t)$.

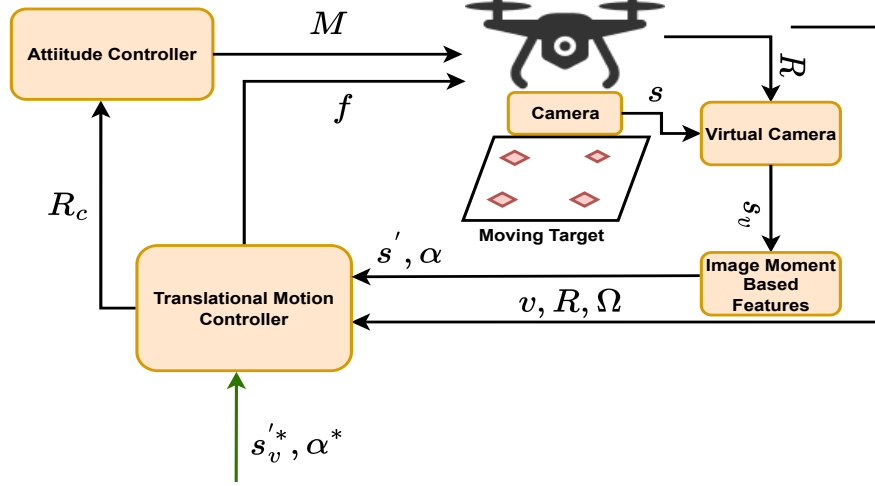


Figure 7.2: Control architecture.

7.2.1 Translational Motion Controller

First, we design the target translational motion estimator following similar developments in Chapter 3.

Target Motion Estimation

Consider filtering the error dynamics (7.12) with the measurable signals $e_s(t)$, $\dot{\psi}(t)$ and $\bar{v}(t)$ and unknown signal $\bar{v}_p(t)$ using the following differential structures

$$\gamma_1 \dot{e}_{fs} + e_{fs} = e_s, \quad e_{fs}(0) = 0, \quad (7.22)$$

$$\gamma_1 \dot{h}_s + h_s = S_k([0, 0, \dot{\psi}]^T) e_s + \bar{v}, \quad h_s(0) = 0, \quad (7.23)$$

$$\gamma_1 \dot{S}_s + S_s = \bar{v}_p, \quad S_s(0) = 0. \quad (7.24)$$

Here $\gamma_1 \in \mathbb{R}_+$ is a design parameter. Following the results of Section 3.2.1 in Chapter 3, the filtered measurable outputs $e_{fs}(t) \in \mathbb{R}^3$ and $h_s(t) \in \mathbb{R}^3$ can be used to get the solution

of (7.24), i.e the filtered output corresponding to unknown signal $\bar{v}_p(t)$ as follows

$$S_s(t) = h_s(t) + \frac{1}{\gamma_1} \left(e_s(t) - e_{fs}(t) - \exp\left(-\frac{1}{\gamma_1}t\right) e_s(0) \right). \quad (7.25)$$

Further, the estimate $\hat{v}_p(t)$ of the target velocity expressed in $\{V\}$ is given as

$$\hat{v}_p(t) = S_s(t). \quad (7.26)$$

Next, we propose a backstepping-based translational motion controller design method.

Translational Dynamics Control

For the sake of clarity, we proceed with the following steps.

Step 1 (Reference Velocity Design): Let's first consider a Lyapunov candidate L_1 for feature tracking error $e_s(t)$ as follows

$$L_1 = \frac{1}{2} e_s^T e_s. \quad (7.27)$$

Differentiating $L_1(t)$ and using a virtual velocity reference $\bar{v}_d$, we get

$$\dot{L}_1 = e_s^T (-\bar{v} + \bar{v}_p \pm \bar{v}_d). \quad (7.28)$$

We design $\bar{v}_d = k_s e_s + \hat{v}_p(t)$ while $k_s > 0$ is a control parameter. Thus,

$$\dot{L}_1 = -k_s e_s^T e_s + e_s^T (-\bar{v} + \bar{v}_d + \bar{v}_p - \hat{v}_p). \quad (7.29)$$

The next step is to design the thrust control input for the velocity dynamics of the quadrotor to follow the designed virtual velocity reference $\bar{v}_d(t)$ expressed in virtual frame $\{V\}$. Before that, let's consider the velocity tracking error for translational dynamics $e_v(t)$ and the target translational velocity estimation error $e_{\bar{v}_p}(t)$

$$e_v = v - v_d \quad (7.30)$$

$$e_{\bar{v}_p} = \bar{v}_p - \hat{v}_p, \quad (7.31)$$

where $v_d(t)$ is the virtual velocity reference $\bar{v}_d$ expressed in the inertial frame $\{I\}$ i.e. $v_d(t) = R_z(\psi(t))R_C^B \bar{v}_d(t)$. Using the relations (7.30) and (7.31) in equation (7.29) yields

$$\begin{aligned}\dot{L}_1 &= -k_s e_s^T e_s + e_s^T R_C^{B^T} R_z(\psi)^T (-v + v_d) + e_s^T e_{\bar{v}_p}, \\ &= -k_s e_s^T e_s - e_s^T R_C^{B^T} R_z(\psi)^T e_v + e_s^T e_{\bar{v}_p}.\end{aligned}\quad (7.32)$$

We will utilize the intermediate Lyapunov candidate $L_1(t)$ to design the thrust control input. Let's derive the dynamics of $me_v(t)$

$$m\dot{e}_v = mge_3 - fRe_3 + \Delta(X, U) - m\dot{v}_d. \quad (7.33)$$

Adding and subtracting $f_d(t)$, a virtual thrust control input in the error dynamics (7.33) yields

$$\begin{aligned}m\dot{e}_v &= mge_3 - fRe_3 + \Delta(x, u) - m\dot{v}_d \pm f_d. \\ &= mge_3 + \Delta f - m\dot{v}_d - f_d + \Delta(X, U),\end{aligned}\quad (7.34)$$

where $\Delta f(t) = f_d(t) - f(t)Re_3$. We would need $\dot{v}_d(t)$ to design the thrust control input. From the definition of $v_d(t)$ and the designed auxiliary virtual velocity $\bar{v}_d(t)$

$$v_d = R_z(\psi(t))R_C^B(k_s e_s + \hat{\bar{v}}_p). \quad (7.35)$$

Step 2 (Time derivative of $v_d(t)$): To obtain an estimate of $\dot{v}_d$, we use the following filter with filter gain $\gamma_2 \in \mathbb{R}_+$

$$\gamma_2 \dot{v}_{fd} + v_{fd} = v_d, \quad v_{fd}(0) = 0. \quad (7.36)$$

Consider another filter to filter the unknown variable $\dot{v}_d$ as

$$\gamma_2 \dot{g}_{v_d} + g_{v_d} = \dot{v}_d, \quad g_{v_d}(0) = 0. \quad (7.37)$$

Integrating (7.37) for $g_{v_d}(t)$ yield

$$g_{v_d}(t) = \frac{1}{\gamma_2} \left(v_d(t) - v_{fd}(t) - \exp\left(-\frac{1}{\gamma_2}t\right)v_d(0) \right). \quad (7.38)$$

Note that for very small values of γ_2 , the term $g_{v_d}(t)$ can be made to converge to $\dot{v}_d(t)$ as shown in Swaroop *et al.* (2000); Roy *et al.* (2018).

Step 3 (Trust Input Design): Following the backstepping analysis in the subsequent section, we design the control input $f(t)$ at any time $t \in [t_{j-1}, t_j]$ as follows

$$f = f_d \cdot Re_3, \quad (7.39)$$

$$f_d = mge_3 - mg_{v_d} + k_v e_v - R_z(\psi)R_C^B e_s + \hat{\Delta}_j(X, U), \quad (7.40)$$

where $k_v > 0$ is the control parameter and $\hat{\Delta}_j(X, U)$ is the inferred uncertainty as

$$\hat{\Delta}_j(X, U) = \hat{W}_L^T \Phi_{Lj}(X, U). \quad (7.41)$$

Here $\hat{W}_L(t)$ is the estimate of ideal weight matrix W_L . Note that $\hat{W}_L(t)$ is updated in real-time while $\Phi_{Lj}(X, U)$ is updated at the t_{j-1} upon the completion of training on the secondary DNN. At $t_0 = 0$, the weights of both DNNs are the same and randomly initialized, respecting the bounded weight assumption on the outer layer weights. We next design the composite adaption-based update law for $\hat{W}_L(t)$.

Outer Layer Weight Update Law

To design a composite update law for outer layer weights $\hat{W}_L$ of the DNN, we consider the filter structures similar to Step 2, for obtaining the tracking error derivative $\dot{e}_v(t)$. For the measurable signals $e_v(t)$, $ge_3 - \frac{f}{m}Re_3 - g_{v_d}$ and $\Phi_{Lj}(X, U)$, let's consider the following filter structures with filter design parameter $\gamma_3 > 0$

$$\gamma_3 \dot{e}_{v_f} + e_{v_f} = e_v, \quad e_{v_f}(t_{j-1}) = 0, \quad (7.42)$$

$$\gamma_3 \dot{u}_f + u_f = ge_3 - \frac{f}{m}Re_3 - g_{v_d}, \quad u_f(t_{j-1}) = 0, \quad (7.43)$$

$$\gamma_3 \dot{\Phi}_f + \Phi_f = \Phi_{Lj}(X, U), \quad \Phi_f(t_{j-1}) = 0. \quad (7.44)$$

Further consider the same filters for unknown signals $\Delta(X, U)$ and $\dot{e}_v(t)$ as

$$\gamma_3 \dot{g}_f + g_f = \dot{e}_v, \quad g_f(t_{j-1}) = 0, \quad (7.45)$$

$$\gamma_3 \dot{h}_f + h_f = W_L^T \Phi_{Lj}(X, U) + \varepsilon_j(x, u), \quad h_f(t_{j-1}) = 0. \quad (7.46)$$

We solve equation (7.45) using integration by parts to get

$$g_f(t) = \frac{1}{\gamma_3} \left(e_v(t) - e_{v_f}(t) - \exp\left(-\frac{1}{\gamma_3}t\right) e_v(t_{j-1}) \right). \quad (7.47)$$

Further substituting the (7.33) in (7.45) would yield

$$h_f = W_L^T \Phi_f + \varepsilon_f = m(g_f - u_f), \quad (7.48)$$

where ε_f is the filtered output of $\varepsilon_j(X, U)$ via

$$\gamma_3 \dot{\varepsilon}_f + \varepsilon_f = \varepsilon_j(X, U), \quad \varepsilon_f(t_{j-1}) = 0. \quad (7.49)$$

The relation (7.48) provides a measurable signal corresponding to filtered uncertainty h_f . We now design a prediction error for the composite adaption in the outer layer weight update law. Let's define the following uncertainty predictor model $\hat{h}_f = \hat{W}_L^T \Phi_f$ and the corresponding prediction error

$$\zeta_f = h_f - \hat{h}_f = \tilde{W}^T \Phi_f + \varepsilon_f, \quad (7.50)$$

where $\tilde{W} = W_L - \hat{W}_L$. Equipped with an expression involving a measure of weight matrix error $\tilde{W}$, the outer layer weight matrix update law for any time $t \in [t_{j-1}, t_j)$, $\forall j \in \mathbb{N}$ is designed as follows

$$\dot{\hat{W}}_L = \text{proj}(\hat{W}_L, \Gamma(\Phi_{Lj}(X, U)e_v^T + k_w \Phi_f \zeta_f^T)), \quad (7.51)$$

where $\text{proj}(\cdot)$ is the projection operator Landau *et al.* (2011) to ensure the column bounds similar to Assumption 9 and $k_w, \Gamma > 0$ are the design parameters.

Basis Function Training

The main objective of basis function training is to make $\Phi_{Lj}(x, u) \rightarrow \Phi_L(x, u)$ as j progresses, making the $\|\varepsilon_j\|$ smaller and thus yielding a smaller ultimate bound for the tracking error. Analogous to what is established in Joshi and Chowdhary (2019), given an input vector $[x^T, u^T]^T$, $\hat{\Delta}_j(X, U)$ serves as a decent label for training the feature

vector or the inner layers of the DNN.

To train the inner layers of the DNN, we create a buffer $Z = \{X_{pk}, Y_{pk}\}_{k=1}^{p_{max}}$ that can store p_{max} training samples of the input vector $X_p = [X^T, U^T]^T$ and label $Y_p = \hat{\Delta}_j(X, U)$. Once the buffer is full, new data samples are added following the standard techniques given in the literature Joshi and Chowdhary (2019); Hewing *et al.* (2019); Scholkopf *et al.* (1999). Once the buffer is full and a user-defined condition of when to start the training is met, we take a set Z_{train} with $p_{train}, p_{train} \leq p_{max}$ number of samples out of the whole buffer for training the DNN and use the following empirical loss function to train the DNN

$$\mathcal{L}(Z_{train}, \hat{W}_{0j:L-1j}) = \frac{1}{p_{train}} \sum_{i=1}^{p_{train}} \|Y_i - \hat{Y}_i\|^2, \quad (7.52)$$

where $\hat{Y}_i = \hat{W}_L^T(t_{j-1})\Phi_{Lj}(X_i, U_i)$. Note that the outer layer weight matrix $\hat{W}_L^T(t_{j-1})$ remains fixed for the whole training period and copied from the main DNN at the start of the training. We refer the reader to (Joshi and Chowdhary, 2019, Section IV) for buffer creation and training details.

This completes the controller design for the translational dynamics of the quadrotor. Next, we briefly discuss the design of the attitude controller.

7.2.2 Attitude Tracking Controller

Let's design the desired orientation matrix $R_c(t)$. Considering the requirement of making $e_\alpha(t) \rightarrow 0$, one can directly design the desired heading $\psi_d(t)$ which would then be utilized to construct the desired rotation matrix. Using the error dynamics (7.13), the heading of the target $\bar{\psi}_p(t)$ expressed in $\{V\}$ can be reconstructed as

$$\bar{\psi}_p(t) = e_\alpha(t) + \bar{\psi}(t) + \bar{\psi}_{Offset}, \quad (7.53)$$

where $\bar{\psi}_{Offset}$ is the desired relative heading of the target with respect to the quadrotor. Given the relation (7.53), we can easily derive that the desired heading trajectory is derived as $\bar{\psi}_d(t) = \bar{\psi}_p(t) - \bar{\psi}_{Offset}$. One can further obtain the target heading $\psi_p(t)$ using the explicit relation $R_z(\bar{\psi}_p) = R_z(\psi_p)R_C^B$ similar approach can be applied to compute $\psi_d(t)$. Thus, ensuring $\psi(t) \rightarrow \psi_d(t)$ guarantees that $e_\alpha(t) \rightarrow 0$.

Given the desired heading direction $\hat{x}_{B_d}(t) = [\cos(\psi_d(t)), \sin(\psi_d(t)), 0]^T$, the desired attitude $R_c(t)$ is constructed exploiting the virtual control input $f_d(t)$ given in (7.40) as

$$R_c = [\hat{x}_{B_c}, \hat{y}_{B_c}, \hat{z}_{B_c}], \quad (7.54)$$

$$\hat{z}_{B_c} = \frac{f_d}{\|f_d\|} \quad \hat{y}_{B_c} = -\frac{\hat{x}_{B_d} \times \hat{z}_{B_c}}{\|\hat{x}_{B_d} \times \hat{z}_{B_c}\|} \quad \hat{x}_{B_c} = \hat{y}_{B_c} \times \hat{z}_{B_c}. \quad (7.55)$$

To design the control input M , we leverage the developments from Chapter 5, specifically the process that involves passing $R_c(t)$ through auxiliary filter dynamics to generate a filtered desired attitude reference, which the proposed attitude tracking controller is designed to follow. The auxiliary filters the signal $R_c(t) \in SO(3)$ into a filtered signal $R_d(t) \in SO(3)$ strictly belonging to the special orthogonal group and an angular velocity $\Omega_d(t)$ associated to it as

$$\dot{R}_d = R_d \hat{\Omega}_d, \quad (7.56)$$

$$\dot{\Omega}_d = -\frac{1}{\gamma_5} e_{\Omega_{dc}} + e_R(R_d, R_c) + \frac{1}{\gamma_4} E(R_d, R_c) \Omega_d, \quad (7.57)$$

where $\gamma_4, \gamma_5 \in \mathbb{R}_+$ are the constant design parameters. Error $e_R(R_d, R_c)$ is the attitude error between R_c and R_d and $e_{\Omega_{dc}} = \Omega_d + \frac{1}{\gamma_4} e_R(R_d, R_c)$.

Equipped with an auxiliary attitude reference $R_d(t)$, $\Omega_d(t)$, we design the attitude control input M Lee (2012) to track the auxiliary reference signals $R_d(t), \Omega_d(t), \dot{\Omega}_d(t)$ as follows

$$M = -k_R e_R(R, R_d) - k_\Omega e_\Omega(R, \Omega, R_d, \Omega_d) + \Omega \times J \Omega - J(\hat{\Omega} R^T R_d \Omega_d - R^T R_d \dot{\Omega}_d), \quad (7.58)$$

where $\{k_R, k_\Omega \in \mathbb{R}_+\}$ are constant control design parameters. For detailed expressions of the attitude errors $e_R(R, R_d)$, $e_\Omega(R, \Omega, R_d, \Omega_d)$, $e_R(R_d, R_c)$ and $E(R_d, R_c)$, please refer to Chapter 5.

Remark 11. Another common approach in dynamic controller design involves using high-gain inner-loop controllers to regulate the attitude dynamics Li et al. (2021a); O'Connell et al. (2022); Shi et al. (2019); Serra et al. (2016) to track the desired orientation $R_c(t)$. The high-gain inner-loop controller ensures the attitude dynamics are

fast enough to assume $R \approx R_c$. This simplifies the translational error dynamics (7.34), effectively driving $\Delta f \rightarrow 0$. The approach is not only practical (being directly compatible with native autopilot systems like PX4 and Ardupilot) but is also theoretically supported by small-gain arguments, particularly in scenarios where aggressive maneuvers are unnecessary, such as during standard landing operations. Thus, one control design method is to directly send $f(t)$ and desired attitude commands using $R_c(t)$. To maintain clarity and avoid the semi-global arguments due to translational dynamics error based upper bounds on $\dot{\Omega}_c$ (as detailed in Lemma 9 of Chapter 5), we assume that $R \approx R_c$ is achieved sufficiently fast i.e. $\Delta f \approx 0$.

7.2.3 Stability Analysis

Given Remark 11, we now prove the convergence of translational dynamics errors to an ultimately bound.

Theorem 10. *Consider the system dynamics (7.14), (7.15) with control and outer weight update laws (7.39) and (7.51). Assuming $\Delta f = 0$ is achieved sufficiently fast with (7.58), the errors $e(t) = [e_s, e_v, e_{\bar{v}_p}]$ is uniformly ultimately bounded for $k_s > \frac{1}{2}, \gamma_1 < 1$.*

Proof. Let's consider the following Lyapunov candidate

$$L = L_1 + \frac{1}{2}e_v^T m e_v + \frac{1}{2}e_{\bar{v}_p}^T e_{\bar{v}_p} + \frac{1}{2}\text{tr}(\tilde{W}^T \Gamma^{-1} \tilde{W}). \quad (7.59)$$

Taking time-derivative of L using (7.40), (7.32), and (7.51) yields

$$\begin{aligned} \dot{L} = & -k_s e_s^T e_s - e_s^T R_C^B R_z^T(\psi) e_v + e_s^T e_{\bar{v}_p} - \frac{1}{\gamma_1} e_{\bar{v}_p}^T e_{\bar{v}_p} + e_{\bar{v}_p}^T \dot{\bar{v}}_p \\ & - k_v e_v^T e_v + e_v^T (R_z(\psi) R_C^B e_s + \tilde{W}^T \Phi_{Lj}(X, U) + \epsilon_j(X, U)) \end{aligned} \quad (7.60)$$

$$- \text{tr}(\tilde{W}^T \text{proj}(\hat{W}_L, \Gamma(\Phi_{Lj}(X, U) e^T + k_w \Phi_f \zeta_f^T))), \quad (7.61)$$

Further, using (7.50) and projection theory Landau *et al.* (2011) yields

$$\dot{L} = -k_s e_s^T e_s + e_s^T e_{\bar{v}_p} - k_v e_v^T e_v + e_v^T \epsilon_j(X, U) - k_w \text{tr}(\zeta_f \zeta_f^T - \zeta_f \epsilon_f^T) - \frac{1}{\gamma_1} e_{\bar{v}_p}^T e_{\bar{v}_p} + e_{\bar{v}_p}^T \dot{\bar{v}}_p. \quad (7.62)$$

Further using the fact that for any vectors a, b , $tr(aa^T) = \|a\|^2$ and $tr(ab^T) = a^T b$, we get

$$\dot{L} = -k_s e_s^T e_s + e_s^T e_{\bar{v}_p} - k_v e_v^T e_v + e_v^T \varepsilon_j(X, U) - k_w \|\zeta_f\|^2 + k_w \zeta_f^T \varepsilon_f - \frac{1}{\gamma_1} e_{\bar{v}_p}^T e_{\bar{v}_p} + e_{\bar{v}_p}^T \dot{v}_p. \quad (7.63)$$

Before proceeding further, we calculate the upper bounds of $\dot{v}_p(t)$ using the fact that

$$\dot{v}_p(t) = -S_k([0, 0, \dot{\psi}]^T) R_z(\bar{\psi})^T v_p(t) + R_z(\bar{\psi})^T \dot{v}_p(t). \quad (7.64)$$

Thus, the upper bound on $\dot{v}_p(t)$ can be given as

$$\|\dot{v}_p(t)\| \leq h_5 h_1 + h_2, \quad (7.65)$$

where $h_5 \in \mathbb{R}_+$ is the upper bound of the heading rate of the virtual frame expressed in the virtual frame.

With Property 2, taking bounds using Cauchy inequality in equation (7.63) yields

$$\begin{aligned} \dot{L} \leq & -k_s \|e_s\|^2 + \frac{1}{2} k_s \|e_s\|^2 + \frac{1}{2} \|e_{\bar{v}_p}\|^2 - \frac{1}{2} k_v \|e_v\|^2 - \frac{k_w}{2} \|\zeta_f\|^2 + \frac{1}{2k_v} \bar{\varepsilon}^2 \\ & + \frac{k_w}{2} \|\varepsilon_f\|^2 - \frac{1}{\gamma_1} \|e_{\bar{v}_p}\|^2 + \frac{1}{2\gamma_1} \|e_{\bar{v}_p}\|^2 + \gamma_1 h_1^2 h_5^2 + \gamma_1 h_2^2. \end{aligned} \quad (7.66)$$

Further, from filter structure (7.49), for a very small γ_3 and $\varepsilon_f(0) = 0$, we have $\|\varepsilon_f\| \leq \bar{\varepsilon}'$. Thus,

$$\begin{aligned} \dot{L} \leq & -(k_s - \frac{1}{2}) \|e_s\|^2 - \frac{1}{2} k_v \|e_v\|^2 - \frac{k_w}{2} \|\zeta_f\|^2 - (\frac{1}{2\gamma_1} - \frac{1}{2}) \|e_{\bar{v}_p}\|^2 \\ & + \gamma_1 h_1^2 h_5^2 + \gamma_1 h_2^2 + (\frac{1}{2k_v} + \frac{k_w}{2}) \bar{\varepsilon}^2. \end{aligned} \quad (7.67)$$

Using the fact that $\frac{1}{2} \|e\|^2 + \frac{1}{2} e_v^T m e_v + \frac{1}{2} e_{\bar{v}_p}^T e_{\bar{v}_p} \geq L - \lambda_{\max}(\Gamma^{-1}) \|\tilde{W}\|^2$ in (7.67) and further solving yields

$$\dot{L} \leq -\beta L + D, \quad (7.68)$$

where $\beta = \min\left((2k_s - 1), k_v, (\frac{1}{\gamma_1} - 1)\right)$ and $D = \beta \lambda_{\max}(\Gamma^{-1}) \|\tilde{W}\|^2 + \gamma_1 h_1^2 h_5^2 + \gamma_1 h_2^2 +$

$(\frac{1}{2k_v} + \frac{k_w}{2})\bar{\epsilon}^2$. Using the fact that $\|\tilde{W}\|$ is bounded due to Assumption 9 and differential weight update law with projection (7.51), relation (7.68) proves that $e(t)$ is uniformly ultimately bounded Khalil (2002). $\square$

7.3 Results

The results section presents the Python simulation outcomes to demonstrate the effectiveness of the proposed IBVS-based control framework for quadrotor tracking and landing on a moving planar target. We present the simulation results under two distinct types of uncertainty scenarios to evaluate the robustness and adaptability of the proposed control framework. A simulation environment is created in Python, featuring a quadrotor equipped with a downward-facing camera aligned with $R_C^B = I$, and a planar target marked with four distinct points. Further, we use MATLAB to create the visualization and to plot the interactive results from Python simulations. The simulation assumes a North-East-Down (NED) inertial frame to consistently represent the positions and orientations of the quadrotor and the target. This setup provides a realistic scenario to evaluate the proposed control framework for tracking and landing tasks.

The camera model incorporates realistic parameters, including a focal length of 15 mm, principal point $u_0 = v_0 = 512$ pixels, pixel dimensions $\rho_h = \rho_w = 10^{-5}$ m/pixel, and a resolution of 1024×1024 pixels. The planar target is represented by four markers located at $P_1 = [-0.05, -0.05, 0.0]^T m$, $P_2 = [0.05, -0.05, 0.0]^T m$, $P_3 = [0.05, 0.05, 0.0]^T m$, and $P_4 = [-0.05, 0.05, 0.0]^T m$ all expressed in the inertial frame. The quadrotor starts at an initial position of $[0.0, 0.2, -3.0]^T m$ with an angular orientation (RPY) of $[0, 0, 0.6]^T rad$, also expressed in the inertial frame. The desired visual features are chosen to ensure that the quadrotor tracks the moving target while maintaining a final altitude of 0.5m above it, aligning with the length of the landing gear. The simulated scenario corresponds to a realistic landing operation, with results demonstrating the quadrotor's ability to robustly track and land on the target. The IBVS dynamic controller parameters are selected as follows: $k_s = 1$, $k_v = 5.6$, $k_R = 8.81$, $k_\Omega = 2.54$, $\Gamma = 0.7I$, $k_w = 0.5$, $\gamma_1 = 10^{-2}$, $\gamma_2 = 10^{-2}$, $\gamma_3 = 10^{-2}$, $\gamma_4 = 10^{-1}$, $\gamma_5 = 10^{-1}$. The target is considered to move with linear and angular velocities varying over time $v_p(t) = [0.5\cos(t), 0.6\sin(t), 0]^T m/sec$ and $\psi_p(t) = \frac{\pi}{10}\sin(t) rad/sec$, respectively.

The DNN used in the simulation comprises four hidden layers with neuron configurations of $n_1 = 30$ for the first layer and $n_{2,3,4} = 50$ for the subsequent layers. Hyperbolic tangent activation functions are employed at each layer to enable smooth nonlinear approximations. The buffer and batch sizes are set to $p_{max} = p_{train} = 100$, with secondary DNN training initiated once 50 new data samples are added to the buffer after reaching its capacity. Data storage in the buffer leverages the singular value maximization approach as outlined in Chowdhary and Johnson (2011). The training process uses a learning rate 0.01 with 150 epochs for the secondary DNN. All simulations use the DOP853 integration method with a sampling time of 0.01 seconds, ensuring accurate numerical results and stability in dynamic system simulations. To synthesize the control input from the non-affine control law (7.39), we use the control input f from the last time step as an input to (7.40) on the right-hand side.

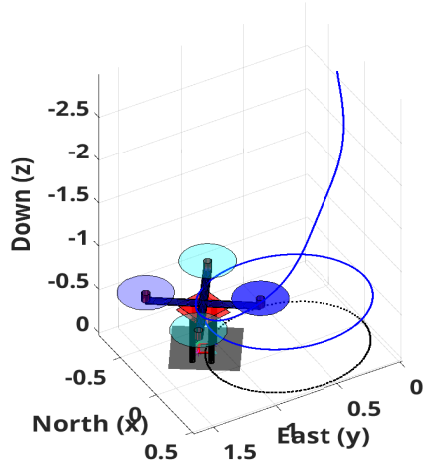
7.3.1 Ground Effect Model:

In the first simulation, we incorporate a ground effect model as described in Pedro Sanchez-Cuevas and Ollero (2017). For a quadrotor with rotors placed a distance d apart, the ground effect model is given by the following equation:

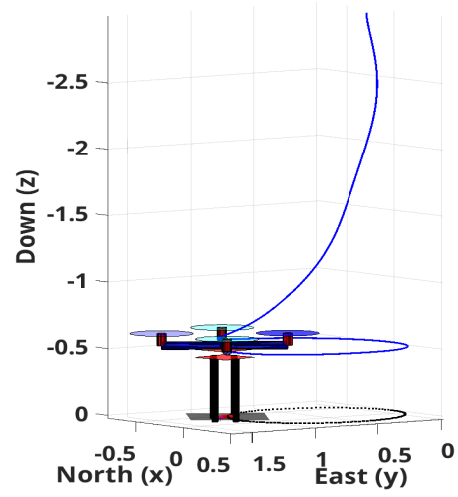
$$\frac{f_g}{f_\infty} = \frac{1}{1 - (r/4z)^2 - \frac{r^2 z}{\sqrt{(d^2 + 4z^2)^3}} - \frac{r^2 z}{2\sqrt{(2d^2 + 4z^2)^3}}}, \quad (7.69)$$

where r is the radius of the rotor, and z is the height of the quadrotor from the ground. f_g is the thrust under the influence of ground effect where f_∞ is the thrust otherwise. This model introduces a correction factor to the thrust as a function of the distance between the quadrotor and the ground, the rotor radius, and the distance between the rotors, thereby modeling the increased aerodynamic forces that act as the quadrotor approaches the ground. We consider $r = 0.20m$ and $d = 0.45m$ for the simulation.

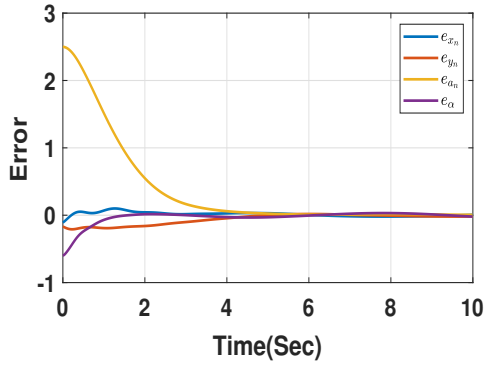
Figure 7.3 showcases the simulation results with the proposed controller. The trajectory plots in the top (Figure 7.3a) and side (Figure 7.3c) views demonstrate the quadrotor's ability to accurately track the moving planar target and achieve a smooth descent, culminating in a successful landing. The feature tracking errors (e_s and e_α), shown in Figure 7.3b, converge to zero, indicating precise alignment and positioning relative to the target. Furthermore, the velocity estimation errors in Figure 7.3d confirm the accu-



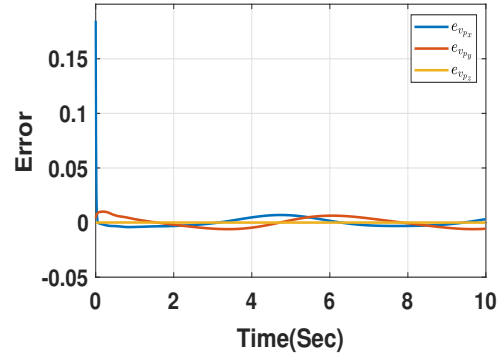
(a) Trajectory plot (Top view)



(c) Trajectory plot (Side view)

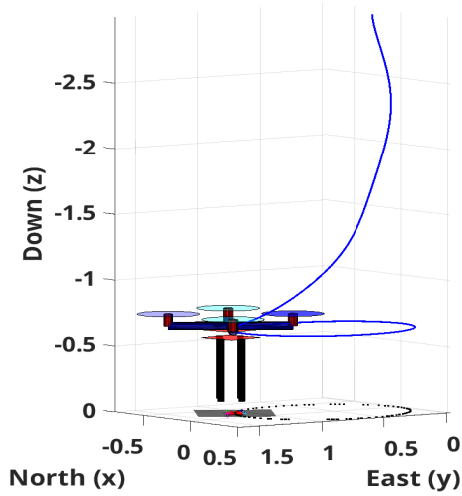


(b) Feature tracking error (e_s, e_α)

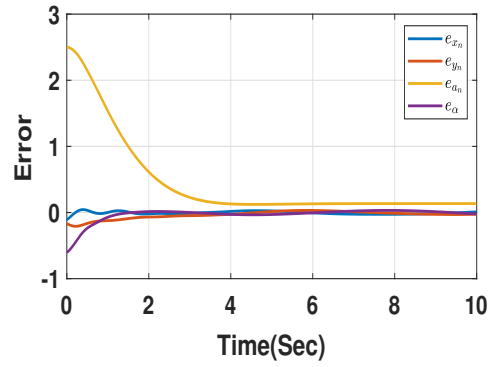


(d) Target velocity estimation error

Figure 7.3: Simulation results demonstrating trajectory tracking (top and side views), feature tracking error, and velocity estimation error with the proposed controller.



(a) Trajectory plot (Side view)



(b) Feature tracking error (e_s, e_α)

Figure 7.4: Simulation results without DNN-based uncertainty mitigation.

rate estimation of the target's motion, which is critical for achieving a smooth landing.

In contrast, Figure 7.4 illustrates the simulation results without DNN-based uncertainty mitigation (without $\hat{\Delta}_j(X, U)$). The trajectory in Figure 7.4a reveals the quadrotor's inability to land on the target due to the ground effect, as it remains suspended above the target. This limitation is further highlighted in Figure 7.4b, where, despite some reduction in e_s , $e_{a_{x_n}}$ fails to converge to zero, signifying incomplete alignment and descent. The observed behavior can be attributed to the inability of the controller to compensate for the ground effect-induced disturbances.

These results underscore the effectiveness of the proposed DNN-based controller, which successfully compensates for uncertainties, including the ground effect, to achieve precise target tracking and landing.

7.3.2 Artificial State and Input-Dependent Uncertainties:

The second simulation considers a more challenging scenario involving complex uncertainties. These include a combination of state and control input-dependent variations and actuator ineffectiveness, simulating a broader spectrum of real-world disturbances. We consider the following uncertainty

$$\Delta = 0.05 \frac{f}{m} Re_3 + \begin{bmatrix} x_1 \tanh(x_2) \\ \sin(x_1) \text{sech}^2(f) \\ \tanh(x_1) \sin(x_3) \end{bmatrix}. \quad (7.70)$$

The results in Figures 7.5 and 7.6 demonstrate the quadrotor's performance under complex artificial uncertainties with and without the proposed DNN-based uncertainty mitigation. Figure 7.5 highlights the effectiveness of the proposed controller in accurately tracking the target trajectory (as evident in the top and side views), mitigating the influence of uncertainties, and achieving precise landing on the moving target. The feature tracking error in Figure 7.5b converges effectively, confirming the control objective is achieved. Additionally, the accurate estimation of the target velocity in Figure 7.5d underscores the robustness of the proposed estimator incorporated IBVS controller.

In contrast, the baseline controller, shown in Figure 7.6, fails to address the impact

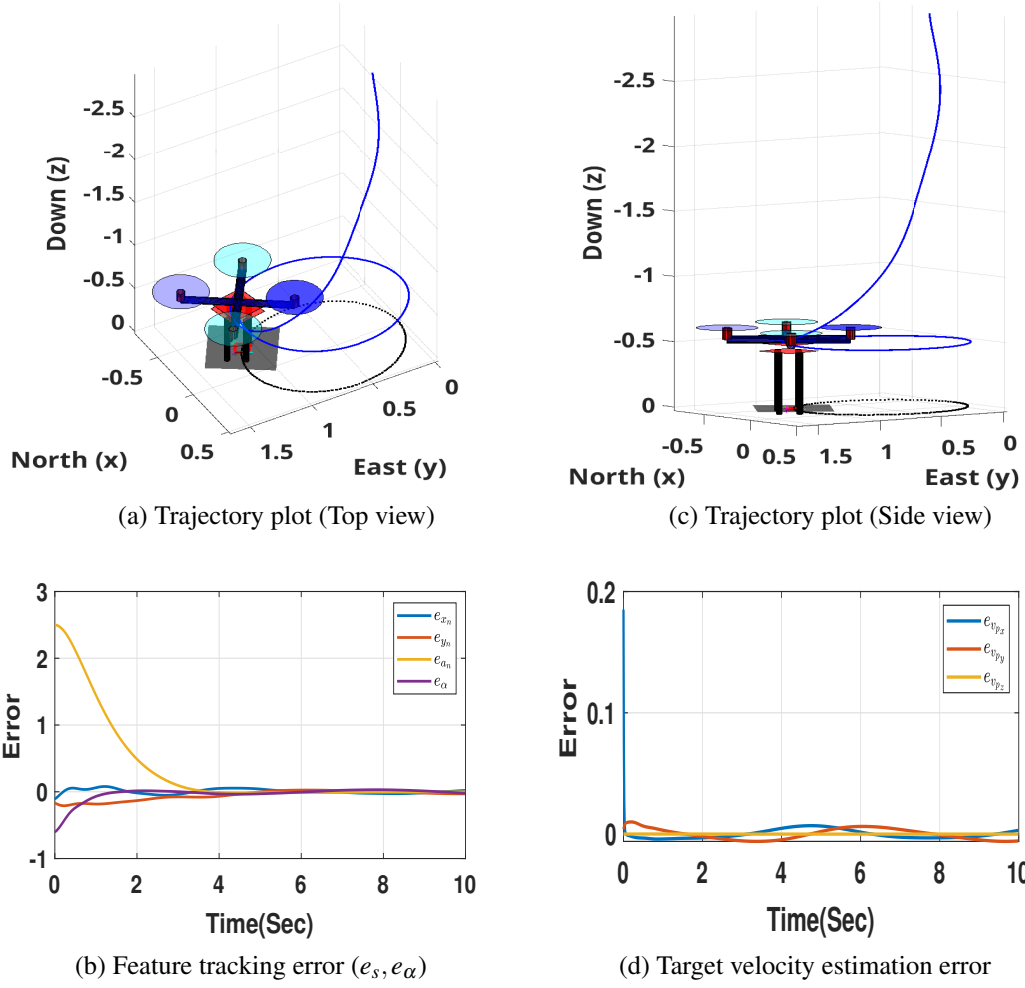


Figure 7.5: Simulation results for artificial uncertainty with the proposed controller.

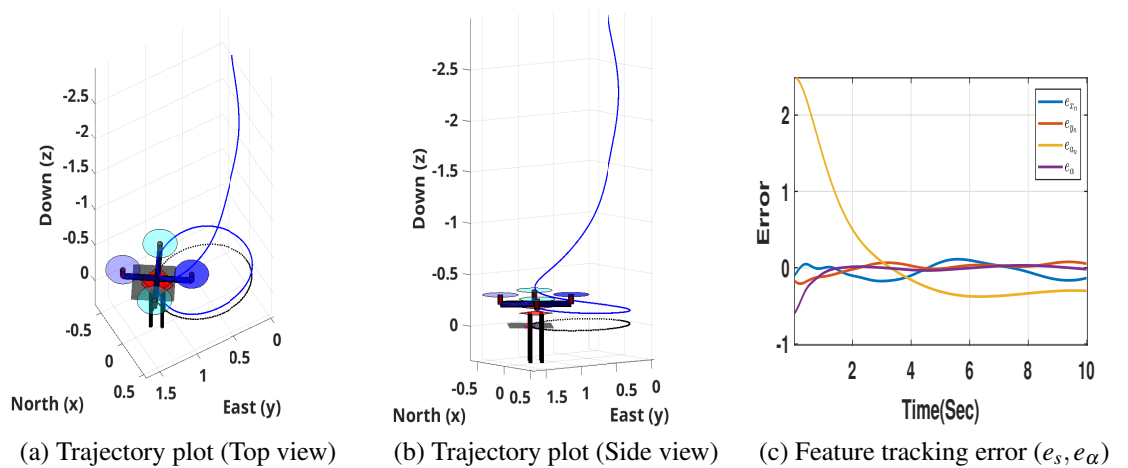


Figure 7.6: Simulation results without DNN-based uncertainty mitigation, showing trajectory plots (top and side views) and feature tracking error.

of the artificial uncertainties. The trajectory plots in Figures 7.6a and 7.6b reveal significant deviations, leading to an incorrect landing location rather than the target. Further-

more, the feature tracking error in Figure 7.6c exhibits a lack of convergence, reflecting the controller's inability to compensate for the complex state and input-dependent variations. These results confirm that the proposed DNN-based approach significantly enhances system reliability, ensuring precise tracking and landing in challenging conditions.

7.4 Conclusion

This chapter presented an IBVS-based control framework for a quadrotor tasked with tracking and landing on a moving planar target with arbitrary and unknown motion. The framework effectively addressed uncertainties by incorporating target motion estimation and a DNN-based neuro-adaptive approach for controlling translational dynamics while maintaining stability and ensuring precise landing. The desired orientation for the attitude controller was derived from the reconstructed heading information of the target, and geometric control principles were employed to ensure robust and singularity-free attitude tracking. Convergence analysis confirmed that both tracking and estimation errors remain uniformly bounded, demonstrating the stability of the proposed framework. Extensive simulations validated the system's capability to track dynamic targets and perform accurate landings under realistic conditions.

Chapter 8

Conclusion and Opportunities for Further Research

This thesis presents significant advancements in the domain of vision-based control and robust dynamic maneuvering of quadrotors, focusing on challenges associated with tracking and landing on moving targets under real-world uncertainties. It begins with developing a novel Image-Based Visual Servoing (IBVS) framework capable of tracking targets with arbitrary, unknown, and time-varying 3D motion. A dynamic surface-like filter-based velocity estimator is introduced, which enhances the robustness of the visual servoing system by accurately estimating target velocity, even in the presence of disturbances and inaccuracies. Building on this, the thesis extends dynamic target tracking IBVS control to underactuated quadrotor systems, overcoming the limitations of quadrotors in tracking full 6-DOF velocity references and managing unknown target yaw motion.

The thesis further addresses the critical issue of tracking moving targets using IBVS controllers with unknown desired depth, presenting two innovative control approaches. These approaches not only expand the feasible range of target motions but also ensure convergence of parameter estimation errors, further enhancing the practical applicability of IBVS in dynamic target tracking scenarios. In addressing trajectory tracking for quadrotors, this work introduces a geometric control framework to track desired trajectories with limited knowledge, focusing on position and heading information. The proposed framework overcomes the limitations of Euler angle-based methods, providing a hierarchical structure for attitude and translational motion control that improves trajectory tracking performance.

Another key feature of this thesis is the development of a real-time, DNN-based adaptive control framework to manage state and control input uncertainties. By leveraging deep neural networks, the control system adapts in real-time to unstructured uncertainties, ensuring stability and significantly improving both transient and steady-state tracking performance. Finally, the thesis proposes an IBVS-based control framework for landing on moving targets, combining target motion estimation with a DNN-based

neuro-adaptive framework. The proposed controllers effectively combine IBVS-derived references with translational control mechanisms, enabling the quadrotor to dynamically track and land on targets with arbitrary motion patterns. The proposed solution ensures precise, stable, and accurate landing under dynamic conditions, demonstrating the robustness and applicability of the approach through extensive simulations.

In summary, this thesis presents a comprehensive set of contributions that advance the field of visual servoing and adaptive control for quadrotors. The proposed frameworks significantly improve quadrotors' practical applicability and reliability in complex, dynamic environments by addressing key challenges such as dynamic target tracking, depth uncertainty, and partially known trajectory tracking and handling uncertainties.

8.1 Future Research Directions

While this thesis has significantly contributed to achieving robust quadrotor landing on moving targets using visual feedback, several avenues remain open for future exploration.

8.1.1 Incorporating Field of View Constraints

The current methods assume an ideal scenario where the target remains within the camera's view throughout the operation. Quadrotors, being underactuated systems, often face situations where visibility constraints are violated. For example, in an Image-Based Visual Servoing (IBVS) task, the quadrotor may need to tilt to move laterally and reduce image plane errors. This tilting motion can unintentionally increase image errors, especially if the target feature is near the edge of the image plane, leading to visibility loss. Virtual camera approaches, which compensate for the camera's rotation using roll and pitch estimates from an IMU, may introduce issues. For instance, visibility constraints defined on the virtual image plane may still appear valid even if the target has already left the actual camera FOV, rendering the constraints ineffective in practice. This highlights the need for more robust formulations to address the mismatch between the virtual and actual image planes while ensuring the target remains within

the camera's field of view.

8.1.2 Marker Free IBVS

Another promising future direction for this thesis is the development and implementation of IBVS for marker-free targets. The methods presented in this thesis rely on artificial markers, such as AprilTags, for tracking. However, markers can be impractical due to the need for precise placement, the potential for misalignment or occlusion, and their limited applicability in environments where they cannot be easily deployed. Therefore, it remains an open challenge to develop an IBVS controller that operates without markers, particularly for landing tasks, and to explore its performance in zero-shot settings where no prior knowledge or predefined markers are available.

8.1.3 Efficient Differentiators for Noisy Signals

A promising future direction is the development of efficient differentiators for noisy signals. Much of the work in this thesis relies on real-time time derivatives of signals, which are often susceptible to noise. However, a trade-off exists between the accuracy of a real-time differentiator and its ability to effectively eliminate derivatives of high-frequency noise in the differentiated signal. Traditional differentiators may amplify the noise, leading to unreliable or unstable control performance. Therefore, it is crucial to develop robust differentiators to accurately compute time derivatives and smoothen the input signals to mitigate noise. Approaches such as low-pass filtering, Kalman filtering, or adaptive filtering could be explored to achieve this balance and enhance the overall performance of the IBVS controller, particularly in real-world applications where sensor noise is inevitable.

8.1.4 Hybrid Visual Servoing for High and Low-Speed Target Tracking

Another promising future direction is developing a generalized solution for tracking both high and low-speed targets with visual feedback. Traditional cameras have limited frame rates (fps), which may not be sufficient for effectively tracking high-speed

targets. Event cameras, which capture changes in the scene at the pixel level, offer a solution for high-speed tracking by providing high temporal resolution. However, event cameras struggle with low-speed targets, as their output becomes sparse and less informative under slow-motion conditions. Therefore, a hybrid visual servoing approach that combines traditional and event cameras' strengths is needed. This technique would enable robust tracking across a wide range of target speeds, leveraging the high temporal resolution of event cameras for fast-moving targets and the spatial accuracy of traditional cameras for slower targets. Such a hybrid approach could significantly improve the performance of visual servoing systems in dynamic environments, where the target's speed may vary unpredictably.

8.1.5 Real-World Testing of the Proposed IBVS-based Landing Controller

Another important future direction involves extensive experimentation of the proposed IBVS-based landing controller. While the controller has shown promise in simulations, further testing in real-world environments is necessary to assess its practical limitations and performance under various conditions. This includes evaluating the computational requirements of the control and estimation algorithms and determining the efficiency of their discretization for real-time implementation. Such experimentation can provide valuable insights into potential improvements and help address any unforeseen challenges in scaling the system for practical applications.

Bibliography

1. **Abdessameud, A.** and **F. Janabi-Sharifi**, Dynamic image-based tracking control for vtol uavs. In *52nd IEEE Conference on Decision and Control*. IEEE, 2013.
2. **Abdessameud, A.** and **F. Janabi-Sharifi** (2015). Image-based tracking control of vtol unmanned aerial vehicles. *Automatica*, **53**, 111–119. ISSN 0005-1098. URL <https://www.sciencedirect.com/science/article/pii/S0005109814006116>.
3. **Báa, T., P. tépán, V. Spurný, D. Heřt, R. Pěnika, M. Saska, J. R. Thomas, G. Loianno, and V. R. Kumar** (2019). Autonomous landing on a moving vehicle with an unmanned aerial vehicle. *Journal of Field Robotics*, **36**, 874 – 891. URL <https://api.semanticscholar.org/CorpusID:91179656>.
4. **Bell, Z. I., P. Deptula, H. Chen, E. A. Doucette, and W. E. Dixon**, Velocity and path reconstruction of a moving object using a moving camera. In *American Control Conference (ACC)*. 2018.
5. **Bensalah, F.** and **F. Chaumette**, Compensation of abrupt motion changes in target tracking by visual servoing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 1. 1995.
6. **Bernard, D. D. C., F. Riccardi, M. Giurato, and M. Lovera** (2017). A dynamic analysis of ground effect for a quadrotor platform. *IFAC-PapersOnLine*, **50**(1), 10311 – 10316. ISSN 2405-8963. URL <http://www.sciencedirect.com/science/article/pii/S2405896317320803>. 20th IFAC World Congress.
7. **Bhargavapuri, M., A. K. Shastri, H. Sinha, S. R. Sahoo, and M. Kothari** (2019). Vision-based autonomous tracking and landing of a fully-actuated rotorcraft. *Control Engineering Practice*, **89**, 113 – 129. ISSN 0967-0661. URL <http://www.sciencedirect.com/science/article/pii/S0967066118306415>.
8. **Boss, C. J., V. Srivastava, and H. K. Khalil**, Robust tracking of an unknown trajectory with a multi-rotor uav: A high-gain observer approach. In *American Control Conference (ACC)*. IEEE, 2020.
9. **Bourquardez, O., R. Mahony, N. Guenard, F. Chaumette, T. Hamel, and L. Eck** (2009). Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle. *IEEE Transactions on Robotics*, **25**(3), 743–749.
10. **Butcher, J. C.**, *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.
11. **Chaumette, F.** (2004). Image moments: a general and useful set of features for visual servoing. *IEEE Transactions on Robotics*, **20**(4), 713–723.

12. **Chaumette, F., P. Rives, and B. Espiau**, Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing. In *IEEE International Conference on Robotics and Automation(ICRA)*. 1991.
13. **Chaumette, F. and A. Santos** (1993). Tracking a moving object by visual servoing. *IFAC Proceedings Volumes*, **26**(2, Part 3), 643–648. ISSN 1474-6670. URL <https://www.sciencedirect.com/science/article/pii/S1474667017488069>.
14. **Chen, C.-W., H.-A. Hung, P.-H. Yang, and T.-H. Cheng** (2021). Visual servoing of a moving target by an unmanned aerial vehicle. *Sensors*, **21**(17). ISSN 1424-8220. URL <https://www.mdpi.com/1424-8220/21/17/5708>.
15. **Chen, J., T. Liu, and S. Shen**, Tracking a moving target in cluttered environments using a quadrotor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016.
16. **Chowdhary, G. and E. Johnson**, A singular value maximizing data recording algorithm for concurrent learning. In *American Control Conference (ACC)*. IEEE, 2011.
17. **Clearpath Robotics** (n.d.). Husky. <http://wiki.ros.org/Robots/Husky>. Accessed: 2025-07-07.
18. **Corke, P.**, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer Publishing Company, Incorporated, 2013, 1st edition. ISBN 3642201431.
19. **Dani, A. P., Z. Kan, N. R. Fischer, and W. E. Dixon**, Structure and motion estimation of a moving object using a moving camera. In *American Control Conference (ACC)*. 2010.
20. **Dani, A. P., Z. Kan, N. R. Fischer, and W. E. Dixon**, Structure estimation of a moving object using a moving camera: An unknown input observer approach. In *50th IEEE Conference on Decision and Control (CDC) and European Control Conference (ECC)*. 2011.
21. **Danjun, L., Z. Yan, S. Zongying, and L. Geng**, Autonomous landing of quadrotor based on ground effect modelling. In *34th Chinese Control Conference (CCC)*. 2015. ISSN 1934-1768.
22. **Dasgupta, R., S. B. Roy, O. S. Patil, and S. Bhasin**, A singularity-free hierarchical nonlinear quad-rotorcraft control using saturation and barrier lyapunov function. In *2019 American Control Conference (ACC)*. 2019.
23. **Doitsidis, L., S. Weiss, A. Renzaglia, M. Achteik, E. Kosmatopoulos, R. Y. Siegwart, and D. Scaramuzza** (2012). Optimal surveillance coverage for teams of micro aerial vehicles in GPS-denied environments using onboard vision. *Autonomous Robots*. URL <https://hal.inria.fr/hal-00692534>.
24. **Emran, B. J. and H. Najjaran** (2021). Global tracking control of quadrotor based on adaptive dynamic surface control. *International Journal of Dynamics and Control*, **9**, 240–256.
25. **Falanga, D., A. Zanchettin, A. Simovic, J. Delmerico, and D. Scaramuzza**, Vision-based autonomous quadrotor landing on a moving platform. In *IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. 2017. ISSN 2475-8426.

26. **Farrell, J. A., M. Polycarpou, M. Sharma, and W. Dong** (2009). Command filtered backstepping. *IEEE Transactions on Automatic Control*, **54**(6), 1391–1395.
27. **Ferrandez, S. M., T. Harbison, T. Weber, R. Sturges, and R. Rich** (2016). Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. *Journal of Industrial Engineering and Management*, **9**(2), 374–388. ISSN 2013-0953. URL <http://www.jiem.org/index.php/jiem/article/view/1929>.
28. **Garg, T., S. B. Roy, and K. G. Vamvoudakis**, Pi-like estimator-based adaptive extremum seeking control using initial excitation. *In 61st IEEE Conference on Decision and Control (CDC)*. IEEE, 2022.
29. **Gasparino, M. V., P. K. Mishra, and G. Chowdhary**, Unmatched uncertainty mitigation through neural network supported model predictive control. *In 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, 2023.
30. **Gautam, A., P. B. Sujit, and S. Saripalli**, A survey of autonomous landing techniques for uavs. *In International Conference on Unmanned Aircraft Systems (ICUAS)*. 2014.
31. **Giernacki, W., M. Skwierczyński, W. Witwicki, P. Wroński, and P. Kozierski**, Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering. *In 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*. IEEE, 2017.
32. **Goel, R. and S. B. Roy** (2024). Composite adaptive control for time-varying systems with dual adaptation. *IEEE Transactions on Automatic Control*, 1–8.
33. **Gomez-Balderas, J.-E., G. Flores, L. García Carrillo, and R. Lozano** (2013). Tracking a ground moving target with a quadrotor using switching control: nonlinear modeling and control. *Journal of Intelligent & Robotic Systems*, **70**, 65–78.
34. **Gupte, S., Paul Infant Teenu Mohandas, and J. M. Conrad**, A survey of quadrotor unmanned aerial vehicles. *In Proceedings of IEEE Southeastcon*. 2012. ISSN 1091-0050.
35. **Ham, A. M.** (2018). Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming. *Transportation Research Part C: Emerging Technologies*, **91**, 1 – 14. ISSN 0968-090X. URL <http://www.sciencedirect.com/science/article/pii/S0968090X18304121>.
36. **Hashimoto, K. and H. Kimura**, Visual servoing with nonlinear observer. *In IEEE International Conference on Robotics and Automation (ICRA)*, volume 1. 1995.
37. **Herissé, B., T. Hamel, R. Mahony, and F.-X. Russotto** (2012). Landing a vtol unmanned aerial vehicle on a moving platform using optical flow. *IEEE Transactions on Robotics*, **28**(1), 77–89.
38. **Hewing, L., J. Kabzan, and M. N. Zeilinger** (2019). Cautious model predictive control using gaussian process regression. *IEEE Transactions on Control Systems Technology*, **28**(6), 2736–2743.
39. **Huang, Y., M. Zhu, Z. Zheng, and K. H. Low** (2021). Homography-based visual servoing for underactuated vtol uavs tracking a 6-dof moving ship. *IEEE Transactions on Vehicular Technology*, 1–1.

40. **Jabbari Asl, H.** (2019). Robust vision-based tracking control of vtol unmanned aerial vehicles. *Automatica*, **107**, 425–432. ISSN 0005-1098. URL <https://www.sciencedirect.com/science/article/pii/S0005109819302973>.
41. **Jabbari Asl, H., G. Oriolo, and H. Bolandi** (2014). An adaptive scheme for image-based visual servoing of an underactuated uav. *International Journal of Robotics and Automation*, **29**.
42. **Joshi, G. and G. Chowdhary**, Deep model reference adaptive control. *In 58th IEEE Conference on Decision and Control (CDC)*. IEEE, 2019.
43. **Joshi, G., J. Virdi, and G. Chowdhary**, Asynchronous deep model reference adaptive control. *In Conference on Robot Learning*. PMLR, 2021.
44. **Khalil, H. K.** (1996). Adaptive output feedback control of nonlinear systems represented by input-output models. *IEEE Transactions on Automatic Control*, **41**(2), 177–188.
45. **Khalil, H. K.**, *Nonlinear systems; 3rd ed.*. Prentice-Hall, Upper Saddle River, NJ, 2002.
46. **Kidger, P. and T. Lyons**, Universal approximation with deep narrow networks. *In Conference on learning theory*. PMLR, 2020.
47. **Koch, S. and M. Reichhartinger** (2019). Discrete-time equivalents of the super-twisting algorithm. *Automatica*, **107**, 190–199.
48. **Koch, S., M. Reichhartinger, and M. Horn**, On the discretization of the super-twisting algorithm. *In 58th IEEE Conference on Decision and Control (CDC)*. 2019.
49. **Koenig, N. and A. Howard**, Design and use paradigms for gazebo, an open-source multi-robot simulator. *In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3. 2004.
50. **Koubâa, A., A. Allouch, M. Alajlan, Y. Javed, A. Belghith, and M. Khalgui** (2019). Micro air vehicle link (mavlink) in a nutshell: A survey. *IEEE Access*, **7**, 87658–87680.
51. **Landau, I. D., R. Lozano, M. M’Saad, and A. Karimi**, *Adaptive control: algorithms, analysis and applications*. Springer Science & Business Media, 2011.
52. **Lavretsky, E.** (2009). Combined/composite model reference adaptive control. *IEEE Transactions on Automatic Control*, **54**(11), 2692–2697.
53. **Lee, D., T. Ryan, and H. J. Kim**, Autonomous landing of a vtol uav on a moving platform using image-based visual servoing. *In IEEE International Conference on Robotics and Automation*. 2012. ISSN 1050-4729.
54. **Lee, T.** (2012). Exponential stability of an attitude tracking control system on $so(3)$ for large-angle rotational maneuvers. *Systems Control Letters*, **61**(1), 231–237. ISSN 0167-6911. URL <https://www.sciencedirect.com/science/article/pii/S0167691111002829>.
55. **Lee, T., M. Leok, and N. H. McClamroch**, Geometric tracking control of a quadrotor uav on $se(3)$. *In 49th IEEE Conference on Decision and Control (CDC)*. 2010.

56. **Lewis, F. L., A. Yesildirek, and K. Liu** (1996). Multilayer neural-net robot controller with guaranteed tracking performance. *IEEE Transactions on neural networks*, **7**(2), 388–399.
57. **Li, J., H. Xie, K. H. Low, J. Yong, and B. Li** (2021a). Image-based visual servoing of rotorcrafts to planar visual targets of arbitrary orientation. *IEEE Robotics and Automation Letters*, **6**(4), 7861–7868.
58. **Li, J.-M., C. W. Chen, and T.-H. Cheng** (2019). Estimation and tracking of a moving target by unmanned aerial vehicles.
59. **Li, J.-M., C.-W. Chen, and T.-H. Cheng** (2021b). Motion prediction and robust tracking of a dynamic and temporarily-occluded target by an unmanned aerial vehicle. *IEEE Transactions on Control Systems Technology*, **29**(4), 1623–1635.
60. **Lin, J., Y. Wang, Z. Miao, H. Wang, and R. Fierro** (2022). Robust image-based landing control of a quadrotor on an unpredictable moving vehicle using circle features. *IEEE Transactions on Automation Science and Engineering*.
61. **Llanes, C., Z. Kakish, K. Williams, and S. Coogan**, Crazysim: A software-in-the-loop simulator for the crazyflie nano quadrotor. In *IEEE International Conference on Robotics and Automation (ICRA)*. 2024.
62. **Mahony, R., V. Kumar, and P. Corke** (2012). Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics Automation Magazine*, **19**(3), 20–32.
63. **MATLAB**, version 9.10.0 (R2021a). The MathWorks Inc., Natick, MA, 2021.
64. **Meyer, J., A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. von Stryk**, Comprehensive simulation of quadrotor uavs using ros and gazebo. In **I. Noda, N. Ando, D. Brugali, and J. J. Kuffner** (eds.), *Simulation, Modeling, and Programming for Autonomous Robots*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-34327-8.
65. **Mishra, P. K., M. V. Gasparino, and G. Chowdhary** (2025). Deep model predictive control with stability guarantees. *IEEE Transactions on Automatic Control*, 1–8.
66. **Narendra, K. S. and A. M. Annaswamy**, *Stable adaptive systems*. Courier Corporation, 2012.
67. **Nubert, J., J. Köhler, V. Berenz, F. Allgöwer, and S. Trimpe** (2020). Safe and fast tracking on a robot manipulator: Robust mpc and neural network control. *IEEE Robotics and Automation Letters*, **5**(2), 3050–3057.
68. **Oh, K.-K., M.-C. Park, and H.-S. Ahn** (2015). A survey of multi-agent formation control. *Automatica*, **53**, 424–440. ISSN 0005-1098. URL <https://www.sciencedirect.com/science/article/pii/S0005109814004038>.
69. **Ozawa, R. and F. Chaumette** (2013). Dynamic Visual Servoing with Image Moments for an Unmanned Aerial Vehicle Using a Virtual Spring Approach. *Advanced Robotics*, **27**(9), 683–696. URL <https://hal.inria.fr/hal-00807642>.

70. **O’Connell, M., G. Shi, X. Shi, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung** (2022). Neural-fly enables rapid learning for agile flight in strong winds. *Science Robotics*, **7**(66), eabm6597.
71. **Paris, A., B. T. Lopez, and J. P. How** (2019). Dynamic landing of an autonomous quadrotor on a moving platform in turbulent wind conditions.
72. **Pascual-Escudero, B., A. Nayak, S. Briot, O. Kermorgant, P. Martinet, M. Safey El Din, and F. Chaumette** (2020). Complete Singularity Analysis for the Perspective-Four-Point Problem. *International Journal of Computer Vision*. URL <https://hal.archives-ouvertes.fr/hal-03070525>.
73. **Patil, O. S., D. M. Le, M. L. Greene, and W. E. Dixon** (2021). Lyapunov-derived control and adaptive update laws for inner and outer layer weights of a deep neural network. *IEEE Control Systems Letters*, **6**, 1855–1860.
74. **Patil, O. S., D. M. Le, E. J. Griffiths, and W. E. Dixon** (2025). Lyapunov-based deep residual neural network (resnet) adaptive control. *IEEE Access*, **13**, 117943–117952.
75. **Patil, O. S., R. Sun, S. Bhasin, and W. E. Dixon** (2022). Adaptive control of time-varying parameter systems with asymptotic tracking. *IEEE Transactions on Automatic Control*, **67**(9), 4809–4815.
76. **Patre, P. M., S. Bhasin, Z. D. Wilcox, and W. E. Dixon** (2010). Composite adaptation for neural network-based controllers. *IEEE Transactions on Automatic Control*, **55**(4), 944–950.
77. **Pedro Sanchez-Cuevas, G. H. and A. Ollero** (2017). Characterization of the aerodynamic ground effect and its influence in multirotor control. *International Journal of Aerospace Engineering*, **2017**.
78. **Rolnick, D. and M. Tegmark** (2017). The power of deeper networks for expressing natural functions. *arXiv preprint arXiv:1705.05502*.
79. **Roy, S. B., S. Bhasin, and I. N. Kar**, Parameter convergence via a novel pi-like composite adaptive controller for uncertain euler-lagrange systems. In *55th IEEE Conference on Decision and Control (CDC)*. 2016.
80. **Roy, S. B., S. Bhasin, and I. N. Kar** (2018). Combined mrac for unknown mimo lti systems with parameter convergence. *IEEE Transactions on Automatic Control*, **63**(1), 283–290.
81. **Sanchez-Lopez, J. L., J. Pestana, S. Saripalli, and P. Campoy** (2014). An approach toward visual autonomous ship board landing of a vtol uav. *Journal of Intelligent & Robotic Systems*, **74**, 113–127.
82. **Scholkopf, B., S. Mika, C. J. Burges, P. Knirsch, K.-R. Muller, G. Ratsch, and A. J. Smola** (1999). Input space versus feature space in kernel-based methods. *IEEE transactions on neural networks*, **10**(5), 1000–1017.
83. **Serra, P., R. Cunha, T. Hamel, D. Cabecinhas, and C. Silvestre** (2016). Landing of a quadrotor on a moving target using dynamic image-based visual servo control. *IEEE Transactions on Robotics*, **32**, 1524–1535.

84. **Shahriari, N., S. Fantasia, F. Flacco, and G. Oriolo**, Robotic visual servoing of moving targets. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2013.
85. **Sharma, M. and I. Kar** (2021). Nonlinear disturbance observer based geometric control of quadrotors. *Asian Journal of Control*, **23**(4), 1936–1951. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/asjc.2318>.
86. **Shi, G., X. Shi, M. O’Connell, R. Yu, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung**, Neural lander: Stable drone landing control using learned dynamics. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.
87. **Shtessel, Y., C. Edwards, L. Fridman, A. Levant, et al.**, *Sliding mode control and observation*, volume 10. Springer, New York, 2014.
88. **Slotine, J.-J. E. and W. Li** (1989). Composite adaptive control of robot manipulators. *Automatica*, **25**(4), 509–519.
89. **Slotine, J.-J. E. and W. Li**, *Applied Nonlinear Control*. NJ: Prantice-Hall, Englewood Cliffs, 1991.
90. **Stanford Artificial Intelligence Laboratory et al.** (2018). Robot operating system (ros). URL <https://www.ros.org>.
91. **Swaroop, D., J. K. Hedrick, P. P. Yip, and J. C. Gerdes** (2000). Dynamic surface control for a class of nonlinear systems. *IEEE Transactions on Automatic Control*, **45**(10), 1893–1899.
92. **Tao, G.** (2014). Multivariable adaptive control: A survey. *Automatica*, **50**(11), 2737–2764. ISSN 0005-1098.
93. **The MAVROS Website** (n.d.). Mavros. <http://wiki.ros.org/mavros>.
94. **Thomas, J., J. Welde, G. Loianno, K. Daniilidis, and V. Kumar** (2017). Autonomous flight for detection, localization, and tracking of moving targets with a small quadrotor. *IEEE Robotics and Automation Letters*, **2**(3), 1762–1769.
95. **Tyagi, P., Y. Kumar, and P. Sujit**, Nmpc-based uav 3d target tracking in the presence of obstacles and visibility constraints. In *International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2021.
96. **Yang, L., X. Wang, Z. Liu, and L. Shen** (2024). Robust online predictive visual servoing for autonomous landing of a rotor uav. *IEEE Transactions on Intelligent Vehicles*, 1–17.
97. **Zhang, F., D. Dawson, M. de Queiroz, and W. Dixon** (2000). Global adaptive output feedback tracking control of robot manipulators. *IEEE Transactions on Automatic Control*, **45**(6), 1203–1208.
98. **Zhao, S., W. Dong, and J. A. Farrell**, Quaternion-based trajectory tracking control of vtol-uavs using command filtered backstepping. In *American Control Conference (ACC)*. IEEE, 2013.