Adaptive and Optimal Motion Control of Multi-UAV Systems

by

Nasrettin Köksal

A thesis presented to the University of Waterloo in fulfillment of the thesis requirement for the degree of Doctor of Philosophy in Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2019

© Nasrettin Köksal 2019
Ex exhausting Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner          Professor Farrokh Janabi-Sharifi
Supervisor                  Associate Professor Baris Fidan
Internal Members           Professor William Melek
                           Associate Professor Soo Jeon
Internal-external Member    Associate Professor Nasser Lashgarian Azad
Author’s Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.
Abstract

This thesis studies trajectory tracking and coordination control problems for single and multi unmanned aerial vehicle (UAV) systems. These control problems are addressed for both quadrotor and fixed-wing UAV cases. Despite the fact that the literature has some approaches for both problems, most of the previous studies have implementation challenges on real-time systems. In this thesis, we use a hierarchical modular approach where the high-level coordination and formation control tasks are separated from low-level individual UAV motion control tasks. This separation helps efficient and systematic optimal control synthesis robust to effects of nonlinearities, uncertainties and external disturbances at both levels, independently. The modular two-level control structure is convenient in extending single-UAV motion control design to coordination control of multi-UAV systems. Therefore, we examine single quadrotor UAV trajectory tracking problems to develop advanced controllers compensating effects of nonlinearities and uncertainties, and improving robustness and optimality for tracking performance. At first, a novel adaptive linear quadratic tracking (ALQT) scheme is developed for stabilization and optimal attitude control of the quadrotor UAV system. In the implementation, the proposed scheme is integrated with Kalman based reliable attitude estimators, which compensate measurement noises. Next, in order to guarantee prescribed transient and steady-state tracking performances, we have designed a novel backstepping based adaptive controller that is robust to effects of under-actuated dynamics, nonlinearities and model uncertainties, e.g., inertial and rotational drag uncertainties. The tracking performance is guaranteed to utilize a prescribed performance bound (PPB) based error transformation. In the coordination control of multi-UAV systems, following the two-level control structure, at high-level, we design a distributed hierarchical (leader-follower) 3D formation control scheme. Then, the low-level control design is based on the optimal and adaptive control designs performed for each quadrotor UAV separately. As particular approaches, we design an adaptive mixing controller (AMC) to improve robustness to varying parametric uncertainties and an adaptive linear quadratic controller (ALQC). Lastly, for planar motion, especially for constant altitude flight of fixed-wing UAVs, in 2D, a distributed hierarchical (leader-follower) formation control scheme at the high-level and a linear quadratic tracking (LQT) scheme at the low-level are developed for tracking and formation control problems of the fixed-wing UAV systems to examine the non-holonomic motion case. The proposed control methods are tested via simulations and experiments on a multi-quadrotor UAV system testbed.
Acknowledgments

First, I would like to thank Prof. Baris Fidan. I am indebted to him for all his contribution of time, ideas and material, which made the completion of this thesis possible.

I would also like to thank Prof. Farrokh Janabi-Sharifi, Prof. William Melek, Prof. Soo Jeon and Prof. Nasser Lashgarian Azad for being in my committee and for their helpful comments.

Here, I would like to thank Prof. Teng-Hu Cheng from National Chiao Tung University for his hospitality and support during my summer research visit in Taiwan.

I would like to extend my sincere thanks to Hao An and Mehdi Jalalmaab for countless works we have collaborated.

Finally, my special thanks go to my lovely family for all their support.
Dedication

To my beloved ones ❤
Table of Contents

List of Figures xi
List of Tables xv
List of Abbreviations xvi
List of Symbols xviii

1 Introduction 1
   1.1 General Overview and Motivation .................................. 1
   1.2 Contributions of the Thesis ...................................... 4
   1.3 Organization of the Thesis ....................................... 6

2 Background and Control Architecture 8
   2.1 Modeling of UAV Motion Dynamics ................................. 8
      2.1.1 Quadrotor UAV Motion Dynamics ............................. 9
      2.1.2 Fixed-wing UAV Motion Dynamics .......................... 12
   2.2 Literature on Single-UAV Motion Control ....................... 15
      2.2.1 Quadrotor UAV Motion Control: Nonlinearities .......... 15
      2.2.2 Quadrotor UAV Motion Control: Model Uncertainties .... 18
      2.2.3 Quadrotor UAV Motion Control: Optimality ............... 20
      2.2.4 Fixed-wing UAV Motion Control ........................... 21
2.3 Literature on Multi-UAV Formation Control ........................................... 22
2.4 Single UAV Motion Control Architecture .............................................. 24
  2.4.1 Quadrotor UAVs ................................................................. 24
  2.4.2 Fixed-wing UAVs ............................................................... 25
2.5 Multi-UAV Formation Control Architecture .......................................... 26

3 Adaptive Linear Quadratic Attitude Tracking Control with Sensor Fusion .... 27
  3.1 Introduction .............................................................................. 27
  3.2 Quadrotor UAV Dynamics .......................................................... 30
    3.2.1 Attitude Model ................................................................. 30
    3.2.2 Yaw Model .................................................................. 31
    3.2.3 Altitude Model ............................................................... 31
  3.3 Problem Statement ..................................................................... 31
  3.4 Control Approach ...................................................................... 33
  3.5 IMU Sensor Data Fusion ............................................................. 33
    3.5.1 Attitude Determination from IMU Sensors ......................... 34
    3.5.2 Attitude Estimation Using Kalman Filter ............................ 35
    3.5.3 Attitude Estimation by Complementary Filter ................... 37
  3.6 Adaptive Optimal Attitude Tracking Control Design ......................... 37
    3.6.1 Adaptive Parameter Identification Scheme ......................... 37
    3.6.2 Generic Linear Quadratic Tracking Control Design ............. 39
    3.6.3 Adaptive Linear Quadratic Tracking (ALQT) Control Design ... 40
  3.7 Yaw and Altitude Control ............................................................ 41
    3.7.1 Yaw Control .................................................................. 41
    3.7.2 Altitude Control ............................................................. 42
  3.8 Experimental Tests and Comparative Simulations ............................... 42
    3.8.1 Test Platform .................................................................. 42
3.8.2 Control Design Specifications and On-line Calculation of Control Parameters 43
3.8.3 Experimental Results 45
3.8.4 Comparative Simulations and Observations 47
3.9 Summary and Remarks 55

4 Robust Adaptive Control of a Quadrotor UAV with Guaranteed Tracking Performance
4.1 Introduction 56
4.2 Quadrotor UAV Dynamics 57
4.3 Control Problem 58
4.4 Robust Adaptive Tracking Control Design with Guaranteed Error Performance 60
  4.4.1 Position Control Design with Guaranteed Error 60
  4.4.2 Adaptive Attitude Control Design 65
4.5 System Stability Analysis 67
4.6 Alternative Adaptive Attitude Control Design 69
4.7 Simulations and Experimental Tests 69
  4.7.1 Testbed Platform and Benchmark Controllers 69
  4.7.2 Simulation Tests 71
  4.7.3 Experimental Tests 71
4.8 Summary and Remarks 82

5 Adaptive Mixing Formation Control of a Multi-UAV System
5.1 Introduction 83
5.2 Quadrotor UAV Dynamics 85
5.3 Rigid Graph Modeling of the Multi-UAV System 86
5.4 Problem Statement 87
5.5 Distributed Control Design 88
  5.5.1 High-level Control Design 88
5.5.2 Low-level Control Design ................................. 91
5.6 On-line Parameter Identification ............................. 92
5.7 Adaptive Attitude Control Laws .............................. 94
  5.7.1 Adaptive Linear Quadratic Control (ALQC) ............... 95
  5.7.2 Proposed Adaptive Mixing Control (AMC) ............... 95
5.8 Real-time Testbed and Simulations .......................... 100
  5.8.1 Real-time Testbed System ................................. 100
  5.8.2 Adding the noise effect into the simulation ............ 100
  5.8.3 Control Design Parameters ............................... 101
  5.8.4 Simulation Results ..................................... 102
5.9 Summary and Remarks ....................................... 107

6 Optimal Tracking and Formation Control of Fixed-wing UAVs 108
  6.1 Lateral Motion Model of Small Fixed-wing UAV ............. 108
  6.2 Problem Statement for Fixed-level Motion .................. 109
  6.3 High-level Control: Desired Trajectory and Heading Derivation 111
  6.4 Low-level Control: Optimal Linear Quadratic Tracking (LQT) Control Design 112
  6.5 Calculation of LQT Control Parameters ....................... 113
  6.6 Simulations and Results ................................... 114
    6.6.1 Simulation results of a single fixed-wing UAV .......... 115
    6.6.2 Simulation results of a multiple fixed-wing UAV system . 115
  6.7 Summary and Remarks ...................................... 120

7 Concluding Remarks ........................................... 121

Bibliography ..................................................... 123
# List of Figures

1.1 Autonomous system architecture for a UAV. . . . . . . . . . . . . . . . . . 2
1.2 Formation control approaches for a multi-UAV system. . . . . . . . . . . . 3

2.1 The quadrotor UAV’s attitude motion: (a) pitch; (b) roll; (c) yaw, and (d) coordinate representation with thrusts, moments and gravity force. . . . . . . . 10
2.2 An example fixed-wing UAV and its motion axes. . . . . . . . . . . . . . . 13
2.3 Motion control architecture of quadrotor UAVs [90]. . . . . . . . . . . . . . 24
2.4 Motion control architecture of small fixed-wing UAVs [133]. . . . . . . . . . 25
2.5 Formation control architecture of multi-UAV systems. . . . . . . . . . . . . 26

3.1 The overall quadrotor UAV control block diagram. . . . . . . . . . . . . . . 32
3.2 The ADIS16405 IMU Module on the Qball-X4 quadrotor UAV. . . . . . . 34
3.3 Complementary Filter. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
3.4 The Qball-X4 quadrotor UAV test platform. . . . . . . . . . . . . . . . . . 43
3.5 Off-line calculation of \( \bar{P}_6 \) for the estimate \( \hat{\theta}_\phi \in [\bar{\theta}_\phi, \bar{\phi}_\phi] \). . . . . . . . . . 44
3.6 The Qball-X4 quadrotor during the experiment. . . . . . . . . . . . . . . . 46
3.7 IMU data measurements from gyroscope. . . . . . . . . . . . . . . . . . . . 48
3.8 IMU data measurements from accelerometer. . . . . . . . . . . . . . . . . . 48
3.9 IMU data measurements from magnetometer. . . . . . . . . . . . . . . . . . 49
3.10 LS based estimate \( \hat{\theta}_\phi \) of the uncertain inertia parameter \( \theta^*_\phi \). . . . . . . . . . 49
3.11 Attitude estimation of the Qball-X4. . . . . . . . . . . . . . . . . . . . . . . 50
3.12 Attitude angle estimation of the Qball-X4 from 50 to 100 [sec].
3.13 Attitude tracking error of the Qball-X4 using complementary filter.
3.14 Attitude tracking error of the Qball-X4 using Kalman filter.
3.15 Optimal attitude control inputs for the complementary filter.
3.16 Optimal attitude control inputs for Kalman filter.
3.17 Motor PWM control inputs $v_r$.
3.18 PID vs ALQT performance comparison: attitude tracking error.
3.19 PID vs ALQT performance comparison: attitude control input.
3.20 LS based estimate $\hat{\theta}_\phi$ of the uncertain inertia parameter $\theta^*_\phi$ for the simulation.

4.1 The overall control structure for the quadrotor UAV.
4.2 Graphical representation of (4.12) for prescribed tracking error behavior with $\delta = 0.2$ and $\dot{\delta} = 1$.
4.3 Spiral trajectory tracking errors of the proposed control design, compared with the benchmark controllers.
4.4 Transient error performances for 0-50[sec].
4.5 Spiral motion with proposed control design.
4.6 Attitude tracking of the proposed control design.
4.7 LS based estimation $\hat{\theta}_\phi$ of $\theta^*_\phi$.
4.8 Tracking errors with the proposed indirect adaptive control design Section 4.4.2 and the direct adaptive control in Section 4.6.
4.9 Altitude tracking errors of the proposed control design, compared with the benchmark controllers.
4.10 Trajectory tracking of the proposed control design.
4.11 Attitude tracking of the proposed control design.
4.12 LS based estimation $\hat{\theta}_\phi$ of $\theta^*_\phi$.
4.13 Motor PWM inputs $v_r$ for the proposed control design.
4.14 Circular hovering test motion.
4.15 Trajectory tracking for the circular hovering test.
4.16 Mean Square tracking error for the circular hovering test. .......... 79
4.17 LS based estimation $\hat{\theta}_\phi$ of $\theta^*_\phi$ for the circular hovering test. .......... 79
4.18 Square hovering test motion. ........................................ 80
4.19 Waypoint tracking for the square hovering test. .................... 80
4.20 Mean Square tracking error for the square hovering test. .......... 81
4.21 LS based estimation $\hat{\theta}_\phi$ of $\theta^*_\phi$ for the square hovering test. ......... 81

5.1 Directed underlying graph of a leader-follower persistent 3D formation $F_S$ with $N$ vehicles. ........................................... 86
5.2 The Overall Control Structure for $i$th quadrotor UAV. ................ 89
5.3 Low-pass filter for reference angle generator .......................... 91
5.4 Adaptive control schemes: (a) AMC and (b) ALQC. .................... 93
5.5 Qball-X4’s pitch rate measurement for 20 [sec]. ....................... 100
5.6 Spiral formation motion in 3D for ALQC. ............................ 103
5.7 Spiral formation motion in 3D for AMC. ............................. 103
5.8 Formation distances among quadrotor UAVs for ALQC. ............. 104
5.9 Formation distances among quadrotor UAVs for AMC. ............. 104
5.10 Mean Square tracking errors for ALQC. ............................. 105
5.11 Mean Square tracking errors for AMC. ............................. 105
5.12 Bump functions of candidate subsets for the estimation $\hat{\theta}_{\phi_1}$ of the leader quadrotor UAV. ...................................... 106
5.13 $\hat{\theta}_{\phi_1}$ estimation of $\theta^*_{\phi_1}$ for the leader quadrotor UAV. .............. 106

6.1 Fixed-wing UAV formation control block diagram. .................... 111
6.2 Spiral surveillance motion of a fixed-wing UAV. ....................... 115
6.3 Lateral trajectory tracking performances. ............................ 116
6.4 Heading tracking performance. ...................................... 116
6.5 Commanded and actual velocities. .................................. 117
6.6 Commanded and actual angular velocities. ........................ 117
6.7 Spiral surveillance motion of a multiple fixed-wing UAV system in formation. 118
6.8 Formation maintenance performances for $d_{ij}$. ............................ 118
6.9 Heading tracking performances for $V_i$. ........................................... 119
6.10 Commanded and actual velocities for $V_i$. ................................. 119
6.11 Commanded and actual angular velocities for $V_i$. ......................... 120
## List of Tables

3.1 The ADIS16405 IMU Specifications [171] ........................................ 34  
3.2 The Qball-X4 quadrotor UAV dynamic parameters [172] ............... 43  
3.3 Mean square error of \( \dot{\phi} \) .................................................... 47  
3.4 Average battery consumption by \( \dot{\phi} \) ..................................... 47  
4.1 The Qball-X4 quadrotor UAV dynamic parameters ......................... 70  
5.1 The off-line calculated candidate controller gains ......................... 101  
6.1 Small fixed-wing UAV specifications used in simulations ............... 114
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UAV</td>
<td>Unmanned Aerial Vehicle</td>
</tr>
<tr>
<td>UGV</td>
<td>Unmanned Ground Vehicle</td>
</tr>
<tr>
<td>SLAM</td>
<td>Simultaneous Localization and Mapping</td>
</tr>
<tr>
<td>NGC</td>
<td>Navigation, Guidance and Control</td>
</tr>
<tr>
<td>NS</td>
<td>Navigation System</td>
</tr>
<tr>
<td>GS</td>
<td>Guidance System</td>
</tr>
<tr>
<td>CS</td>
<td>Control System</td>
</tr>
<tr>
<td>IMU</td>
<td>Inertial Measurement Unit</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>RADAR</td>
<td>Ranging or Radio Direction and Ranging</td>
</tr>
<tr>
<td>LIDAR</td>
<td>Light Detection and Ranging</td>
</tr>
<tr>
<td>SONAR</td>
<td>Sound Navigation and Ranging</td>
</tr>
<tr>
<td>P</td>
<td>Proportional</td>
</tr>
<tr>
<td>PD</td>
<td>Proportional-Derivative</td>
</tr>
<tr>
<td>PID</td>
<td>Proportional-Integral-Derivative</td>
</tr>
<tr>
<td>LQ</td>
<td>Linear Quadratic</td>
</tr>
<tr>
<td>LQR</td>
<td>Linear Quadratic Regulator</td>
</tr>
<tr>
<td>LQT</td>
<td>Linear Quadratic Tracking</td>
</tr>
<tr>
<td>ALQT</td>
<td>Adaptive Linear Quadratic Tracking</td>
</tr>
<tr>
<td>ALQC</td>
<td>Adaptive Linear Quadratic Control</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>AMC</td>
<td>Adaptive Mixing Control</td>
</tr>
<tr>
<td>ARE</td>
<td>Algebraic Riccati Equation</td>
</tr>
<tr>
<td>DRE</td>
<td>Differential Riccati Equation</td>
</tr>
<tr>
<td>BC</td>
<td>Backstepping Control</td>
</tr>
<tr>
<td>ABC</td>
<td>Adaptive Backstepping Control</td>
</tr>
<tr>
<td>PI</td>
<td>Parameter Identification</td>
</tr>
<tr>
<td>LS</td>
<td>Least Squares</td>
</tr>
<tr>
<td>PWM</td>
<td>Pulse Width Modulation</td>
</tr>
<tr>
<td>CG</td>
<td>Center of Gravity</td>
</tr>
<tr>
<td>DOF</td>
<td>Degree of Freedom</td>
</tr>
</tbody>
</table>
List of Symbols

\{O_b, x_b, y_b, z_b\} \quad \text{Body frame}

\{O_g, x, y, z\} \quad \text{Global frame}

R_m \in SO(3) \quad \text{Rotational matrix from } O_b \text{ to } O_g

p = [p_x, p_y, p_z]^T \quad \text{Position of } O_g

\varphi \triangleq [\varphi, \vartheta, \psi]^T \quad \text{Euler Angles of } O_b

w_{\varphi} = [\dot{\varphi}, \dot{\vartheta}, \dot{\psi}]^T \quad \text{Angular velocity of quadrotor UAV}

J_{\varphi} = \text{diag}(J_\varphi, J_\vartheta, J_\psi) \quad \text{Rotational Inertia Matrix of quadrotor UAV}

d_{\varphi} = [d_\varphi, d_\vartheta, d_\psi]^T \quad \text{Rotational Drag Parameters of quadrotor UAV}

F_b = [F_{xb}, F_{yb}, F_{zb}]^T \quad \text{Applied forces generated by actuator motors of quadrotor UAV}

T_r = [T_1, T_2, T_3, T_4]^T \quad \text{Total thrust forces generated by actuator motors of quadrotor UAV}

T_z, T_\varphi \quad \text{Altitude and attitude thrust forces of quadrotor UAV}

u_z, u_\varphi \quad \text{Altitude and attitude control inputs of quadrotor UAV}

v_r = [v_1, v_2, v_3, v_4]^T \quad \text{PWM control inputs of quadrotor UAV}

G \quad \text{PWM generator matrix of quadrotor UAV}

t \quad \text{Time}

i \quad \text{Number of UAVs}

r \quad \text{Number of actuator motors of quadrotor UAV}

m \quad \text{Total Mass of quadrotor UAV}

K \quad \text{Positive Armature Gain of quadrotor UAV}

K_\psi \quad \text{Thrust-to-Moment Gain of quadrotor UAV}
$b$ The actuator motor bandwidth of quadrotor UAV

$l$ Distance between CG and actuator motors of quadrotor UAV

$g$ Gravitational Acceleration

$v_c$ Lateral speed command input of fixed-wing UAV

$w_c$ Heading command input of fixed-wing UAV

$p_{zc}$ Altitude command input of fixed-wing UAV

$e$ Tracking error

$J$ Cost function

$Q, R$ Weighting Matrices

$P$ Symmetric, positive definite matrix

$g(t)$ Vector signal

$\bar{g}(t)$ Approximate vector signal

$P$ Positive covariance

$\theta^*$ Uncertain parameter

$\hat{\theta}$ Estimated parameter

$\Phi$ Regressor signal

$m_n$ Normalizing signal

$\beta$ Forgetting factor

$\epsilon$ Estimation error

$T_s$ Sampling time

$Pr(.)$ Projection operator

$\rho_j$ Smooth decreasing performance function

$S_j$ Smooth increasing function

$\varepsilon_j$ Transformed error

$\tilde{\delta}_j, \hat{\delta}_j$ Prescribed scalars

$\rho_{j0}, \rho_{j\infty}, k_j$ PPB design parameters

$pr$ Reference way-points
$V_i$ \quad $i$th UAV

$V_j$ \quad $j$th neighbor UAV

$R_{(ij)}$ \quad Relative position between responsible UAV Pairs

$R^*_{(ij)}$ \quad Desired relative position between responsible UAV Pairs

$d_{(ij)}$ \quad Distance between responsible UAV Pairs

$d^*_{(ij)}$ \quad Desired distance between responsible UAV Pairs

$n$ \quad Number of candidate controllers

$\Omega^h_i$ \quad Candidate control subset

$\eta^h_i$ \quad Bump function

$a^h_i, b^h_i$ \quad Bump function parameters
Chapter 1

Introduction

1.1 General Overview and Motivation

Recent robotic interests have focused on improvements of human-like decision-making systems such as humanoid robots, animal-like robots, unmanned vehicle systems because of their self-control mechanisms and autonomous decision abilities without any external command. These kinds of studies aim at reaching fully-autonomous systems to use in different environments and in many tasks instead of human in the near future. By this motivation, control researchers over the last decades are interested in autonomous unmanned aerial vehicles (UAVs) which are used for various defense and civilian applications. Furthermore, UAVs have been critical flight systems to perform in dangerous and unsuitable environments compared to conventional aerial vehicles since they have dynamical and design advantages such as their smaller sizes, dynamical simplicities, maneuverabilities, high performances and being unpersonalized.

In the existing UAV literature, researchers have utilized various types of UAVs in research and development of autonomous flight tasks. According to the importance of demand and supply in-flight duties, UAVs can be classified into three main types, namely, fixed-wing [20, 23], rotary, including single-rotor helicopters [1, 80], multi-rotor aerial copters [135], quadrotors [8, 30, 35], and hybrid aircrafts, including tilt-rotor UAVs [164].

Small fixed-wing UAVs recently have gained attention for particular flight tasks and especially been preferred in air defense missions instead of large and expensive aerial vehicles, e.g. Predator, Global Hawk, and Aerosonde. In particular, fixed-wing UAVs are very effective for surveillance tasks in high altitudes and they can be easily developed in different
sizes depending on flight mission requirements. As mentioned in [21], design and control methods of small fixed-wing UAVs differ from larger fixed-wing UAVs and conventional aircrafts. In earlier studies [21, 23], autonomous fixed-wing UAVs have been developed successfully with a small and strong lightweight platform, low-power consumption and facilitated system architectures including navigation, guidance and control units.

For rotary type UAVs, there are several varieties of them such as single-rotor, tri-rotor, quad-rotor, hexa-rotor, octo-rotor copters. Rotary type UAVs are able to take off and land vertically inside dangerous and hard-to-reach areas in 3D thanks to their actuator design and holonomic motion capabilities. One of the rotary type UAVs is quadrotor UAV that recently has been popular for researchers and customers. Quadrotor UAVs have favorable accurate dynamic models and stability characteristics as well as hovering at close proximity of specified locations compared to other rotary and fixed-wing UAVs. As another reason to gain more fame, they have lower cost and simple structure in design. Therefore, for research studies and commercial usage, control researchers have been interested in developing new controllers for quadrotor UAVs to provide well-formed performances for various complicated indoor and outdoor tasks such as patrol duties, agricultural activities, delivery services, surveillance and rescue. Some earlier research prototypes have been studied in [8, 30, 35, 108, 145]. In the last decade, for difficult mission applications of quadrotor UAVs, technical requirements have been increased, and accordingly it has been needed to develop new control methods and improve their performances.
In autonomous flight control missions of single-UAV systems, overall system architectures and their components are discussed using different approaches in [23, 84]. These components are gathered into three main groups: navigation, guidance, and control (NGC) systems as presented in Figure 1.1. In the literature, NGC systems are not only developed to accomplish their objectives separately, but also interrelated throughout autonomous flight missions. The purpose of NGC systems is briefly explained as follows. Navigation system (NS) provides motion (state) and environmental information to other parts of NGC systems to support the overall system architecture. The motion (state) information can be orientation, position, angular and linear velocities which are measured or estimated by using hardware equipments on single-UAVs such as inertial measurement unit (IMU) and global positioning system (GPS) as well as using software computing algorithms: Kalman filter and observers. To detect environmental information during autonomous flights, NS contains visual monitoring and sensing methods using camera, sound navigation and ranging (SONAR), light detection and ranging (LIDAR), ranging or radio direction and ranging (RADAR), infrared sensor, etc. to support guidance system (GS). In GS algorithms, desired trajectories are produced by way-point tracker to supply control system (CS) using motion and environmental information from NS. Before the desired trajectory generation, GS also includes path planning strategies to locate way-points, optimally. Overall, GS algorithms are to guide single-UAVs, e.g., both fixed-wing and quadrotor UAVs, by generating the desired position. CS is responsible for controlling and stabilizing single-UAVs.
using information from NS and GS, and then it completes autonomous flight tasks.

Another research field that has recently gained significant attention is cooperative control of multi-UAV systems. Cooperative multi-UAV systems are potentially more effective than the use of single-UAV systems in various complicated tasks including defense patrol duties, agricultural monitoring, surveillance, and rescue. One particular aspect of such cooperative multi-UAV tasks, is coordinated motion control, which involves path planning, flocking, consensus, obstacle and inter-agent avoidance, formation acquisition and maintenance [20, 22, 57, 89, 93, 112, 134]. In coordinated motion of multi-UAV systems, formation is used for a UAV team to perform certain cooperative mission requirements, optimally [163]. As presented in Figure 1.2, formation control schemes of multi-UAV systems can be classified into two main approaches as centralized and decentralized [120]. The centralized approach needs a global supervisor to coordinate all members of a UAV team, and its theoretical analysis gets complicated when the number of UAVs enlarges. Therefore, the literature works mostly focus on the decentralized approach to design easy and practical control solutions.

In this thesis, we focus on two UAV types: rotary and fixed-wing, in particular, quadrotor and small fixed-wing UAVs. By motivation of contributing to tracking control problems of single-UAVs, we develop novel and advanced (adaptive and optimal) control methods using model-based (linear and nonlinear) approaches in the CS. Also, we partially deal with the design problems of the NS and the GS during real-time flight tests. As another motivation, with the easiness of proposed flight control architectures and distributed approach in high (formation) level, we extend single UAV control designs to formation control of Multi-UAVs. Proposed techniques as presented in the main contribution chapters are practical and easily implementable as verified by experiments and real-data based high-fidelity simulations.

1.2 Contributions of the Thesis

This thesis contributes to the literature on tracking and formation control of single and multi-UAV systems, with focus on adaptive, optimal and nonlinear aspects. The main contributions of the thesis chapters are stated as follows:

- Chapter 3: A novel adaptive linear quadratic tracking (ALQT) control scheme is designed for optimal attitude tracking of a quadrotor UAV based on IMU sensor data fusion [91].
(i) The proposed control design is experimentally validated in the presence of real-world uncertainties in quadrotor UAV parameters and sensors measurement.

(ii) To improve tracking in the presence of IMU sensor noises, reliable attitude estimation schemes based on Kalman and complementary filters are designed and compared with each other via experimental tests on the quadrotor UAV.

• Chapter 4: A backstepping based robust adaptive control with guaranteed tracking error performance is proposed for trajectory tracking of a quadrotor UAV [90].

  (i) The under-actuated nonlinear dynamics is separated into three sub-models: lateral, altitude and attitude dynamics, and considered in the two-layer: position and attitude. The separation allows to easily design particular controllers for each model based on their different control demands.

  (ii) Transient and steady-state tracking performances of the quadrotor UAV are guaranteed within prescribed bounds in the presence of inertia and drag uncertainties. The effectiveness of the proposed control design is experimentally validated.

• Chapter 5: A distributed adaptive mixing control design is presented for formation maintenance of a multi quadrotor UAV system during commanded path tracking maneuvers.

  (i) A two-level control structure is introduced for constructing high (formation) and low (individual) level controllers, separately. The low-level controllers are designed to compensate effect of real dynamics issues for enhancing tracking performance and robustness while the rigid graph theory based tools are utilized for formation maintenance at the high level.

  (ii) At the low-level by using a smooth switching method based on online estimation, the adaptive mixing control (AMC) scheme is proposed to increase individual tracking performance and robustness as well as formation maintenance.

• Chapter 6: Optimal trajectory tracking control of fixed-wing UAVs and formation control of a multiple fixed-wing UAV system are designed for 2D surveillance tasks.

  (i) The proposed single-UAV LQT control and multi-UAV two-level control designs are extended and applied to fixed-wing UAV systems for fixed altitude 2D trajectory tracking and formation control problems.
1.3 Organization of the Thesis

The thesis consists of one background and literature review chapter and four contribution chapters. Results, discussions, and summaries of numerical simulations and experiments are provided in each contribution chapter to make it self-contained.

Chapter 2 presents modeling of UAV motion dynamics for quadrotor and small fixed-wing UAV systems. Then, the literature is reviewed for single-UAV motion control and multi-UAV formation control of quadrotor and small fixed-wing UAV systems. Lastly, control architectures are presented and discussed.

Chapter 3 presents an infinite-horizon ALQT control scheme for optimal attitude tracking of a quadrotor UAV. The proposed control scheme is experimentally validated in the presence of real-world uncertainties in quadrotor system parameters and sensor measurement. The designed control scheme guarantees asymptotic stability of the closed-loop system with the help of complete controllability of the attitude dynamics in applying optimal control signals. To achieve robustness against parametric uncertainties, the optimal tracking solution is combined with an on-line least squares based parameter identification scheme to estimate the instantaneous inertia of the quadrotor. Sensor measurement noises are also taken into account for the on-board IMU sensors. To improve controller performance in the presence of sensor measurement noises, two sensor fusion techniques are employed, one based on Kalman filtering and the other based on complementary filtering. The ALQT control performance is compared for the use of these two sensor fusion techniques.

In Chapter 4, a backstepping based robust adaptive control design with guaranteed transient and steady-state tracking performances is proposed for a quadrotor UAV. Backstepping techniques, combined with a prescribed performance function based error transformation, are employed to achieve the bounded transient and steady-state tracking errors of the strict-feedback position system which comprises of the lateral position and the altitude dynamics. To compensate the effects of model uncertainties such as inertia and drag uncertainties on attitude regulation, an indirect adaptive control scheme is designed, where the least squares based parameter identification algorithm is combined with a backstepping based nonlinear control law. Simulation and experimental test results are provided to verify the effectiveness of the proposed control design.

Chapter 5 presents a distributed adaptive mixing control design for the formation maintenance of a multi quadrotor UAV system during commanded path tracking maneuvers. The formation control design is constructed at the two-level: At the high (formation) level, the rigid and persistent motion is satisfied in 3D to maintain the predefined for-
formation shape. At the low (individual) level, an indirect adaptive mixing control (AMC) law is designed based on Least Squares (LS) parameter identification (PI) to ensure better tracking performances and robustness to parametric uncertainties and disturbances in quadrotor UAV equations of motion. The proposed scheme adaptively blends a set of pre-designed linear quadratic control gains based on the bump function, and it provides smooth transition between pre-designed control sets. The proposed scheme is also compared with an adaptive linear quadratic control (ALQC) design. Stability analyses of both controllers are provided. Formation performances are tested and compared by real-time based simulations.

In Chapter 6, a linear quadratic tracking (LQT) control design is studied for the lateral motion tracking of fixed-wing UAVs. Then, the low-level LQT control design is extended to the formation control of a multiple fixed-wing UAV system by using the two-level, hierarchical, distributed formation structure. For both cases, the proposed control designs are validated for 2D surveillance tasks by numerical simulations.

Chapter 7 offers conclusions and discussions on the proposed methods.
Chapter 2

Background and Control Architecture

This chapter first introduces equations of motion for quadrotor and small fixed-wing UAV systems. Then, regarding current control strategies and problems, the literature is reviewed on tracking and formation control of single and multi-UAV systems for both UAV types. Lastly, control architectures of single and multi-UAV systems are discussed and presented.

2.1 Modeling of UAV Motion Dynamics

UAV systems can be classified according to many different aspects based on their dynamics, design structure, motion facility or working environment, etc. In this thesis, depending on the taking-off and landing abilities of UAV systems, we consider that there are two types, namely, quadrotor and small fixed-wing UAV systems. In the literature, both systems have various equations of motion models as presented in [8, 20, 23, 30, 35, 41, 64, 88, 108, 133]. These motion models can consist of highly nonlinear dynamics with coupled states and aerodynamics parameters, simplified nonlinear models with ignored effects, linearized forms neglecting many important aspects, kinematic models avoiding complex control design, dynamical uncertainties, and environmental or model disturbances.

We first define the equations of motion models in the form of a generic nonlinear representation for both UAV types to efficiently utilize in further control analyses. On $\mathbb{R}^3$, the equations of motion models are written in the nonlinear model as

$$\dot{X} = F(X, u),$$ (2.1)
where \( X \in \mathbb{R}^n \) and \( u \in \mathbb{R}^m \) are state and control input vectors of UAV systems, respectively.

### 2.1.1 Quadrotor UAV Motion Dynamics

The coordinates of the quadrotor UAV system’s body frame \( \{ O_b, x_b, y_b, z_b \} \) centered at the center of gravity (CG), the global frame \( \{ O_g, x, y, z \} \), thrusts, moments and gravity are represented in Figure 2.1. Using Euler angles \( \varphi \triangleq [\phi, \vartheta, \psi]^T \) and the rotational matrix \( R_m \in SO(3) \) from \( O_b \) to \( O_g \), and following the Newton-Euler formalism, nonlinear dynamics of the quadrotor UAV are described in terms of applied forces and moments, as

\[
F = R_m F_b = m \ddot{p} \in \mathbb{R}^3 \quad \text{and} \quad M = J_{\varphi} \dot{\varphi} + w_{\varphi} \times J_{\varphi} w_{\varphi} \in \mathbb{R}^3, \quad (2.2)
\]

where \( F_b = [F_{xb}, F_{yb}, F_{zb}]^T = [0, 0, \sum_{r=1}^{4} T_r]^T \) is the applied force vector generated by actuators’ thrust forces \( T_r, \ r = 1, 2, 3, 4 \), in the body frame; \( m \) is the total mass of the system; \( J_{\varphi} = \text{diag}(J_{\phi}, J_{\vartheta}, J_{\psi}) \) is the rotational inertia matrix in the body frame; \( w_{\varphi} = [\dot{\varphi}, \dot{\vartheta}, \dot{\psi}]^T \) is the angular velocity of \( O_b \). Equations (2.2) lead to the following equations of motion [89]:

\[
\begin{align*}
\ddot{p}_x &= \frac{(T_1 + T_2 + T_3 + T_4)(\sin \psi \sin \phi + \cos \phi \sin \vartheta \cos \psi)}{m}, \\
\ddot{p}_y &= \frac{(T_1 + T_2 + T_3 + T_4)(-\sin \phi \cos \psi + \cos \phi \sin \vartheta \sin \psi)}{m}, \\
\ddot{p}_z &= \frac{(T_1 + T_2 + T_3 + T_4)(\cos \phi \cos \vartheta)}{m} - g, \\
\ddot{\phi} &= \frac{l(T_1 - T_2)}{J_{\phi}} + \frac{(J_{\vartheta} - J_{\psi}) \dot{\psi} \dot{\vartheta}}{J_{\vartheta}} - d_{\phi} \ddot{\phi}, \\
\ddot{\vartheta} &= \frac{l(T_3 - T_4)}{J_{\vartheta}} + \frac{(J_{\phi} - J_{\psi}) \dot{\varphi} \dot{\psi}}{J_{\phi}} - d_{\vartheta} \ddot{\vartheta}, \\
\ddot{\psi} &= \frac{K_{\psi}(T_1 + T_2 - T_3 - T_4)}{J_{\psi}} + \frac{(J_{\psi} - J_{\vartheta}) \dot{\varphi} \dot{\psi}}{J_{\psi}} - d_{\psi} \ddot{\psi},
\end{align*}
\]

where \( p = [p_x, p_y, p_z]^T \) is the position of \( O_b \); \( d_{\varphi} = [d_{\phi}, d_{\vartheta}, d_{\psi}]^T \) are rotational drag parameters; \( T_r, \ r = 1, \ldots, 4 \) are thrust forces generated on each actuator; \( l \) is the distance between the center of gravity (\( O_b \)) and each propeller; \( K_{\psi} \) is thrust-to-moment gain; \( g \) is gravitational acceleration.
Besides, as an attempt to generate thrust forces using actuators, we use the first-order thrust-input model \[ T_r(s) = K \frac{b}{s+b} v_r(s) \] (2.4)

where \( b \) is the actuator bandwidth; \( K \) is a positive armature gain.

In the control design, we separate the attitude and altitude dynamics. Using the control inputs of separated dynamics, a pulse width modulation (PWM) input generator is
obtained as

\[ v_r = Gu = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & -1 & 1 & 1 \\ 1 & 0 & -1 & 1 \\ -1 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} u_\varphi \\ u_z \end{bmatrix}, \quad (2.5) \]

where \( v_r = [v_1, v_2, v_3, v_4]^T \in \mathbb{R}^4 \) is the PWM input for each actuator; \( G \) is the PWM generator matrix; \( u_\varphi = [u_\phi, u_\vartheta, u_\psi]^T \in \mathbb{R}^3 \) is attitude control inputs; \( u_z \in \mathbb{R} \) is altitude control input. Employing (2.5) we map the generated control signals \( u = [u_\varphi^T, u_z]^T \) to the actual PWM signals \( v \) for the four motors. Similar to (2.5), we define the effective altitude thrust \( T_z \triangleq (T_1 + T_2 + T_3 + T_4)/4 \), (2.6)

\[ T_\phi \triangleq (T_1 - T_2)/2, \quad (2.7) \]

\[ T_\vartheta \triangleq (T_3 - T_4)/2, \quad (2.8) \]

\[ T_\psi \triangleq (T_1 + T_2 - T_3 - T_4)/4. \quad (2.9) \]

Combining the nonlinear dynamics (2.3), thrust-input model (2.4) and the relation (2.6)-(2.9), we derive the nonlinear state variable model in the form of (2.1) as

\[ \dot{X} = F(X, u) = \begin{bmatrix} X_3 \\ X_4 \\ \frac{4}{m} f_1(X_2)X_6 - \zeta \\ A_1 X_4 + A_2 f_2(X_4) + B X_5 \\ -bX_5 + Kbu_\varphi \\ -bX_6 + Kbu_z \end{bmatrix}, \quad (2.10) \]

with state and control input vectors of the quadrotor UAV

\[ X = [X_1, X_2, X_3, X_4, X_5, X_6]^T \in \mathbb{R}^{16} \quad \text{and} \quad u = [u_\varphi^T, u_z]^T \in \mathbb{R}^4, \quad (2.11) \]

where \( X_1 = p = [p_l^T, p_z]^T, X_2 = \varphi \triangleq [\phi, \vartheta, \psi]^T, X_3 = \dot{X}_1 = v, X_4 = \dot{X}_2 = w_\varphi, X_5 = T_\varphi \triangleq [T_\phi, T_\vartheta, T_\psi]^T \) are 3-D vectors; \( X_6 = T_z; \zeta = [0, 0, g]^T; A_1 = \text{diag}(d_\phi, d_\vartheta, d_\psi); A_2 = \)
\[
\begin{aligned}
\text{diag}\left(\frac{J}{J_\theta}, \frac{J}{J_\psi}, \frac{J}{J_\varphi}\right); & \quad f_1(X_2) = \begin{bmatrix}
\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\
\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\
\sin \phi \cos \theta
\end{bmatrix}; & \quad f_2(X_4) = \begin{bmatrix}
\dot{\phi} \\
\dot{\psi} \\
\dot{\theta}
\end{bmatrix}; & \quad B = \\
\text{diag}(\frac{\sigma_\varphi}{J_\varphi}) = \text{diag}(\frac{2l}{J_\phi}, \frac{2l}{J_\phi}, 4K_\psi).
\end{aligned}
\]

In order to have a more feasible problem setting for further advanced control design, we now present the separated dynamics of (2.10) as follows.

**Lateral Position Model:** The separated lateral position dynamics is

\[
\begin{align*}
\dot{p}_l &= v_l, \quad (2.12) \\
\ddot{p}_l &= \frac{4T_z}{m} f_{1l}, \quad (2.13)
\end{align*}
\]

where \( f_{1l} = \begin{bmatrix}
\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\
\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\
\sin \phi \cos \theta
\end{bmatrix}. \)

**Altitude Model:** The separated altitude dynamics is

\[
\begin{align*}
\dot{p}_z &= v_z, \quad (2.14) \\
\ddot{p}_z &= \frac{4}{m}(\cos \phi \cos \theta)T_z - g, \quad (2.15) \\
\dot{T}_z &= -bT_z + Kbu_z, \quad (2.16)
\end{align*}
\]

**Attitude Model:** The separated attitude dynamics is

\[
\begin{align*}
\dot{\phi} &= w_\phi, \quad (2.17) \\
\dot{\psi} &= A_1 w_\psi + A_2 f_2(w_\psi) + BT_\varphi, \quad (2.18) \\
\dot{T}_\varphi &= -bT_\varphi + Kbu_\varphi \quad (2.19)
\end{align*}
\]

**Remark 2.1.1.** We are now able to design separated controllers to obtain more feasible control performances based on the self-control demands of the sub-models. In both altitude and attitude dynamics, an additional dynamics is defined to obtain \( u_z \) and \( u_\varphi \) which can be utilized to generate the PWM control inputs \( v_r \) of (2.4).

### 2.1.2 Fixed-wing UAV Motion Dynamics

Fixed-wing UAV systems have the six-degree-of-freedom (DOF) and servo command inputs which are aileron (a), elevator (b), rudder (c), and throttle (d) as shown in Figure 2.2.
The dynamics of fixed-wing UAVs consists of twelve-states which contain coupled states, model nonlinearities, nonlinear aerodynamics parameters, wind and other disturbances [23]. These complexities in the motion dynamics do not allow to develop advanced control methodologies, easily. On the other hand, small fixed-wing UAV systems are mainly considered to develop advanced control methods compared with large fixed-wing UAVs and conventional aircrafts. Small fixed-wing UAV’s simplified equations of motion models (piccolo-controlled type) are suitable for advanced control solutions as studied in [20, 133].

In general, roll and yaw dynamics provide the lateral motion of the small fixed-wing UAV since there is a coupling between each other. In the piccolo-controlled type lateral motion, roll dynamics is ignored, and it is stabilized by autopilot devices. Hence, the simplified yaw and lateral velocity models provide lateral motion control [20, 133]. Since the angle of attack and pitch models are ignored and stabilized by autopilot devices, the altitude (longitudinal) motion only depends on the piccolo-controlled type height dynamics [20, 133]. Therefore, the simplified nonlinear equations of the small fixed-wing UAV motion (piccolo-controlled type) are considered for the position (lateral and altitude) and attitude as follows.

\[
\begin{align*}
\dot{p}_x &= v \cos(\psi), \\
\dot{p}_y &= v \sin(\psi), \\
\dot{p}_z &= \frac{1}{\alpha_z} \dot{\phi} + \frac{1}{\alpha_z} (p_{zc} - p_z), \\
\dot{v} &= \frac{1}{\alpha_v} (v_c - v), \\
\dot{\psi} &= \omega, \\
\dot{\omega} &= \frac{1}{\alpha_\omega} (\omega_c - \omega),
\end{align*}
\]
where \( p = [p_x, p_y, p_z]^T \) are the position of \( O_g \); \( v \) is lateral speed; \( \psi \) is heading angle; \( v_c, w_c \) and \( p_{zc} \) are lateral speed, heading and altitude control inputs. Then, the nonlinear state variable model in the form of (2.1) is derived as

\[
X = F(X, u) = \begin{bmatrix}
X_3 \cos(X_4) \\
X_3 \sin(X_4) \\
\frac{-1}{\alpha_v} X_3 + \frac{1}{\alpha_v} v_c \\
X_5 \\
\frac{-1}{\alpha_w} X_5 + \frac{1}{\alpha_w} w_c \\
X_7 \\
\frac{-1}{\alpha_z} X_7 - \frac{1}{\alpha_z} X_6 + \frac{1}{\alpha_z} p_{zc}
\end{bmatrix},
\] (2.26)

with state and control input vectors of the small fixed-wing UAV

\[
X = [X_1, X_2, X_3, X_4, X_5, X_6, X_7]^T \in \mathbb{R}^7 \quad \text{and} \quad u = [v_c, w_c, p_{zc}]^T \in \mathbb{R}^3,
\] (2.27)

where \( X_1 = p_x \), \( X_2 = p_y \), \( X_3 = v \), \( X_4 = \psi \), \( X_5 = w \), \( X_6 = p_z \), and \( X_7 = \dot{p}_z \); \( \alpha_v, \alpha_w, \alpha_z \) and \( \alpha_z \) are inertial related dynamical parameters.

For further control designs, (2.26) are separated to the lateral and the altitude motion models as follows.

**Lateral Model:** The separated lateral dynamics is

\[
\begin{align*}
\dot{p}_x &= v \cos(\psi), \\
\dot{p}_y &= v \sin(\psi), \\
\dot{\psi} &= \omega, \\
\dot{\omega} &= \frac{1}{\alpha_w} (\omega_c - \omega), \\
\dot{v} &= \frac{1}{\alpha_v} (v_c - v).
\end{align*}
\] (2.28-2.32)

**Altitude Model:** The separated height dynamic is

\[
\ddot{p}_z = -\frac{1}{\alpha_z} \dot{p}_z + \frac{1}{\alpha_z} (p_{zc} - p_z).
\] (2.33)
2.2 Literature on Single-UAV Motion Control

In the literature, various control techniques and approaches have been studied extensively for quadrotor and fixed-wing UAV systems since both have different motion characteristics and control demands. This section presents a brief summary of the single-UAV motion control literature for both UAV systems. As discussed in the following subsections, it is noticed that some control methods have been studied for nonlinearity, uncertainty and optimality problems. However, the literature still needs advanced and feasible control solutions and analyses for realistic model issues such as model nonlinearities, parametric uncertainties, sensor noises and disturbances. By this motivation, new control strategies can be developed and combined with each other to design real-time implementable, optimal and robust controllers with guaranteed tracking error performance.

2.2.1 Quadrotor UAV Motion Control: Nonlinearities

It is well known that linear methods are easy and straightforward in terms of control designs and stability analyses. However, linear control schemes perform poorly for highly nonlinear systems since ignored dynamical terms affect linearized models, negatively. On the other hand, nonlinear control schemes have some disadvantages since they require more sophisticated hardware and equipment with larger memory and faster processor in real-time implementation. Also, their stability analysis is not easy for proof.

In the literature, the dynamics of quadrotor UAVs may consist of highly coupled states, nonlinear aerodynamic coefficients, external disturbances or uncertain parameters. To handle these effects, more advanced control designs are required. Before advanced methods were developed, there exist proportional-integral-derivative (PID) and linear quadratic (LQ) control based classical studies for simplified and linearized dynamics [29, 30, 36]. On the other hand, some earlier works [31, 107, 108] are studied solving nonlinear effects by using nonlinear control techniques such as feedback linearization, sliding mode, and backstepping control methods. These studies are partially implemented on quadrotor UAVs with restricted dynamics and without full autonomous flights. Using visual feedback without GPS and accelerometer, the approach described in [8] presents feedback linearization and backstepping-like control laws. The implementation of the study is not fully autonomous since the quadrotor UAV is restricted in vertical and yaw motion.

Unlike Newton-Euler modeling, [35] uses Lagrange approach to derive the motion dynamics for the control analyses. The authors use a nested saturation strategy to design the
proposed controllers, and the system is stabilized based on Lyapunov analysis. The experiment is technically not full autonomous flight since position and orientation sensing are provided via cables. In [41], the coupled dynamics derived by Lagrangian method is used to design backstepping control. The control structure is considered in two parts; attitude inner and position outer loop controllers. The control design is not easy and straightforward since position dynamics is bilinear. In the design, two neural networks are used for the estimation of system unknowns. In [107, 108], the dynamics is analyzed into three interconnected subsystems: under-actuated (for lateral and rotation), fully-actuated (for altitude and yaw) and propeller (for four rotors) subsystems. Then, the full-state backstepping methods are used for tracking control of the quadrotor UAV. The system stability is based on Lyapunov theorem. In [108], tests are lack of autonomous flight because of restricted yaw and altitude motions.

In contrast to the studies mentioned above, [145, 146] use the quaternion representation to define the attitude behavior. This helps to avoid singularities in the calculation of control laws comparing with Euler representation. In particular, [145] focuses on the attitude stabilization using the quaternion-based feedback control. Then, orientation is stabilized using $PD^2$ feedback control. This stabilized controller achieves a good transient and disturbance rejection performances during the high speed and large angle motions. [146] also contains the unit-quaternion, which is globally nonsingular, without velocity measurement to design a controller for tracking of desired attitude motion. Another control method is geometric tracking control to avoid singularities of Euler angles and ambiguities of quaternions in the attitude behavior during the complex and acrobatic maneuvers. This approach presents a global dynamic definition for the avoidances. Using the geometric method, [96] present nonlinear tracking controller which is used on the Euclidean group $SE(3)$. The study is extended to [97] which presents nonlinear output-tracking controllers for tracking of translational and rotational models. In this work, the authors consider that both dynamics contain the bounded uncertainties which are neglected in [96].

For robust nonlinear solutions on quadrotor UAVs, the sliding mode controllers (SMC) are designed in [142, 143] to track the desired position and yaw, and stabilization. For overall control design, the motion equations are considered in two parts: fully-actuated and under-actuated subsystems. The authors use a continuous approximation of sign function to avoid chattering effects in the controller. In [25], a feedback linearization-based controller is designed together with high order sliding mode (HOSM) observer against external disturbances such as wind and noise. The HOSM observer is also used for state measurements. Although the feedback linearization avoids nonlinearities, there still occurs the external disturbances. Hence, the HOSM observer is not only used to restraint the disturbances by estimation, but it also supports the stability and robustness of the closed-loop system.
[94] presents two nonlinear controllers based on either Feedback linearization (FL) and adaptive sliding mode (ASM) controllers. The FL controller is sensitive for external disturbances and sensor noises since it includes the high-order derivative terms. In the ASM, the controller has a robust structure to cancel model errors, external disturbances, and sensor noises. Without estimation of model uncertainties, sliding mode controller generates large input gains, and this causes application problems because of the motor-limitations of actual systems. Therefore, to avoid these uncertain effects, the authors use an adaptation in the SMC design. To make more easier analyses, the authors have simplified nonlinear dynamics for both control design.

For robust controllers against uncertainties and external disturbances, researchers have developed several control approaches such as robust nonlinear control methods, observers, robust compensators. In [27], the SMC based on disturbance observer is studied. This approach gives a robust continuous control in the presence of uncertainties and disturbances. The control design also avoids high control gains which cause inapplicable cases in practice. [139] presents a linear matrix inequality based controller gain synthesis. Using an approximate FL, the control gains are easily tuned. The design is not only achieved optimal gains for the cost function, but it also guarantees robust stability performance. [138] focuses on attitude performance. By using Lyapunov methodology, the nonlinear control design is developed to eliminate model uncertainties. The on-line estimation model is designed using a time-delay approach. Combining the anti-windup technique with the controller, the robustness of the system is proven.

Compared with the above mentioned robust designs, [102, 103, 104, 105] use a robust compensation approach combining with nominal control designs such as PD. The overall control design eliminates the effects of model nonlinearities, model uncertainties, and external disturbances. In these studies, nonlinearities and external disturbances are gathered into an uncertainty representation in the linear model, and they are taken as multiple uncertainties. Nonlinear dynamics are not ignored since they are taken into account as multiple uncertainties. Then, the robust decentralized control is designed within two control loops: the nominal control design which are position and attitude controllers, and the robust external compensator which deals with the control inputs. The nominal control design provides translational and rotational tracking. In the robust compensator, the system inputs are regulated. Combining the nominal control inputs and the compensator inputs, the effective control inputs are generated for the actual quadrotor UAV.
2.2.2 Quadrotor UAV Motion Control: Model Uncertainties

In some cases, quadrotor UAVs face model (dynamical) uncertainties while they operate to achieve trajectory tracking and stability objectives. These uncertainties come from unmeasurable or unknown parameters in the motion dynamics. One of the control techniques in the literature to solve dynamical uncertainties is adaptive control by using a direct method or estimating unknown parameters via an indirect method. Direct adaptive control approach updates uncertain parameters in control laws, directly. Indirect adaptive law is designed to estimate unknown parameters, separately and then calculating control gains.

In [115], the direct adaptive control is developed using backstepping methods for the tracking problem. Considering the unknown mass parameter, [73] uses an under-actuated model divided into three subsystems to design an adaptive backstepping controller. The authors use Lyapunov theorem to prove the stability of translational and yaw tracking. The stability analyses are considered to ensure adaptive control laws and estimation models. In [166], the authors develop on-line estimation laws to design a nonlinear adaptive regulation controller. In the proposed design, unknown parameters, moments of inertia, aerodynamic damping coefficients, length, and force-to-moment factor are estimated, then using estimated parameters, the effects of uncertainties are compensated via the control design. The stability of the system is proven by Lyapunov method under parametric uncertainties.

[45] presents a direct adaptive control application by using model reference adaptive control (MRAC) method. The control design is applied to the linearized model under parametric uncertainties. The authors also develop a nominal controller to compare performances of both design. The flight tests show that the MRAC design is more effective for robust responses than the nominal control design. In [46], nominal control, MRAC and combined/composite MRAC (CMRAC) design based on Lyapunov theorem are compared. The MRAC uses the direct adaptive method following [45]. In the CMRAC, the authors develop a combination of direct and indirect adaptive control. The proposed CMRAC aims to accomplish a smoother transient performance than others. All control designs are compared to each other using an indoor test facility. Both adaptive controllers give more robust responses against parametric uncertainties, in particular, actuator failure.

There are several studies on indirect adaptive control designs for linear models of quadrotor UAVs. A method proposed in [9] utilizes an indirect adaptive controller by using recursive least squares (LS) estimation for developing a linear parameter varying (LPV) controller. The study lacks altitude and yaw analyses. [89] presents an indirect adaptive linear quadratic controller (ALQC) considering inertial uncertainties for the pitch/roll dynamics. An on-line parameter identification (PI) is developed via the LS algorithm. In [33], an indirect mixing adaptive method is combined with the actuator failure problem.
Regarding intelligent estimation, [170] presents a neural network-based adaptive control under various uncertainties. In the design, the author uses a norm estimation approach instead of element-wise estimation to improve the real-time capability of the system, and the estimation method also saves the on-board computational resource. In [19], a radial basis function neural network (RBFNN) is used to approximate the perturbations and combined with backstepping. Adjustment of the RBFNN is based on on-line learning, and the RBFNN design does not need prior knowledge of uncertainties and disturbances.

Unlike most of the existing estimation based adaptive control literature, [119] develops a nonlinear function approximator, direct approximate, based on the Cerebellar Model Arithmetic Computer (CMAC). This method has fast adaptation and computation performance, and it is well-fit for applying with a direct adaptive controller. However, the control design has a weakness against sinusoidal disturbances. In the CMAC, the controller is adapted to the unknown payload and compensated disturbances. For the system robustness, the update method limits weight growth by catching large enough values to compensate the effects of unknown payloads. Under parametric and non-parametric uncertainties, a decentralized adaptive control method is presented in [114] to stabilize altitude and attitude dynamics and to cancel the effects of the uncertainties. The controller is asymptotically stabilized by a Lyapunov-based MRAC technique. Each dynamic channel is tuned by itself based on its error, and this model has a simple structure compared with existing adaptation methods. [63] uses a novel unified passivity-based adaption with the backstepping procedure to overcome the effects of the uncertain mass. The control approach consists of two main objectives which are velocity field following and timed trajectory tracking.

In some instances, uncertainties and disturbances are considered together to develop robust controllers. [167] presents a robust adaptive nonlinear control design to satisfy tracking performances. The robust integral of the signum of the error (RISE) method and the immersion/invariance-based adaptive control method are used in inner and outer loops of the quadrotor UAV to overcome the effects of parametric uncertainties and unknown external disturbances. Using Lyapunov and LaSalle’s invariance-based analyses, the stability of the system is proven. In [76, 77], considering modeling error and disturbance uncertainties associated with aerodynamic and gyroscopic effects, payload mass, and other external forces/torques which come from the flying environment, the authors develop a robust adaptive tracking controller for nonlinear and linear models using the same control methodology. Using Lyapunov-energy function, the controllers are developed, and the stabilities are proven. For both, adaptive laws are designed to overcome modeling errors and disturbance uncertainties, and they do not need prior bound knowledge. In [77], PD-like control, gravity compensator, desired acceleration and desired angular acceleration models, and adaptive laws are combined to obtain position and attitude controllers.
In some cases, external uncertainties arise from disturbances such as external forces, wind, noises, etc. As a solution, disturbance rejection methods are used for these uncertainties. For constant wind disturbances, [34] present a globally stabilized robust path tracking controller by utilizing a disturbance rejection design. The control is based on Lyapunov-based backstepping methods to guarantee zero tracking error. Simulation and experimental results show the effectiveness and robustness of the design. In [148], the authors design a disturbance rejection control for internal and external disturbances in attitude dynamics. The design includes a robust disturbance-observer (DOB) and a nonlinear feedback control strategy. The DOB design uses an optimal approach, $H_\infty$ theory, to overcome disturbance effects. The attitude tracking error model is developed using modified Rodrigues parameters (MRPs). The nonlinear feedback control is also based on backstepping techniques. In [147], the authors present another DOB based control strategy to overcome the effects of modeling error and external disturbances.

### 2.2.3 Quadrotor UAV Motion Control: Optimality

In the literature, one of the main control interests is to generate optimal control actions for quadrotor UAVs. These control methods provide effective control actions with optimal tracking and low energy consumption. Comparing with other control approaches, there are a few optimal control solutions applied to quadrotor UAVs. linear quadratic regulator (LQR), $H_\infty$ and model predictive control (MPC) are generally used as earlier solutions.

In the earlier works, there firstly exist LQR implementations on linearized models [29, 30, 36]. The LQR method is combined with adaptive and fault tolerance solutions under parametric uncertainties in [33, 89]. These studies are experimentally validated, and the results show the effectiveness of adaptive LQR designs. Furthermore, $H_\infty$ techniques are used in some studies. Despite several $H_\infty$ strategies are developed using a linear model, there exists an $H_\infty$ design for 2DOF based on the simplified nonlinear model in [37]. [125] present a nonlinear $H_\infty$ control to stabilize rotation and to support backstepping strategy in translational, optimally. In [126], a robust nonlinear $H_\infty$ control and integral MPC models are used for inner and outer loops in the overall design. As an optimal nonlinear control, [148] present a robust disturbance observer based on $H_\infty$ strategy to provide a robust and optimal performance during the disturbance rejection.

Taking into account model constraints and disturbances, MPC is used as another optimal approach. [4] presents a robust and optimal MPC design under constraints and wind-gust disturbance for the attitude model. A set of Piecewise Affine (PWA) models is designed for each linearized subset attitude model. In [6], the authors extend the attitude
control [5] to the full motion control design. A switching MPC is used for translation and rotation dynamics. A robust MPC is designed in [7] to decrease the effects of disturbances for tracking performance. As existing optimal control methods, in [3], a constrained finite time optimal control (CFTOC) is designed to stabilize the experimental attitude model under constraints and wind disturbance. In another study, [156] uses $L_1$ optimal robust control for a quadrotor UAV system. The control strategies are developed using feedback linearization. The control design is implemented for no measurement noise and noise cases.

2.2.4 Fixed-wing UAV Motion Control

This subsection reviews control methodologies that are studied and applied to small fixed-wing UAVs in the earlier works. Regarding advanced nonlinear, optimal and adaptive approaches studied on the low-level control design of fixed-wing UAVs, the literature is restricted since the motion dynamics of fixed-wing UAVs are complicated and highly nonlinear [23, 55, 66, 106]. In some studies [20, 81, 133], fixed-wing UAVs are considered and studied as small vehicles with simplified kinematic and dynamic motion models to avoid highly nonlinearities. Therefore, by applying their control methods on the simplified models, the commanded heading angle, velocity, and altitude are generated for autopilot avionic devices which are inner-loop control equipment to generate actual motor inputs for aileron, elevator, rudder, and throttle. Furthermore, these control approaches can be efficiently utilized on unmanned ground vehicles (UGVs) for experimental validations in 2D since the simplified planar motion models of fixed-wing UAVs are well-suited to UGVs (since same non-holonomic motion characteristics). As a case study, [132] presents an experimental validation of the trajectory tracking control design by using a mobile robot platform as a fixed-wing UAV motion control at the fixed-level.

In [133], the authors develop a constrained nonlinear tracking control for the simplified kinematic and dynamic motion models of the fixed-wing UAV. The control model is designed with low-level altitude-hold, velocity-hold and heading-hold autopilots which are represented lateral and longitudinal motion. These autopilots reduce the 12-state strongly nonlinear model to 6-state equations of motion model which uses altitude, heading and velocity command inputs. Hence, using the 6-state motion model and considering system constraints, Control Lyapunov Function (CLF) approach is used for the overall control design. By the help of equipping low-level autopilots, [131] also uses an accurate 7-state kinematic model for nonlinear backstepping control to derive high-level velocity and roll angle control laws. In another study, [81] considers the kinematic model with the autopilots as low-level and develops a nonlinear model predictive control (NMPC) for the high-level controller before the low-level autopilot avionics. This study deals with solving
on-line an optimal trajectory tracking problem under limited turn. In [20], the authors use the Piccolo-controlled model, which is another simplified motion model with autopilots, to develop a Lyapunov-based backstepping controller in lateral and longitudinal motion. For lateral motion tasks, [61] presents a direct model reference adaptive approach with constant speed for the low-level heading control. As a realistic perspective, [18] presents the state estimation techniques via various methods and real-time sensors equipped with the fixed-wing UAV. It also discusses the control and stability details of autopilot avionics before the decoupled lateral and longitudinal dynamics controllers. As for adaptive, optimal and nonlinear control solutions at the high (guidance) control level, [55, 128, 160, 168] present path-following laws before the low-level control design.

2.3 Literature on Multi-UAV Formation Control

In the literature, various formation control architectures have been used. Regarding formation control schemes, there exist two main approaches: centralized and decentralized. The Centralized approach offers a global decision-making unit for multi-agent systems, and complexities may occur in mathematical analyses of large-scale systems. However, the Decentralized (distributed) approach provides a sub-decision making units for each agent in multi-agent systems. Hence, the distributed approach is more practical and easy to implement on real systems. In another categorization based on interaction topology of multi-agent systems, formation structures have been classified as hierarchical and non-hierarchical. For these structures, there are three main approaches used in the literature, namely, leader-follower, virtual-leader and behavioral-based [11, 15, 42]. The leader-follower approach works in a hierarchy and does not need sophisticated sensors in most cases. This approach depends on leader performance. The virtual-follower method uses a virtual leader for each agent to follow with non-hierarchy. This approach needs more complex communication capabilities; however, its performance is more robust than leader-follower. The behavioral-based approach needs predefined behaviors, such as formation-keeping and obstacle-avoidance. This approach mostly needs more complex computing capabilities. Moreover, symmetrical and asymmetrical information flows have been used in hierarchical and non-hierarchical patterns to ease the communication complexities.

A survey study [120] gives another overview of formation control for multi-agent systems based on sensing and interaction topology. So that, three categories are presented position-based, displacement-based, and distance-based controls. Position-based controls are usually not required interaction topologies, but they need long-range and more advanced sensors. Distance-based controls need interaction topology (rigid or persistence)
among members of multi-agent systems. The distance-based controls have sensing advantages since they only need inter-agent distances. In displacement-based controls, both sensing capabilities and interaction topologies of multi-agent systems are used equally. In the above categorizations, authors have been reviewed literature mostly depending on sensed variables, controlled variables, coordinate systems, and interaction topology.

In distance-based control of an asymmetric (directed), hierarchical formation, a rigid or persistent interaction (communication) graph is needed to achieve desired positions and then maintain desired formation shape by controlling inter-agent distances. Hence, algebraic graph theory is used to design required communication topologies in this kind of formation structures. For graph theory analyses, the literature mostly considers single-integrator agents and lacks local (individual) control performance effects on formations. Graph rigidity and formation stabilization are discussed with detail in the earlier studies [12, 121, 163]. In particular, [20, 53] discuss cohesive motion control tasks using the distributed approach and present general characteristics of cohesive motion tasks.

In some of the recent works, formation control designs are studied for multi quadrotor and fixed-wing UAV systems. [123] examines a distributed formation control design with the robust local controller utilized on multi-quadrotor helicopters. In [59], using a nonlinear dynamic model in the leader-follower structure, a consensus-based formation control is designed for a two-quadrotor system. Using directed topology in the formation graph, [150] also studies a consensus problem for a multi-quadrotor system under bounded disturbance. [60] deals with the impact of communication design over information flows of directed and undirected graphs for the consensus task of the multi-quadrotor systems. [20] present a distributed cohesive motion control for multiple fixed-wing UAV and quadrotor systems. The study introduces solutions for the maintenance of rigid and persistent motion during trajectory tracking tasks. In [61], the authors present a distributed formation control using more realistic lateral motion dynamics for surveillance tasks of the multi fixed-wing UAV system. Using range-based measurement, [165] study a rigidity maintenance control for a multi-UAV system. Using simplified quadrotor motion dynamics, [38, 137] study nonlinear control approaches combined with formation level. As N-vehicle cases, [81, 82] present single-integrator model based formation control designs for a group of the quadrotor UAVs in 2D. The authors do not consider the performance effects of local controllers of the quadrotor UAVs on formation maintenance and robustness, especially, in case of modeling issues on actual quadrotor UAVs such as uncertainties or disturbances.

As discussed above the studies, literature mostly focuses on high-level control and stability using simple models such as single or double integrator dynamics. However, literature lacks low-level control design and analysis to study and compensate the effect of realistic model issues such as uncertainties and disturbances on formation maintenance.
2.4 Single UAV Motion Control Architecture

In this section, single UAV motion control architectures of quadrotor and small fixed-wing UAV systems are discussed and presented as follows.

2.4.1 Quadrotor UAVs

Motion control architecture of quadrotor UAVs within navigation, guidance, and control (NGC) systems consists of path planner ($P$), control schemes ($H$ and $L$) and system sensors ($S$) as seen in Figure 2.3. The control system (CS) is considered in the two-level as high-level ($H$) and low-level ($L$). The separated dynamics (2.12)-(2.19) are located into the reference angle, altitude and attitude models to accommodate the proposed two-level control structure. In the CS, the on-line trajectory generator provides the desired positions $p_d(t)$. Reference angle model is responsible for generating the desired attitude angles $\varphi_d(t) = [\phi_d(t), \theta_d(t), \psi_d(t)]^T$. The reference angle generator and the attitude dynamics ($C_{at}$) provide lateral and attitude motion of the quadrotor UAV as a cascade control system. The altitude dynamics ($C_{al}$) controls the longitudinal motion. In the low-level, control laws generate control signals $u_z(t)$ and $u_{\varphi}(t)$. Then, the control signals are converted to actuator PWM motor inputs $v_r(t)$ by using the equation (2.5).
2.4.2 Fixed-wing UAVs

As studied in [23, 133], motion control architecture of small fixed-wing UAVs within NGC systems consists of five main parts: path planner ($P$), trajectory smoother ($D$), trajectory tracker ($T$), low-level autopilots ($A$) and sensors ($S$) as shown in Figure 2.4. In the navigation system (NS), hardware equipment and estimation algorithms are used to sense system states. In the guidance system (GS), trajectory smoother generates a more smooth desired trajectory using generated way-points and tracking errors. Since fixed-wing UAV motion dynamics is represented by the highly nonlinear model, the control system (CS) is considered in two control parts. At the first part, the trajectory tracker $T$, the simplified motion model (2.26) is used to design controllers. The separated dynamics (2.28)-(2.32) of the simplified model are located into lateral and altitude models as presented in Section 2.1.2 to accommodate the proposed two-level control structure. Then, trajectory tracker $T$ within the two-level control structure generates velocity, angular velocity and altitude commands to the autopilots. The second part is autopilot devices, and they are responsible for generating actual motor inputs of aileron, elevator, rudder, and throttle using control commands generated in the first control part.

Figure 2.4: Motion control architecture of small fixed-wing UAVs [133].
2.5 Multi-UAV Formation Control Architecture

This section presents a distributed formation control architecture of multi-UAV systems as shown in Figure 2.5 where high (formation) level control design is separated from low (individual) level motion control design via the proposed two-level control structure. In the two-level structure of the distributed formation control architecture, the high (formation) level controller \((H_i)\) of \(i\)th UAV is designed using a hierarchal leader-follower formation, and it consists of two submodules. Formation supervisor \((P_S)\) and desired attitude generator \((P_H)\) generate the desired position and attitude to guide \(i\)th UAV for formation maintenance. In the low-level \((L_i)\), proposed control laws are designed for \(i\)th individual UAV tracking control as well as maintaining desired formation shape via the guidance of the formation level module. Actual control inputs \(u_i(t)\) of the \(i\)th UAV system are generated in this level control.
Chapter 3

Adaptive Linear Quadratic Attitude Tracking Control with Sensor Fusion

3.1 Introduction

In the literature, various control approaches have been proposed for quadrotor UAV systems as discussed in Chapter 2. For attitude tracking control and stabilization, researches have developed solutions such as quaternion-based feedback control for exponential attitude stabilization [145], robust adaptive attitude tracking control [95], robust attitude control for uncertain quadrotors with proportional-derivative (PD) controller combined with a robust compensator [102], robust nonlinear design under uncertainties and delays [105], and fractional sliding modes based attitude control [78].

One of the main control interests for quadrotors UAV is optimization of time and energy (battery) consumption by designing optimal path planning and optimal tracking control. For such optimal attitude tracking, [30] has designed a linear quadratic regulation (LQR) based attitude stabilization. For solving a more general form of the same problem under wind gust disturbances, a switching model predictive attitude controller is developed in [5]. [156] presents $L_1$ optimal robust tracking control to compensate persistent disturbances in translational and rotational (attitude) dynamics.

Linear-quadratic (LQ) based optimal control frameworks constitute a systematic toolset for calculating ideal control gains with guaranteed system stability under LQ design conditions. LQ-based control schemes provide robust and precise steady-state tracking while the performance index (quadratic cost function) adjusts optimality trade-off between track-
ing/regulator performance and battery consumption. A particular LQ-based control approach is infinite-horizon optimal regulation based on linear time-invariant (LTI) models. This approach is widely used in real-time applications since its solution does not have computational complexities for obtaining constant state-feedback control (Kalman) gains by solving algebraic Riccati Equation (ARE). The infinite-horizon LQR has been mostly used in many earlier works as studied for the quadrotor UAV in [30] for attitude state regulation and stability.

On the other hand, Linear-quadratic tracking (LQT) problems have gained less attention compared to LQR problems, since time-varying reference trajectories lead to further analysis and computational complexities. LQT control schemes typically consist of two state-feedback and feed-forward terms. The state-feedback terms guarantee system stability by state-feedback (Kalman) gains which are calculated off-line solving differential Riccati equation (DRE). The feed-forward terms provide optimal tracking of time-varying bounded reference trajectories utilizing differential auxiliary vector signal equation. In practice, computational complexities arise because of the time variations in the feed-forward terms. Accordingly, the literature on LQT control design and applications on real-time systems is limited. [17] presents an off-line solution to the infinite-horizon LQT problem by solving the feed-forward term based on calculating the initial condition of the auxiliary vector signal. The authors present a real-time implementation of this solution on flexible beams system in [2]. Other than the classical solution, [113] presents an on-line reinforcement learning algorithm to solve LQT problem without requiring the knowledge of the system drift dynamics or the command generator dynamics.

Regarding LQT of quadrotor UAV systems, [161] presents a finite-horizon LQT control design with time-varying control gains which are calculated solving off-line discrete time matrix Riccati equations for the linearized full dynamics of the quadrotor UAV. Consideration of finite-horizon LQT with known boundary conditions at the initial and final time instants prevents the computational complexity issues with implementation of this design. However, in many practical cases, including the cases considered in this chapter, since the boundary conditions are unknown, infinite-horizon LQT needs to be considered for designing an alternative optimal linear tracking controller.

In this chapter, by the motivations of LQ-based optimal control advantages as stated above and lack of infinite-horizon LQT control schemes with their applications on real-time systems in literature, we present an infinite-horizon LQT control design, its practical solution and its experimental validation on the real-time quadrotor UAV with inertial parametric uncertainties and Inertial Measurement Unit (IMU) sensor noises. Furthermore, to improve robustness against parametric uncertainties, the presented LQT control design is combined with an adaptive parameter identification (PI) scheme based on least-squares
(LS) estimation. Combining the LQT control design and the PI scheme, an adaptive LQT (ALQT) control scheme is developed for optimal attitude tracking of quadrotor UAVs, with reduced tracking error and battery consumption.

Reliable attitude estimation is one of the main challenges for quadrotor UAV tracking control. Euler angles ($\phi$, $\theta$ and $\psi$) and Quaternions are two common types of attitude representation for UAV systems. IMUs, formed by 3-axis inertial sensors of gyroscopes, accelerometers and magnetometers, measures angular velocities, linear accelerations and the Earth’s magnetic field. Ideally, accelerometer measurements or numerical integration of angular velocities of gyroscopes should be enough for ideal sensors to determine attitude angles. However, in real-world conditions, individual usage of these sensors is not sufficient to determine attitude angles due to large amounts of system noise, drift errors and vibrations.

To obtain fast and accurate attitude states, sensor fusion techniques have been applied to IMU measurements, including wide ranges of complementary filters [14, 26, 65, 101, 109, 110, 153, 158] and Kalman filters [49, 52, 54, 65, 67, 87, 98, 101, 111, 129, 145, 151, 154, 155]. A complementary filter typically combines accelerometer output for low-frequency attitude estimation with integrated gyroscope output for high-frequency estimation. Complementary filters are computationally less demanding, and due to their simplicity and efficiency, these filters are still used for attitude estimation. A variety of complementary filters has been used to estimate attitude quaternions [109, 110, 158] or Euler angles for relatively small roll and pitch aerial vehicle angles [14, 65, 101, 145]. Complex rotations of simultaneous roll, pitch and yaw angles require nonlinear complementary filter fusion techniques [130].

Kalman filter is an optimal recursive estimation scheme that uses a system’s dynamic model, known control inputs, and multiple sequential measurements from sensors to form an estimate of the system states fusing prediction and measurement on-line [47, 52, 67, 129, 151]. The extended Kalman filter (EKF) is developed for nonlinear system state estimation and has been widely used for real-time UAV systems for Euler angle based attitude estimation [49, 54, 65, 101] as well as quaternion based attitude estimation [87, 98, 111, 154, 155]. Unscented Kalman filter (UKF) [29, 40, 43, 159] and adaptive Kalman filter (AKF) [100] are another widely used sensor fusion algorithms. In [49], a novel Kalman filter algorithm is proposed, which consists of an EKF and an inverse Φ-algorithm in a master-slave configuration to estimate reliable angular acceleration signals by fusing IMU sensor data. In [98], it is shown that even for applications with strong real-time constraints, EKF can properly estimate the UAV attitudes, even in the presence of data loss.

As discussed in Chapter 2 and the earlier work [90], we consider the quadrotor UAV con-
trol structure in two levels: high-level and low-level. High-level is mainly about guidance and position controlling in the autonomous motion tasks and generating the trajectories to be tracked by the low-level controller. Provided the trajectory from high-level, the low-level control is responsible for the quadrotor UAV’s attitude and altitude tracking performance and stability. In this chapter, we focus on the low-level control design, following a decentralized approach, considering the three motion dynamics modes separately: adaptive LQT control for the attitude dynamics, proportional (P) control for the yaw dynamics, and proportional-integral-derivative (PID) control for the altitude dynamics, as shown in Figure 3.1. In the overall structure, the attitude measurement noises, which come from IMU sensors, are compensated using a Kalman filter to obtain a more reliable attitude estimation. The effectiveness of the employed Kalman filter is investigated over the experiments that compare the Kalman filter results with a complementary filter. In the next step, we developed an infinite-horizon ALQT controller and validated its effectiveness by performing two set of experiments.

3.2 Quadrotor UAV Dynamics

A nonlinear dynamic model of quadrotor UAV motion dynamics (2.10) is presented in Chapter 2. In this chapter, we have simplified and partitioned this nonlinear dynamic model to obtain separate linear models for each of attitude, yaw, and altitude dynamics.

3.2.1 Attitude Model

Ignoring inertial and drag effects, we obtain a linearized attitude (roll/pitch) dynamics from (2.17)-(2.19) of the nonlinear dynamic model (2.10). Hence, we write the attitude model in the state-space form as

\[
\dot{x}_\varphi(t) = A_\varphi x_\varphi(t) + B_\varphi u_\varphi(t), \quad \varphi \in \{\vartheta, \phi\},
\]

\[
\varphi(t) = C_\varphi x_\varphi(t),
\]

where

\[
A_\varphi = \begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & \frac{2K}{J_\varphi} \\
0 & 0 & -b
\end{bmatrix}, \quad B_\varphi = \begin{bmatrix}
0 \\
0 \\
b
\end{bmatrix}, \quad C_\varphi = \begin{bmatrix}
1 \\
0 \\
0
\end{bmatrix}^T
\]

and

\[
x_\varphi = \begin{bmatrix}
\varphi \\
\dot{\varphi} \\
x_\varphi, u_\varphi, \varphi, \dot{\varphi}, T_\varphi, J_\varphi
\end{bmatrix}
\]

\[
K, b \text{ and } l \text{ represent states, control inputs, Euler angles, angular velocities, thrust forces, rotational inertias, positive armature gain, the actuator bandwidth in attitude (roll/pitch) dynamics and the distance between the center of gravity } O_b \text{ and each propeller, respectively.}
\]
Remark 3.2.1. Attitude ($\varphi$) dynamics represents roll ($\vartheta$) and pitch ($\phi$) dynamics, and yaw ($\psi$) is separated from attitude dynamics for the proposed control design.

3.2.2 Yaw Model

We obtain linearized yaw dynamic as

$$\ddot{\psi} = \frac{4K_{\psi}K}{J_{\psi}} \frac{b}{(s + b)} u_{\psi},$$

where $u_{\psi}$ is the yaw control input, $K_{\psi}$ is thrust-to-moment gain and $J_{\psi}$ is the rotational inertia in yaw motion. Finally, we write the linearized yaw model in form of an input-output transfer function as:

$$\psi = \frac{4K_{\psi}Kb}{s^2(s + b)J_{\psi}} u_{\psi}.$$  

(3.3)

3.2.3 Altitude Model

We have linearized the nonlinear altitude model (2.14)-(2.16) by the use of small angle approximation and taking the effect of gravity as an offset in the linearized model. Accordingly, we obtain the simplified linear altitude model as

$$\ddot{p}_z = \frac{4K}{m} \frac{b}{(s + b)} u_z,$$

where $p_z$ is z-position of $O_b$, $u_z$ is the altitude control input and $m$ is the total mass of the quadrotor UAV system. Finally, we obtain the linearized altitude model in the form of an input-output transfer function as

$$p_z = \frac{4Kb}{s^2(s + b)m} u_z.$$  

(3.5)

3.3 Problem Statement

Considering a quadrotor UAV with attitude (roll/pitch) dynamics (3.1), yaw dynamics (3.3), and altitude dynamics (3.5), as illustrated in Figure 3.1, the objectives of this chapter are threefold:
1. Given the IMU sensor measurements of the attitude angles, design a data fusion algorithm based on (i) Kalman filtering and, for comparative analysis purposes, (ii) complementary filtering, to cancel the IMU sensor noise effects and produce accurate attitude state estimates.

2. Design the control units to generate the command signals $u_z, u_\psi, u_\theta, u_\phi$ for feeding the PWM generator that generates the motor control input signal $v_r$, per the diagram in Figure 3.1: (a) Design an infinite-horizon ALQT controller to generate the optimal attitude control signal $u_\phi(t) = u_\phi^*(t)$ so that $\varphi(t)$ tracks its desired trajectory $\varphi_d(t)$, minimizing the predefined quadratic performance optimal tracking and energy consumption cost function

$$J = \frac{1}{2} \int_0^\infty (Qe_\varphi^2(t) + Ru_\varphi^2(t)) dt,$$

(3.6)
where $Q$ and $R$ are positive constant weighting terms and

$$e_{\varphi}(t) = \varphi(t) - \varphi_d(t)$$

is the attitude tracking error. (b) Design a P yaw controller to generate $u_\psi(t)$ and a PID altitude controller to generate $u_z(t)$.

3. Combining the designs in 1 and 2, above, real-time implement and experimentally validate the overall control scheme

### 3.4 Control Approach

In our infinite-horizon ALQT control design, the optimal control law consists of two terms: the state-feedback and the feed-forward. The state-feedback term maintains the stability of the attitude dynamics. This term is obtained solving an algebraic Riccati equation (ARE). The feed-forward term depends on the desired trajectory and is used for establishing trajectory tracking performances. The above optimal control law is combined with an LS based adaptive PI algorithm to make it robust, adaptive and avoid inertial uncertainties in the attitude dynamics. After this combination, because of the uncertainties, the ARE needs to be solved on-line as well. In the implementation, by comparing the on-line estimates of the uncertain parameters with some critical parameters calculated and stored in a look-up table, the time-varying state-feedback (from the PI) and then the time-varying feed-forward (from slowly-varying desired attitude and the PI) terms are calculated on-line. In the real-time implementation of the designed ALQT scheme, we utilize a practical real-time computation technique based on parameterized analytical solutions of the state-feedback and the feed-forward terms.

### 3.5 IMU Sensor Data Fusion

The quadrotor UAV needs a robust estimation scheme for denoising the attitude angle measurements to provide reliable feedbacks to the proposed ALQT control scheme. The attitude angles are measured using an ADIS16405 IMU as shown in Figure 3.2. Then a Kalman filter is employed to attenuate the effect of measurement noises. The IMU contains a 3-axis gyroscope to measure angular velocities ($\dot{\theta}, \dot{\phi}, \dot{\psi}$), a 3-axis accelerometer to measure accelerations due to Earth’s gravity ($a_x, a_y, a_z$) and a 3-axis magnetometer to measure the magnetic field intensities ($m_x, m_y, m_z$). The specifications are listed in Table 3.1.
Table 3.1: The ADIS16405 IMU Specifications [171].

<table>
<thead>
<tr>
<th></th>
<th>Gyroscope</th>
<th>Accelerometer</th>
<th>Magnetometer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>±305(deg/s)</td>
<td>±18(g)</td>
<td>±3.5(gauss)</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>0.05(deg/s/LSB)</td>
<td>3.33(mg/LSB)</td>
<td>0.5(mgauss/LSB)</td>
</tr>
</tbody>
</table>

Figure 3.2: The ADIS16405 IMU Module on the Qball-X4 quadrotor UAV.

3.5.1 Attitude Determination from IMU Sensors

Roll and pitch angles are obtained based on accelerometer and gravity vector relation. The rotation matrix from the body frame to the inertial frame is defined with the Euler angles as:

\[
R_{b2i} = \begin{bmatrix}
\cos \psi \cos \vartheta & -\sin \psi \cos \phi + \cos \psi \sin \vartheta \sin \phi & \sin \psi \sin \phi + \cos \psi \sin \vartheta \cos \phi \\
\sin \psi \cos \vartheta & \cos \psi \cos \phi + \sin \psi \sin \vartheta \sin \phi & -\cos \psi \sin \phi + \sin \psi \sin \vartheta \cos \phi \\
-\sin \vartheta & \cos \vartheta \sin \phi & \cos \vartheta \cos \phi
\end{bmatrix}.
\] (3.8)

Assuming constant translational velocities [49, 152], i.e., ignoring translational accelerations, we obtain the following relation between the accelerometer output, rotation matrix...
and earth gravity:

\[
\begin{bmatrix}
    a_x \\
    a_y \\
    a_z
\end{bmatrix} = R_{i2b} \begin{bmatrix}
    0 \\
    0 \\
    g
\end{bmatrix} \begin{bmatrix}
    -\sin \vartheta \\
    \cos \vartheta \sin \phi \\
    \cos \vartheta \cos \phi
\end{bmatrix} g,
\]

(3.9)

where \( R_{i2b} = R_{b2i}^T \). From (3.9), attitude angles are calculated as

\[
\varphi_{\text{acc}} = \begin{bmatrix}
    \theta_{\text{acc}} \\
    \phi_{\text{acc}}
\end{bmatrix} = \begin{bmatrix}
    \text{atan2}(-a_x, \sqrt{a_y^2 + a_z^2}) \\
    \text{atan2}(a_y, a_z)
\end{bmatrix},
\]

(3.10)

where \( \text{atan2}(a_y, a_z) \) denotes arc tangent of \( a_y \) and \( a_z \) while it uses the signs of both arguments to determine the quadrant of the result. By determining the roll and pitch angles, the rotation matrix from the body frame to the magnetometer local (NED:North-East-Down) frame is rearranged as

\[
\begin{bmatrix}
    m_x \\
    m_y \\
    m_z
\end{bmatrix} = \begin{bmatrix}
    \cos \vartheta & \sin \vartheta \sin \phi & \sin \vartheta \cos \phi \\
    0 & \cos \phi & -\sin \phi \\
    -\sin \vartheta & \cos \vartheta \sin \phi & \cos \vartheta \cos \phi
\end{bmatrix} \begin{bmatrix}
    m_{xb} \\
    m_{yb} \\
    m_{zb}
\end{bmatrix}.
\]

(3.11)

Hence, yaw (heading) is calculated as

\[
\psi_c = \text{atan2}(m_y, m_x).
\]

(3.12)

In practice, the yaw (heading) is updated by gyroscope data integration instead of a Kalman filter or a complementary filter since the laboratory environment has magnetic (metallic) disturbances on the heading calculation (3.12). Solution methods of magnetic disturbances on heading calculation are discussed with the details in [50].

### 3.5.2 Attitude Estimation Using Kalman Filter

To filter IMU accelerometer noises, a linear Kalman filter is employed in this chapter. At each time step \( k \), this Kalman filter first predicts the state propagation using the dynamic model of the quadrotor UAV, the control inputs applied at step \( k - 1 \) and the state measurement at step \( k - 1 \). Then, it incorporates new measurement data of step \( k \), to determine the state estimate.
Consider the following discrete-time linear time-invariant model of the attitude dynamics, with additive Gaussian measurement noise and disturbance, based on (3.1):

\[
\begin{align*}
x[k+1] &= A_d x[k] + B_d u[k] + w, \\
y[k] &= C_d x[k] + v,
\end{align*}
\]  
(3.13)  
(3.14)

where \( w \) is zero mean Gaussian disturbance noise with covariance \( Q_K \), \( v \) is zero mean Gaussian measurement noise with covariance \( R_K \), and

\[
A_d = \begin{bmatrix}
1 & T_s & 0 \\
0 & 1 & 2bK T_s \\
0 & 0 & 1 - bT_s
\end{bmatrix}, \quad B_d = \begin{bmatrix}
0 \\
0 \\
bT_s
\end{bmatrix}, \quad C_d = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix},
\]  
(3.15)

with sampling time \( T_s \). Note that in implementation of (3.15), since the value of the rotational inertia \( J_\phi \) is uncertain, the nominal value of this parameter is used, as detailed in Remark 3.6.1 in Section 3.6.1. For this system model, the Kalman filter prediction and update equations are as follows.

Prediction:

\[
\begin{align*}
\hat{x}[k+1|k] &= A_d x[k|k] + B_d u[k], \\
P[k+1|k] &= A_d P[k|k] A_d^T + Q_K,
\end{align*}
\]  
(3.16)  
(3.17)

Update:

\[
\begin{align*}
\bar{y}[k] &= y[k] - C_d \hat{x}[k|k-1], \\
M[k] &= P[k|k-1] C_d^T (C_d P[k|k-1] C_d^T + R_K)^{-1}, \\
\hat{x}[k] &= \hat{x}[k|k-1] + M[k] \bar{y}[k], \\
P[k|k] &= (I - M[k] C_d) P[k|k-1].
\end{align*}
\]  
(3.18)  
(3.19)  
(3.20)  
(3.21)

where \( P[k+1|k] \) and \( M[k] \) are the predicted error covariance and the optimal Kalman gain, respectively.

Remark 3.5.1. Measurement covariance matrix is specified by calculating the variance of the sensor noise from off-line sample measurement of IMU sensors. Disturbance covariance matrix is tuned by trial and error method via experimental tests. The states are initialized at zero and the sampling time is selected same as used in overall experiment design.
3.5.3 Attitude Estimation by Complementary Filter

As an alternative to Kalman filtering, we also study the utilization of complementary filter in denoising and fusion of measurement data from accelerometers and gyroscopes. Typically, an accelerometer based orientation estimation works better in static conditions, and on the other hand a gyroscope based orientation estimation gives better results in dynamic conditions. A complementary filter passes the accelerometer signals through a low-pass filter and the gyroscope signals integral through a high-pass filter. Then, the resulting signals are summed up to estimate the attitude angles more reliably in both dynamic and static condition cases. The schematic complementary filter block diagram is depicted in Figure 3.3.

3.6 Adaptive Optimal Attitude Tracking Control Design

In this section, the proposed ALQT control scheme for attitude tracking of a quadrotor UAV is presented.

3.6.1 Adaptive Parameter Identification Scheme

We employ an LS based PI scheme to estimate the uncertain inertial parameters. From the attitude dynamics equation (3.1), following the procedure in [74, 89], we first define a linear parametric model avoiding the need for signal differentiation and the associated
noise sensitivity issue by use of the stable filter \( \frac{1}{(s+\lambda)} \), \( \lambda > 0 \), as follows:

\[
 z_\varphi = \theta_\varphi^* \Phi_\varphi, \\
 z_\varphi = \frac{s}{(s+\lambda)} \dot{\varphi}, \theta_\varphi^* = \frac{1}{J_\varphi}, \Phi_\varphi = \frac{2lKb}{(s+\lambda)(s+b)} u_\varphi,
\]

noting that the Euler rate \( \dot{\varphi} \) (obtained using the IMU and the filters in Section 3.5) and the control signal \( u_\varphi \) are measurable, and \( l, K, b \) are known constant parameters.

**Assumption 3.6.1.** The upper and lower limits of \( \theta_\varphi^*(t) \) are known, i.e. \( 0 < \theta_\varphi < \theta_\varphi^*(t) < \bar{\theta}_\varphi \) for some known \( \theta_\varphi, \bar{\theta}_\varphi > 0 \).

**Remark 3.6.1.** For the setup used in this chapter, the limits of \( \hat{\theta}_\varphi(t) \) are taken \( 10 \leq \hat{\theta}_\varphi(t) \leq 49 \). Accordingly, the nominal value of \( J_\varphi \) is calculated as \( J_\varphi^0 = \frac{2}{\theta_\varphi^0} \approx 0.03 \).

To generate the estimate \( \hat{\theta}_\varphi \) of the uncertain inertia parameter \( \theta_\varphi^* \), we apply the following recursive LS algorithm [74] based on the parametric model (3.22):

\[
 \hat{\theta}_\varphi(t) = Pr(P(t)\epsilon(t)\Phi_\varphi(t)) = \begin{cases} 
 P(t)\epsilon(t)\Phi_\varphi(t), & \text{if } \theta_\varphi < \hat{\theta}_\varphi < \bar{\theta}_\varphi, \\
 0, & \text{otherwise} 
\end{cases}, \\
 \hat{\theta}_\varphi(0) = \hat{\theta}_\varphi^0, \\
 \hat{P}(t) = \begin{cases} 
 \beta P(t) - \frac{\Phi_\varphi^2(t)}{m_n^2(t)} P^2(t), & \text{if } \theta_\varphi < \hat{\theta}_\varphi < \bar{\theta}_\varphi, \\
 0, & \text{otherwise} 
\end{cases}, \\
 \epsilon(t) = \frac{z_\varphi(t) - \hat{\theta}_\varphi(t)\Phi_\varphi(t)}{m_n^2(t)}, \\
 m_n^2(t) = 1 + \alpha_n \Phi_\varphi^2(t), \quad 1 \gg \alpha_n > 0,
\]

where \( P(t) \) is the positive covariance (time varying gain) term with \( P(0) = P_0 > 0 \), \( m_n \) is the normalizing signal, and \( \epsilon \) is the estimation error. \( Pr(. \) is a projection operator which maintains \( \hat{\theta}_\varphi \in [\theta_\varphi, \bar{\theta}_\varphi] \).

**Lemma 3.6.2** (Stability and Convergence). Consider the LS based PI scheme (3.23), applied to the attitude dynamics (3.1). It is guaranteed that all the signals in (3.23), including \( P \) and \( P^{-1} \), are bounded and \( \hat{\theta}_\varphi \in [\theta_\varphi, \bar{\theta}_\varphi] \). Further, if \( \Phi_\varphi^2(t) = \frac{\Phi_\varphi(t)}{m_n(t)} \) is persistently exciting, i.e. \( \frac{1}{T} \int_{t_0}^{t_0+T} \Phi_\varphi^2 \Phi_\varphi d\tau \geq \alpha_0 \) for all \( t \geq 0 \) and some \( T, \alpha_0 > 0 \), then (3.23) ensures that \( \theta_\varphi(t) \to \theta_\varphi^* \) as \( t \to \infty \). The convergence of \( \theta_\varphi(t) \to \theta_\varphi^* \) is exponential for \( \beta > 0 \).

**Proof.** The result is a direct corollary of the more general Theorem 3.7.4 and 3.10.1 in [74].
### 3.6.2 Generic Linear Quadratic Tracking Control Design

To construct the base optimal control law of the proposed ALQT scheme, we follow an infinite-horizon LQT control design approach [116], explained in the sequel for a linear system in the generic state-space form

\[
\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t),
\]

where \( x \in \mathbb{R}^n \), \( u \in \mathbb{R}^r \) and \( y \in \mathbb{R}^m \) are state, control input and output vectors. \( A \in \mathbb{R}^{n \times n} \), \( B \in \mathbb{R}^{n \times r} \) and \( C \in \mathbb{R}^{m \times n} \) are state, input and output matrices. \( m \), \( n \), and \( r \) are generic system dimensions. The objective is to generate \( u(t) \) so that \( y(t) \) tracks a given desired continuous and differentiable output trajectory \( z(t) \in \mathbb{R}^m \) as close as possible with minimum consumption of control effort for all \( t \). Thus, let us define the error vector

\[
e(t) = z(t) - y(t),
\]

and the cost function

\[
J = \frac{1}{2} \int_0^\infty (e^T(t)Qe(t) + u^T(t)Ru(t))dt,
\]

where \( Q \in \mathbb{R}^{m \times m} \) and \( R \in \mathbb{R}^{r \times r} \) are symmetric, positive definite weighting matrices.

In order to generate the optimal control signal \( u(t) = u^*(t) \) that minimizes the cost function (3.26), following Hamiltonian calculation [116], at first, the following DRE is formed:

\[
\dot{P} = -PA - A^TP + PBR^{-1}B^TP - C^TQC,
\]

where \( P \in \mathbb{R}^{n \times n} \) is a symmetric, positive definite matrix. Since the infinite-horizon LQT design [116] is studied, there is no terminal \( F(t_f) = 0 \) in cost function (3.26). Therefore, \( P(t) \) tends to its steady-state value \( \lim_{t_f \to \infty} P(t_f) = \bar{P} \) as the solution of the following ARE:

\[
-\bar{P}A - A^T\bar{P} + \bar{P}BR^{-1}B^T\bar{P} - C^TQC = 0,
\]

where \( \bar{P} \in \mathbb{R}^{n \times n} \) is a symmetric, positive definite matrix calculated by the analytical solution of the ARE (3.28).

Then, as the next step in the LQT design steps with Hamiltonian approach a vector signal \( g(t) \in \mathbb{R}^n \) is generated via the differential equation

\[
\dot{g}(t) = -[A^T - PBR^{-1}B^T]g(t) - C^TQz(t).
\]
The final optimal control signal is generated as

\[ u^*(t) = -R^{-1}B^T \bar{P}x(t) + R^{-1}B^T g(t), \]  

(3.30)

where \(-R^{-1}B^T \bar{P}x(t)\) is the state feedback term and \(R^{-1}B^T g(t)\) is the feedforward term. Note that the control law (3.30) is established in [116] to bear the following optimal tracking property:

**Proposition 3.6.3.** [116]: The control law (3.30) guarantees that (3.24) is closed loop stable and (3.26) is minimized, for any given slowly-varying desired output trajectory \(z(t)\).

To simplify and ease the calculation of the vector signal \(g(t)\), we use an approximation [10] as follows:

**Approximate vector signal** \(\bar{g}(t)\): It is established in [10], if \(z(\cdot)\) is slowly varying then \(\dot{g}(t)\) in (3.29) can be approximated as \(\dot{g}(t) \approx 0\) leading to the approximate solution

\[ g(t) \approx \bar{g}(t) = [A^T - \bar{P}BR^{-1}B^T]^{-1}[-C^T Qz(t)]. \]  

(3.31)

### 3.6.3 Adaptive Linear Quadratic Tracking (ALQT) Control Design

For attitude control, our approach is to apply the control law (3.28), (3.30), (3.31) to the system (3.1). Note that the implementation of the control law (3.30) requires \(\bar{P}\) from (3.28) and \(\bar{g}(t)\) from (3.31), and hence requires knowledge of the system matrices \(A, B, C\). In our case, in (3.1), although \(B_\phi, C_\phi\) are known, \(A_\phi\) is unknown. Hence, following the certainty equivalence approach [74, 116], the following adaptive version of the LQT control law (3.28), (3.30), (3.31) for the cost function (3.6) and the attitude tracking error (3.7) is designed.

The time-varying adaptive ARE, the approximate vector signal \(\bar{g}(t)\) and the adaptive optimal control signal are obtained, respectively as

\[ -\bar{P} \dot{A}_\phi(t) - \dot{A}_\phi^T(t) \bar{P} + \bar{P}B_\phi R^{-1}B_\phi^T \bar{P} - C_\phi^T QC_\phi = 0, \]  

(3.32)

\[ \dot{\bar{g}}(t) = [\dot{\bar{A}}_\phi^T(t) - \bar{P}B_\phi R^{-1}B_\phi^T]^{-1}[-C_\phi^T Qz(t)], \]  

(3.33)

\[ \hat{u}_\phi^*(t) = -R^{-1}B_\phi^T \bar{P}x_\phi(t) + R^{-1}B_\phi^T \bar{g}(t), \]  

(3.34)
where \( \hat{A}_\varphi(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 2lK\hat{\varphi}(t) \\ 0 & 0 & -b \end{bmatrix} \), \( B_\varphi = \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix} \), \( C_\varphi = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \). Solving (3.32) for \( \bar{P} = \begin{bmatrix} \bar{P}_1 & \bar{P}_2 & \bar{P}_3 \\ \bar{P}_2 & \bar{P}_4 & \bar{P}_5 \\ \bar{P}_3 & \bar{P}_5 & \bar{P}_6 \end{bmatrix} \in \mathbb{R}^{3\times3} \), we obtain

\[
0 = -\left(\bar{P}_3^2b^2/R\right) + Q, \\
0 = -\bar{P}_1 + (\bar{P}_3\bar{P}_5b^2)/R), \\
0 = -(2lK\hat{\varphi}\bar{P}_2) + \bar{P}_3b + (\bar{P}_3\bar{P}_5b^2)/R), \\
0 = -2\bar{P}_2 + (\bar{P}_5^2b^2/R), \\
0 = -(2lK\hat{\varphi}\bar{P}_4) + \bar{P}_5b - \bar{P}_3 + (\bar{P}_5\bar{P}_5b^2)/R), \\
0 = -(4lK\hat{\varphi}\bar{P}_5) - 2\bar{P}_6b + (\bar{P}_6^2b^2/R) .
\]

Solving (3.33) for \( \bar{g}(t) = [\bar{g}_1(t) \; \bar{g}_2(t) \; \bar{g}_3(t)]^T \in \mathbb{R}^3 \), we obtain

\[
\bar{g}_1(t) = \left[(\bar{P}_3Q)/(\bar{P}_3R)\right]\varphi_d(t), \\
\bar{g}_2(t) = \left[(\bar{P}_5bQ + RQ)/(2lK\hat{\varphi}\bar{P}_5)\right]\varphi_d(t), \\
\bar{g}_3(t) = \left[(RQ)/(\bar{P}_6b^2)\right]\varphi_d(t).
\]

**Remark 3.6.4.** There is also no constraint on the control signal \( \hat{u}_\varphi(t) \) considered in the control design. However, we consider a limit \(-0.025 \leq u_\varphi(t) \leq 0.025 \) to prevent damages on the quadrotor motors due to high torque commands.

### 3.7 Yaw and Altitude Control

To provide the overall motion in experiments, P and PID controllers are designed respectively, for yaw and altitude dynamics as follows.

#### 3.7.1 Yaw Control

Since yaw dynamics is not directly affecting the lateral motion of the quadrotor UAV system, the yaw motion control is considered independently. Therefore, the following P
control law is used based on the dynamic model (3.3):

\[ u_\psi = K_p \psi e_\psi, \quad (3.37) \]

where \( e_\psi = (\psi_d - \psi) \).

### 3.7.2 Altitude Control

Altitude controller is derived for keeping the quadrotor UAV system in its desired altitude and providing stability at the longitudinal motion. The following PID control law is used based on the dynamic model (3.5):

\[ u_z = K_p z (e_z) + K_i \int_0^t (e_z) dt + K_d \dot{e}_z, \quad (3.38) \]

where \( e_z = (p_{zd} - p_z) \).

**Remark 3.7.1.** With the attitude, yaw and altitude control schemes as designed above, the control inputs \( \hat{u}_\psi \), \( u_\psi \) and \( u_z \) are generated. Then, we combine these inputs [90] to generate each motor PWM inputs \( v_r \), for flight control of the Qball-X4 quadrotor UAV system.

### 3.8 Experimental Tests and Comparative Simulations

#### 3.8.1 Test Platform

The test platform consists of a Qball-X4 quadrotor UAV and a ground control and communication station (host computer) as illustrated in Figure 3.4. The Qball-X4 is developed by Quanser Inc. and equipped with a sonar sensor and an IMU to provide altitude, acceleration, angular rate and magnetometer measurements [172]. It has an on-board avionics data acquisition card (DAQ) and Gumstix embedded computer for interfacing with on-board sensors and driving the four rotor motors. Each motor is linked to one of the four PWM servo output channels on the DAQ. The Qball-X4 dynamic parameters as specified in [172] are presented in Table 3.2. The ground station computer is used for coding the designed control algorithms, and embedding on the Qball-X4 on-board computers before tests as well as generating the high-level control inputs in the form of desired attitude and altitude trajectories on-line during the tests. For control algorithm coding and embedding, Quarc, a MATLAB/Simulink® based interface software developed by Quanser Inc., is used.
Table 3.2: The Qball-X4 quadrotor UAV dynamic parameters [172].

<table>
<thead>
<tr>
<th>m (kg)</th>
<th>l (m)</th>
<th>K (N)</th>
<th>K_ψ (Nm)</th>
<th>b (rad/sec)</th>
<th>J_φ0 (kgm^2)</th>
<th>J_ψ (kgm^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4</td>
<td>0.2</td>
<td>120</td>
<td>4</td>
<td>15</td>
<td>0.03</td>
<td>0.04</td>
</tr>
</tbody>
</table>

Figure 3.4: The Qball-X4 quadrotor UAV test platform.

3.8.2 Control Design Specifications and On-line Calculation of Control Parameters

In the implementation of the ALQT control design explained in detail in Section 3.6.3, the error and the control weighting parameters are chosen as \( Q = 100 \) and \( R = 30000 \). Following (3.35a), the constant entry \( \bar{P}_3(t) \) is calculated as \( \bar{P}_3 = \sqrt{(QR)} / b = 115.4701 \). The other entries of \( \bar{P}(t) \) are calculated solving (3.35) online, noting the dependence of these entries to each other and the parameter estimate \( \hat{\theta}_\varphi \). From (3.35d), the entry \( \bar{P}_2 \) is found in the form of the entries \( \bar{P}_3 \) and written in (3.35c). Then, the equations (3.35c) and (3.35f) are obtained in the form of the entries \( \bar{P}_5 \) and \( \bar{P}_6 \) as follows:

\[
0 = -(lK \hat{\theta}_\varphi \bar{P}_5^2 b^2 / R) + \bar{P}_3 b + (\bar{P}_3 \bar{P}_6 b^2 / R), \tag{3.39a}
\]

\[
0 = -(4lK \hat{\theta}_\varphi \bar{P}_5) - 2\bar{P}_6 b + (\bar{P}_6^2 b^2 / R). \tag{3.39b}
\]
Figure 3.5: Off-line calculation of $\bar{P}_6$ for the estimate $\hat{\varphi} \in [\varphi, \bar{\varphi}]$.

Solving the equations (3.39a) and (3.39b) by using Maple® and MATLAB® softwares, $P_6$, which is chosen as a critical parameter, is off-line calculated for the estimate $\hat{\varphi} \in [\varphi, \bar{\varphi}]$ of $\theta^*$. Then, a lookup table is prepared as plotted in Figure 3.5. The remaining entries of $\bar{P}$ are simultaneously calculated using $\bar{P}_6$ and the estimate $\hat{\varphi}$ as follows:

\begin{align*}
\bar{P}_5 &= ((2\bar{P}_6b) + ((\bar{P}_6^2b^2)/R))/(4lK\hat{\varphi}), \\
\bar{P}_2 &= (\bar{P}_5^2b^2)/(2R), \\
\bar{P}_1 &= (\bar{P}_3\bar{P}_5b^2)/R, \\
\bar{P}_4 &= ((-\bar{P}_3)/2lK\hat{\varphi}) + ((\bar{P}_5b)/2lK\hat{\varphi}) + ((\bar{P}_5\bar{P}_6b^2)/2lK\hat{\varphi}R).
\end{align*}

After obtaining $\bar{P}$, by (3.36) and the reference input $\varphi_d(t)$, the vector signal $\bar{g}(t)$ is found at each time instant, as well. For example, for the nominal value $\hat{\varphi}_0 = 33 \ [1/kgm^2]$, the Riccati coefficient matrix $\bar{P}$ is obtained using (3.39a), (3.39b), (3.40a), (3.40b), (3.40c), (3.40d) as follows:

\begin{align*}
\bar{P} = \begin{bmatrix}
20.0886934 & 2.0177780 & 115.4700538 \\
2.01777802 & 480969.75 & 23.19642512 \\
115.470053 & 23.196425 & 1727.886987
\end{bmatrix}.
\end{align*}

44
After that, the vector signal $\vec{g}(t)$ is found by (3.41) and (3.36) as follows:

$$\vec{g}(t) = \begin{bmatrix}
20.08869343750145 \varphi_d(t) \\
2.017798199381859 \varphi_d(t) \\
115.4700538758503 \varphi_d(t)
\end{bmatrix}. \tag{3.42}$$

The ALQT control design with specifications above is used for pitch and roll control. For yaw tracking a P controller is used with gain $K_{p\psi} = 0.015$, and for altitude tracking a PID controller is used with gains $K_{pz} = 0.006$, $K_{iz} = 0.008$ and $K_{dz} = 0.002$.

In implementation of the adaptive PI scheme (3.23), the forgetting factor, the initial covariance, and the initial parameter estimate, are selected as, respectively, $\beta = 0.001$, $P_0 = 10^5$ and $\theta_0 = 10 [1/kgm^2]$. In the Kalman filter implementation, $Q_K$ and $R_K$ matrices are taken as $Q_K = 10^{-3}I_3$ and $R_K = 2 \times 10^{-4}I_2$. $A_d$, $B_d$ and $C_d$ matrices are numerically obtained for the nominal value of $J_{\varphi 0} = 0.03 \text{ [kgm}^2] \text{]}$ and the sampling time $T_s = 0.005$ as

$$A_d = \begin{bmatrix}
1 & 0.005 & 0 \\
0 & 1 & 8 \\
0 & 0 & 0.925
\end{bmatrix}, B_d = \begin{bmatrix}
0 \\
0 \\
0.075
\end{bmatrix}, C_d = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix}.$$ For the complementary filter, $\varphi_{acc}$ is passed through a low pass filter with transfer function $G_a(s) = (20s + 1)/(100s^2 + 20s + 1)$, and $\dot{\varphi}$ is passed through a high pass filter with transfer function $G_g(s) = (100s)/(100s^2 + 20s + 1)$.

### 3.8.3 Experimental Results

After setting all control parameters with the sampling rate 200 [Hz] by using MATLAB/Simulink® and Quarc interface in the host computer, as explained in Sections 3.8.1 and 3.8.2, we have implemented the proposed control scheme on the Qball-X4 as seen in Figure 3.6 for the following two cases:

- **Test 1.** ALQT with complementary filter: The video of the experiment is presented in URL: [https://www.youtube.com/watch?v=OVZ_zg4SS0Y](https://www.youtube.com/watch?v=OVZ_zg4SS0Y).
- **Test 2.** ALQT with Kalman filter: The video of the experiment is presented in URL: [https://www.youtube.com/watch?v=wsKiPJjj68](https://www.youtube.com/watch?v=wsKiPJjj68).

In both tests, the Qball-X4 starts to perform the tracking control task after hovering for 15 seconds. The real-time IMU data measurements in Test-2 from the gyroscope, the accelerometer and the magnetometer are presented in Figures 3.7, 3.8 and 3.9, respectively.
Applying the methodology explained in Section 3.5.1, the IMU measurements shown in Figure 6 are used to obtain the raw calculation of the roll and pitch parameters (yellow plots), and then to generate the estimates by complementary filter (blue plots) and Kalman filter (red plots) shown in Figures 3.11 and 3.12. Kalman filter provides more reliable data less sensitive to noise. For the yaw estimation, gyroscope data integration is used instead, due to distortion effects by metallic objects of the test environment, as explained in Section 3.5.1.

The tracking error performances of both tests verify that the control objective is satisfied as seen in Figure 3.13 and 3.14. In both tests, the controllers maintain attitude angles close to their desired angles with small attitude tracking errors ±0.1 [rad]. However, as seen in 3.14, ALQT control with Kalman filter is more robust to sensor noises and uncertainties, and results in smaller tracking errors.

As seen in Tables 3.3 and 3.4, ALQT control with Kalman filter gives significantly smaller mean-square error and consumes less battery (energy). It is also observed in additional simulations that the proposed controller consumes less battery energy with more robust control action compared to other classical controllers such as PID.

In real-time, the motor PWM control inputs have the constraint $-0.1 \leq v_r(t) \leq 0.1$ since they work with limited voltage to prevent damages due to high torque commands.
Table 3.3: Mean square error of $e_\phi$.

<table>
<thead>
<tr>
<th>ALQT</th>
<th>Roll [rad]</th>
<th>Pitch [rad]</th>
</tr>
</thead>
<tbody>
<tr>
<td>with Kalman filter</td>
<td>0.0012</td>
<td>0.0012</td>
</tr>
<tr>
<td>with comp. filter</td>
<td>0.0027</td>
<td>0.0029</td>
</tr>
</tbody>
</table>

Table 3.4: Average battery consumption by $\hat{u}^*_\phi$.

<table>
<thead>
<tr>
<th>ALQT</th>
<th>Roll [voltage/sec]</th>
<th>Pitch [voltage/sec]</th>
</tr>
</thead>
<tbody>
<tr>
<td>with Kalman filter</td>
<td>0.00071</td>
<td>0.00066</td>
</tr>
<tr>
<td>with comp. filter</td>
<td>0.00086</td>
<td>0.00140</td>
</tr>
</tbody>
</table>

Hence, a limit is applied for the optimal attitude control inputs as mentioned in Remark 3.6.4 even though LQT design procedure does not have any constraints. As seen in Figures 3.15, 3.16 and 3.17, the proposed controller satisfies admissible and optimal control actions for all $t > 0$ during the tests. Figures 3.15, 3.16 and 3.17 show that the motor PWM and the optimal attitude control inputs are kept within the allowed limits.

The LS based estimation of the uncertain inertia parameters $\hat{\theta}_\theta$ and $\hat{\theta}_\phi$ is presented in Figure 3.10. The estimates, which are purposely initialized at values away from the nominal values (to test the expected convergence), successfully converge to the vicinity of the nominal value $33 \ [1/kgm^2]$ in around $40 \ [sec]$. The convergence rate of the estimation can be adjusted easily adjusting the design parameters of the LS based adaptive law.

### 3.8.4 Comparative Simulations and Observations

For optimal performance comparison with the existing literature, in ideal simulation conditions without noises, we simulate the ALQT control design and compare with a classical PID controller with control gains $K_p = 0.0017; \ K_i = 8.98; \ K_d = 0.005$. As seen in Figure 3.18 and 3.19, the ALQT controller gives smaller tracking errors with less control action. Therefore, in the actual settings with noises, it is expected that the ALQT with a reliable filter will give us better tracking and control input performances compared to a PID controller. Figure 3.20 presents the estimates of the simulation. It is also observed from the literature that the proposed controller gives a good control performance in terms of optimal attitude tracking compared to the attitude tracking errors of [5, 156].
Figure 3.7: IMU data measurements from gyroscope.

Figure 3.8: IMU data measurements from accelerometer.
Figure 3.9: IMU data measurements from magnetometer.

Figure 3.10: LS based estimate $\hat{\theta}_\phi$ of the uncertain inertia parameter $\theta^*_\phi$. 

Figure 3.11: Attitude estimation of the Qball-X4.

Figure 3.12: Attitude angle estimation of the Qball-X4 from 50 to 100 [sec].
Figure 3.13: Attitude tracking error of the Qball-X4 using complementary filter.

Figure 3.14: Attitude tracking error of the Qball-X4 using Kalman filter.
Figure 3.15: Optimal attitude control inputs for the complementary filter.

Figure 3.16: Optimal attitude control inputs for Kalman filter.
Figure 3.17: Motor PWM control inputs $v_y$.

Figure 3.18: PID vs ALQT performance comparison: attitude tracking error.
Figure 3.19: PID vs ALQT performance comparison: attitude control input.

Figure 3.20: LS based estimate $\hat{\theta}_\phi$ of the uncertain inertia parameter $\theta^*_\phi$ for the simulation.
3.9 Summary and Remarks

The adaptive linear quadratic tracking (ALQT) scheme has been developed to control and stabilize the attitude of the Qball-X4 quadrotor UAV system in an optimal sense. The proposed adaptive controller is designed by indirect approach and combined with the LS based parameter identification (PI) to eliminate the influences of inertial uncertainties. Additionally, Kalman filter has been designed for canceling noise effects on the attitude estimation data to provide more reliable feedbacks to the controller and it is compared with Complementary filter. All analytical analyses and designs are verified by the two experimental tests. We witness that the ALQT design in experiments works satisfactorily in terms of the optimal tracking performance. In Kalman filter vs Complementary filter, although both filter designs are good to canceling noise effects on the estimated attitude data, Kalman filter gives a better accuracy and reliable attitude estimation. Thus, the experimental results show that the quadrotor UAV has more robust behavior and better tracking error with the estimated attitude data by Kalman filter comparing with the Complementary filter.
Chapter 4

Robust Adaptive Control of a Quadrotor UAV with Guaranteed Tracking Performance

4.1 Introduction

In the literature, some advanced nonlinear control methods have been studied for tracking problem of the quadrotor UAV, including the backstepping, sliding-mode and feedback linearization based ones [13, 39, 90, 107]. The above mentioned studies rely on the full knowledge of dynamics for robust tracking performances. For the purpose of ensuring the control robustness in case of model uncertainties and disturbances, [103] introduces a robust three-loop design, which comprises of the nominal position and attitude controllers with a robust compensator, to deal with nonlinearities, parametric uncertainties and disturbances. In [127], a robust backstepping control scheme based on integral sliding modes is used to compensate bounded disturbances, including wind gust, side-slip aerodynamics, and drags in both position and the attitude dynamics. In [141], terminal sliding mode control (TSMC) is studied for position and attitude tracking, subject to such disturbances. [140] extends the TSMC approach to the globally fast dynamic TSMC, which not only provides finite-time position and attitude tracking control and eliminates the chattering, but also rejects the external disturbances.

Guaranteeing transient and steady-state tracking performances for highly nonlinear systems with model uncertainties and disturbances is a challenging and important control task. The earlier studies [24, 149] have introduced a prescribed performance function based
error transformation approach, which is capable of transferring the original constrained tracking error system to an unconstrained form. As a big advantage of this approach, the unconstrained system is capable of guaranteeing prescribed performance when the stability of the unconstrained system is proven. Hence, the maximum overshoot/undershoot, the convergence rate and steady-state error are easily tunable.

In [124], a fault tolerant control is studied and prescribed performance bound (PPB) technique is also used to improve the transient attitude tracking performance of UAV system. In [169], a robust controller with guaranteed certain predefined transient performance is designed for the attitude of the rigid spacecraft system on SO(3). [157] studies the backstepping based adaptive compensation control strategy with guaranteed transient attitude tracking performance for the quadrotor UAV with partial loss of rotational speed, limited air flow disturbance and inertial uncertainties. In [72], a robust adaptive control strategy based on sliding mode and PPB techniques for time-varying payload and wind gust disturbance. In [71], a PPB based formation control scheme is designed for systems of multiple quadrotor UAVs.

In this chapter, inspired by [24, 149], we design and implement a robust adaptive controller with guaranteed transient and steady-state tracking error performances in case of under-actuated dynamics, nonlinearities and model uncertainties. To achieve this goal, we design a backstepping based position controller using a PPB based error transformation approach which ensures bounded trajectory tracking errors, and a backstepping based adaptive attitude controller combined with a least squares (LS) based parameter identification (PI) algorithm.

4.2 Quadrotor UAV Dynamics

A nonlinear dynamic modeling of the quadrotor UAV motion dynamics (2.10) is considered from Chapter 2. The nonlinear dynamics is separated into three strict-feedback sub-systems as follows:

a) Lateral position dynamics is

\[ \dot{p}_l = [\dot{p}_x, \dot{p}_y]^T = [v_x, v_y]^T, \]

\[ \dot{v}_l = [\dot{v}_x, \dot{v}_y]^T = \frac{4T_z}{m} [f_{1x}, f_{1y}]^T, \]

\[ f_{1x} = \{ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \}, \]

\[ f_{1y} = \{ \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \}, \]
where \( p_l, v_l \) and \( T_z \) are lateral position, lateral velocity and thrust in the body frame \( O_b \); \( m \) is total mass; \( \{ \phi, \vartheta, \psi \} \) are Euler angles.

\( b) \) Altitude dynamics is

\[
\begin{align*}
\dot{p}_z &= v_z, \\
\dot{v}_z &= \frac{4}{m}(\cos \phi \cos \vartheta) T_z - g, \\
\dot{T}_z &= -bT_z + Kb u_z,
\end{align*}
\]

where \( p_z \) and \( v_z \) are position and velocity in longitudinal motion of the \( O_b \); \( g \) is gravity; \( b \) is actuator bandwidth; \( K \) is positive armature gain.

\( c) \) Attitude dynamics is

\[
\begin{align*}
\dot{\varphi} &= w_{\varphi}, \\
\dot{w}_{\varphi} &= A_1 w_{\varphi} + A_2 f_2(w_{\varphi}) + B T_{\varphi}, \\
\dot{T}_{\varphi} &= -b T_{\varphi} + Kb u_{\varphi},
\end{align*}
\]

where \( A_1 = \text{diag}(a_{11}, a_{12}, a_{13}) = \text{diag}(d_\phi, d_\vartheta, d_\psi); A_2 = \text{diag}(a_{21}, a_{22}, a_{23}) = \text{diag}(\frac{J_\phi - J_\vartheta}{J_\psi}, \frac{J_\theta - J_\psi}{J_\phi}, \frac{J_\psi - J_\phi}{J_\theta}); f_2(w_{\varphi}) = [f_{21}(w_{\varphi}), f_{22}(w_{\varphi}), f_{23}(w_{\varphi})]^T = \begin{bmatrix} \dot{\varphi} \\ \dot{\vartheta} \\ \dot{\psi} \end{bmatrix}; B = \text{diag}(b_1, b_2, b_3) = \text{diag}(2l J_{\phi}, 2l J_{\theta}, 4K_\psi J_{\psi}) \right) \triangleq [T_{\phi}, T_{\vartheta}, T_{\psi}]^T, u_{\varphi} \triangleq [u_{\varphi 1}, u_{\varphi 2}, u_{\varphi 3}]^T = \text{diag}(J_{\phi}, J_{\vartheta}, J_{\psi}) \right) \triangleq [J_{\phi}, J_{\vartheta}, J_{\psi}] \) and \( d_{\varphi} \in (d_{\phi}, d_{\theta}, d_{\psi}) \) are Euler angles, angular velocities, attitude thrusts, attitude control inputs, inertias and drags in the \( O_b \), respectively.

### 4.3 Control Problem

Consider the position (lateral and altitude) dynamics (4.1)-(4.5) and the attitude dynamics (4.6)-(4.8). The following practical assumptions are made.

**Assumption 4.3.1.** Attitude angles are bounded as

\[
-\frac{\pi}{2} < \phi < \frac{\pi}{2}, \quad -\frac{\pi}{2} < \vartheta < \frac{\pi}{2}, \quad -\frac{\pi}{2} < \psi < \frac{\pi}{2}.
\]
Assumption 4.3.2. The desired trajectory \( p_d(t) \in \mathbb{R}^3 \) and its derivatives are known and bounded function of time.

Assumption 4.3.3. The states \( p_l, v_l, p_z, v_z, T_z, \varphi, w_\varphi, T_\varphi \) of the quadrotor UAV are measurable.

The control problem of the chapter is stated as follows:

Problem 4.3.1. Design the control units to generate the command signals \( u_z \) and \( u_\varphi \) for feeding the PWM generator that generates the motor control input signal \( v_r \), per the diagram in Figure 4.1 such that the system output \( p(t) = [p_x(t), p_y(t), p_z(t)]^T \) tracks the desired trajectory \( p_d(t) = [p_{xd}(t), p_{yd}(t), p_{zd}(t)]^T \):

(a) Design a backstepping based position controller using a PPB based error transformation to generate \( \varphi_d \) and \( u_z \) so that the transient and steady-state behaviors of the tracking error

\[
e(t) = p(t) - p_d(t) = [e_x(t), e_y(t), e_z(t)]^T
\]

are guaranteed within the prescribed bounds.
(b) Design a backstepping based adaptive attitude controller combined with a least squares (LS) based parameter identification (PI) algorithm to generate $u_\phi$ so that $\phi_d$ tracks the desired attitude $\phi_d$ and robustness is increased in presence of the model uncertainties.

4.4 Robust Adaptive Tracking Control Design with Guaranteed Error Performance

The low-level closed-loop system ($L$) is considered in the two-layer: position ($P$) and attitude ($A$) control schemes for the three sub-systems of the quadrotor UAV as seen in Figure 4.1. As formally stated in Problem 4.3.1, the proposed control scheme is designed in the following sections.

4.4.1 Position Control Design with Guaranteed Error

The position control scheme is designed by using backstepping procedures combined with the PPB based error transformation system [24, 149]. The following subsection first introduces the error transformation for each component $e_j$, $j \in x, y, z$ of the tracking error (4.10). Then, the lateral and the altitude control designs are presented.

Error Transformed System

a) Performance function: To transform the prescribed performance characteristics into $e_j$ similar to [24, 149], a decreasing smooth performance function $\rho_j(t) : \mathbb{R}_+ \rightarrow \mathbb{R}_+ \setminus \{0\}$ with $\lim_{t \to \infty} \rho_j(t) = \rho_{j\infty} > 0$ is defined. In this thesis, the function is chosen as

$$\rho_j(t) = (\rho_{j0} - \rho_{j\infty})e^{-k_j t} + \rho_{j\infty}, \quad \text{(4.11)}$$

where $\rho_{j0} > \rho_{j\infty}$ and $k_j > 0$ are design parameters. Then, the control performance condition is defined as satisfaction of

$$\delta_j \rho_j(t) > e_j(t) > -\delta_j \rho_j(t), \quad \forall t \geq 0, \quad \text{(4.12)}$$

where $1 \geq \delta_j$ and $\delta_j > 0$ are prescribed scalars. $\delta_j \rho_j(0)$ and $-\delta_j \rho_j(0)$ are the upper and the lower bounds of the overshoot and the undershoot of $e_j(t)$. Use of (4.12) is illustrated in Figure 4.2.
Figure 4.2: Graphical representation of (4.12) for prescribed tracking error behavior with $\overline{\delta} = 0.2$ and $\bar{\delta} = 1$.

b) Error transformation: To solve the control Problem 1-(a), tracking error transformation will be established by transforming condition (4.12) into an equivalent unconstrained one. For this purpose, we define a smooth, strictly increasing and invertible function $S_j(.)$ with following properties:

(i) $\overline{\delta}_j > S_j(\varepsilon) > -\delta_j$, for any $\varepsilon \in \mathbb{R}$

(ii) $\lim_{\varepsilon \to +\infty} S_j(\varepsilon) = \overline{\delta}_j$ and $\lim_{\varepsilon \to -\infty} S_j(\varepsilon) = -\delta_j$,

(iii) $S_j(0) = 0$.

A function satisfying these properties is $S_j(\varepsilon) = \frac{\overline{\delta}_j e^{(\varepsilon+c)} - \delta_j e^{-(\varepsilon+c)}}{e^{(\varepsilon+c)} + e^{-(\varepsilon+c)}}$ where $c = \frac{\ln(\overline{\delta}_j/\delta_j)}{2}$ [149].

The condition (4.12) is satisfied [24, 149] if

$$\varepsilon_j = S_j^{-1}(\Lambda_j(t))$$

$$= \frac{1}{2} \ln(\overline{\delta}_j \Lambda_j(t) + \delta_j \bar{\delta}_j) - \frac{1}{2} \ln(\overline{\delta}_j \delta_j - \delta_j \Lambda_j(t)), \quad (4.13)$$

where $\Lambda_j(t) = \frac{e_j(t)}{\rho_j(t)}$, is guaranteed to be bounded.
The time derivative of $\varepsilon_j$ is in (4.13) computed as
\[
\dot{\varepsilon}_j = \frac{\partial \varepsilon_j}{\partial \Lambda_j} \dot{\Lambda}_j
= \frac{1}{2} \left( \frac{1}{\Lambda_j + \delta_j} - \frac{1}{\Lambda_j - \delta_j} \right) \left( \frac{\dot{e}_j}{\rho_j} - \frac{e_j\dot{\rho}_j}{\rho_j^2} \right)
= \Theta_j \left( \dot{e}_j - \frac{e_j\dot{\rho}_j}{\rho_j} \right),
\tag{4.14}
\]
where $\Theta_j = \frac{1}{2\rho_j} \left( \frac{1}{\Lambda_j + \delta_j} - \frac{1}{\Lambda_j - \delta_j} \right) \neq 0$ is well defined by (i) and (4.13).

To guarantee the transient and steady-state tracking performances, we now combine the PPB based error transformation (4.11)-(4.14) with the lateral position and the altitude dynamics (4.1)-(4.5) by replacing the error dynamics of the equations (4.1) and (4.3) with $\dot{e}_j$ in (4.14) for $j \in \{x, y, z\}$.

**Lateral Position Backstepping Control with PPB**

The reference angle $\varphi_d$ is generated to drive the quadrotor UAV to the desired lateral position $p_{jd} \in \{p_{xd}, p_{yd}\}$. The backstepping procedure is combined with the PPB based transformation as presented in subsection 4.4.1 to guarantee the prescribed lateral tracking error performance.

For each of $j \in \{x, y\}$, first define
\[
z_{j1} = \varepsilon_j \text{ and } z_{j2} = v_j - \alpha_{j1},
\tag{4.15}
\]
where $\alpha_{j1}$ is to be chosen in the sequel.

From (4.14) and the definition of $z_{j2}$ in (4.15), we have
\[
\dot{z}_{j1} = \Theta_j \left( \dot{p}_j - \dot{p}_{jd} - \frac{e_j\dot{\rho}_j}{\rho_j} \right)
= \Theta_j \left( z_{j2} + \alpha_{j1} - \dot{p}_{jd} - \frac{e_j\dot{\rho}_j}{\rho_j} \right).
\tag{4.16}
\]
To stabilize (4.16), $\alpha_{j1}$ is chosen
\[
\alpha_{j1} = \left( -\frac{k_1 z_{j1}}{\Theta_j} + \dot{p}_{jd} + \frac{e_j\dot{\rho}_j}{\rho_j} \right), \quad k_1 > 0,
\tag{4.17}
\]

so that
\[
\dot{z}_1 = \Theta z_2 - k_1 z_1.
\] (4.18)

This leads to
\[
\dot{z}_2 = \dot{v}_j - \dot{\alpha}_1 = \frac{4T_z}{m} f_{1j} - \dot{\alpha}_j.
\] (4.19)

To stabilize (4.19), choosing \( f_{1l} \) such that
\[
f_{1j} = \frac{m}{4T_z} (\Theta z_1 - k_2 z_2 + \dot{\alpha}_j), \quad k_2 > 0,
\] (4.20)

where \( \dot{\alpha}_j = \frac{\partial \alpha_j}{\partial p_j} v_j + \frac{\partial \alpha_j}{\partial v_{1d}} v_{1d} + \frac{\partial \alpha_j}{\partial \dot{v}_{1d}} \dot{v}_{1d} + \frac{\partial \alpha_j}{\partial \rho_j} \dot{\rho}_j + \frac{\partial \alpha_j}{\partial \dot{\rho}_j} \ddot{\rho}_j \), we have
\[
\dot{z}_2 = -\Theta z_1 - k_2 z_2.
\] (4.21)

Using the equations (4.2) and (4.20), the desired pitch and roll angles are generated as
\[
\phi_d = \arcsin (\sin \psi f_{1x} - \cos \psi f_{1y}),
\] (4.22)
\[
\vartheta_d = \arcsin \left( \frac{f_{1x} - \sin \phi \sin \psi}{\cos \phi \cos \psi} \right),
\] (4.23)

and to prevent yaw motion’s direct effect on the lateral motion of the quadrotor UAV, the desired yaw angle is set to zero; as \( \psi_d(t) = 0 \) for all \( t \geq 0 \). Therefore, the desired attitude angle is designed as \( \varphi_d = [\varphi_{d1}, \varphi_{d2}, \varphi_{d3}]^T = [\phi_d, \vartheta_d, 0]^T \).

**Remark 4.4.1.** The total thrust force is \( T_z > 0 \), \( \forall t \geq 0 \) so that the system avoids singularity during the operation.

**Altitude Backstepping Control with PPB**

The backstepping technique is utilized to design the altitude controller with the PPB by defining
\[
z_3 = \varepsilon_z, \quad z_4 = v_z - \alpha_4, \text{ and } z_5 = T_z - \alpha_5.
\] (4.24)

where \( \alpha_5 \) is to be chosen in the sequel.
From (4.14) and the definition of $z_4$ in (4.24), we then write
\[
\dot{z}_3 = \Theta_z (\dot{p}_z - \dot{p}_{zd} - \frac{e_z \dot{z}_d}{p_z}) \\
= \Theta_z (z_4 + \alpha_4 - \dot{p}_{zd} - \frac{e_z \dot{z}_d}{p_z}). \tag{4.25}
\]
To stabilize (4.25), $\alpha_4$ is chosen as
\[
\alpha_4 = \left( -\frac{k_3 z_3}{\Theta_z} + \dot{p}_{zd} + \frac{e_z \dot{z}_d}{p_z} \right), \quad k_3 > 0, \tag{4.26}
\]
so that
\[
\dot{z}_3 = \Theta_z z_4 - k_3 z_3. \tag{4.27}
\]
This leads to
\[
\dot{z}_4 = \dot{v}_z - \dot{\alpha}_4 = \frac{4 \cos \phi \cos \vartheta}{m} T_z - g - \dot{\alpha}_4 \\
= \frac{4 \cos \phi \cos \vartheta}{m} (z_5 + \alpha_5) - g - \dot{\alpha}_4. \tag{4.28}
\]
To stabilize (4.28), $\alpha_5$ is chosen as
\[
\alpha_5 = \frac{m}{4 \cos \phi \cos \vartheta} (g - k_4 z_4 + \dot{\alpha}_4 - \Theta_z z_3), \quad k_4 > 0, \tag{4.29}
\]
where $\dot{\alpha}_4 = \frac{\partial \alpha_4}{\partial p_z} \dot{v}_z + \frac{\partial \alpha_4}{\partial v_z} \dot{v}_zd + \frac{\partial \alpha_4}{\partial p} \dot{p}_z + \frac{\partial \alpha_4}{\partial \dot{p}} \ddot{p}_z$, so that
\[
\dot{z}_4 = -\Theta_z z_3 + \frac{4 \cos \phi \cos \vartheta}{m} z_5 - k_4 z_4. \tag{4.30}
\]
We further have
\[
\dot{z}_5 = \dot{T}_z - \dot{\alpha}_5 = -bT_z + Kbu_z - \dot{\alpha}_5. \tag{4.31}
\]
To stabilize (4.31), the altitude control input is designed as
\[
u_z = \frac{1}{Kb} \left[ bT_z - k_5 \dot{z}_5 - \frac{4 \cos \phi \cos \vartheta}{m} z_4 + \dot{\alpha}_5 \right], \quad k_5 > 0, \tag{4.32}
\]
where $\dot{\alpha}_5 = \frac{\partial \alpha_5}{\partial p_z} \dot{v}_z + \frac{\partial \alpha_5}{\partial v_z} \dot{v}_zd + \frac{\partial \alpha_5}{\partial v} \ddot{v}_z + \frac{\partial \alpha_5}{\partial p} \dot{p}_z + \frac{\partial \alpha_5}{\partial \dot{p}} \ddot{p}_z + \frac{\partial \alpha_5}{\partial \dot{p}} \dddot{p}_z$, so that
(4.31) becomes
\[
\dot{z}_5 = -\frac{4 \cos \phi \cos \vartheta}{m} z_4 - k_5 z_5. \tag{4.33}
\]
Remark 4.4.2. Assumption 4.3.1 guarantees that the designed control law (4.32) is always well-defined with the coefficient \(-\frac{4\cos \phi \cos \theta}{m}\) of \(z_4\) always negative.

4.4.2 Adaptive Attitude Control Design

The proposed backstepping based adaptive control scheme is composed of two-parts: a parameter identification (PI) algorithm and an indirect adaptive backstepping control law as follows.

Parameter Identification Algorithm

The attitude dynamics equation (4.7) can be rewritten for each of \(i \in \{1, 2, 3\}\) in the form of

\[
b_i^{-1}w_\phi = b_i^{-1}a_{1i}w_\phi + b_i^{-1}a_{2i}f_{2i}(w_\phi) + T_{\phi i},
\]

which leads to the equation

\[
T_{\phi i} = \bar{a}_{1i}w_\phi + \bar{a}_{2i}f_{2i}(w_\phi) - \bar{a}_{3i}w_\phi,
\]

where \(\bar{a}_{1i} = b_i^{-1}a_{1i}, \bar{a}_{2i} = b_i^{-1}a_{2i}\) and \(\bar{a}_{3i} = b_i^{-1}\).

To avoid numerical differentiation, (4.35) is rewritten using the stable filter \(\frac{1}{s + \lambda}, \lambda > 0\), as

\[
\frac{1}{s + \lambda}[T_{\phi i}] = \bar{a}_{1i}\left[w_\phi\right] + \bar{a}_{2i}\left[f_{2i}(w_\phi)\right] - \bar{a}_{3i}s\left[w_\phi\right],
\]

where the Euler rates in \(w_\phi \in \mathbb{R}\) and the control thrust inputs in \(T_{\phi i} = \frac{Kb}{s + b}u_\phi \in \mathbb{R}\) are measurable; \(K\) and \(b\) are constant parameters. Based on (4.36), a parametric model is defined

\[
z_{\phi i} = \theta^*_{\phi i} T_{\phi i},
\]

\[
z_{\phi i} = \frac{1}{s + \lambda}[T_{\phi i}] \in \mathbb{R}, \quad \theta^*_{\phi i} = [\bar{a}_{1i}, \bar{a}_{2i}, \bar{a}_{3i}]^T \in \mathbb{R}^3, \quad T_{\phi i} = \begin{bmatrix}
\frac{1}{s + \lambda}[w_\phi] \\
\frac{1}{s + \lambda}[f_{2i}(w_\phi)] \\
-\frac{s}{s + \lambda}[w_\phi]
\end{bmatrix} \in \mathbb{R}^3.
\]

Assumption 4.4.1. The upper and lower limits of \(\theta^*_{\phi 3}(t)\) are known, i.e. \(0 < \theta^*_{\phi 3} \leq \hat{\theta}^*_{\phi 3}(t) \leq \bar{\theta}^*_{\phi 3}\) for some known \(\hat{\theta}^*_{\phi 3}, \bar{\theta}^*_{\phi 3} > 0\).
Next, the recursive LS algorithm [74] is applied with forgetting factor to (4.37) to produce the estimate \( \hat{\theta}_{\varphi_i}(t) = [\hat{\theta}_{\varphi_{i1}}, \hat{\theta}_{\varphi_{i2}}, \hat{\theta}_{\varphi_{i3}}]^T \) of \( \theta^*_{\varphi_i} \) as follows:

\[
\dot{\hat{\theta}}_{\varphi_i}(t) = Pr(P(t)\Phi_{\varphi_i}(t)\epsilon_i(t)), \quad \hat{\theta}_{\varphi_i}(0) = \hat{\theta}_{\varphi_0},
\]

\[
\dot{\hat{\theta}}_{\varphi_i}(t) = \begin{cases} 
\beta P - P \frac{\Phi_{\varphi_i}^T P}{m_{\varphi_i}^2} P, & \text{if } \theta_{\varphi_{i3}} < \hat{\theta}_{\varphi_{i3}} < \bar{\theta}_{\varphi_{i3}} \\
0, & \text{otherwise}
\end{cases},
\]

\[
\epsilon_i(t) = \frac{z_{\varphi_i}(t) - \hat{z}_{\varphi_i}(t)}{m_{\varphi_i}^2(t)}, \quad m_{\varphi_i}^2(t) = 1 + \Phi_{\varphi_i}^T(t)\Phi_{\varphi_i}(t),
\]

\[
\hat{z}_{\varphi_i}(t) = \hat{\theta}_{\varphi_i}^T(t)\Phi_{\varphi_i}(t),
\]

where \( P(0) = P_0 \in \mathbb{R}^{3 \times 3} \) is a positive definite (covariance) matrix, \( m_n \) is the normalizing signal and \( \epsilon \) is the estimation error, and \( Pr(.) \) is the parameter projection operator [74] used to guarantee that \( \hat{\theta}_{\varphi_{i3}} \in [\underline{\theta}_{\varphi_{i3}}, \bar{\theta}_{\varphi_{i3}}] \).

**Lemma 4.4.3** (Stability and Convergence). Consider the LS based PI algorithm (4.38), applied to the attitude dynamics (4.6)-(4.8). It is guaranteed that all the signals in (4.38), including \( P \) and \( P^{-1} \), are bounded. Further, if \( \Phi_{\varphi ni} = \frac{\Phi_{\varphi i}}{m_{\varphi i}} \) persistently exciting, i.e. if

\[
\frac{1}{T} \int_{t}^{t+T} \Phi_{\varphi ni}^T(t)\Phi_{\varphi ni}(t) d\tau \geq \alpha_0
\]

for all \( t \geq 0 \) and some \( T, \alpha_0 > 0 \), then (4.38) ensures that \( \theta_{\varphi_i}(t) \to \theta^*_{\varphi_i} \) as \( t \to \infty \). The convergence of \( \theta_{\varphi_i}(t) \to \theta^*_{\varphi_i} \) is exponential for \( \beta > 0 \).

**Proof.** The result is a direct corollary of the more general Theorem 3.7.4 in [74].

**Adaptive Control Design**

A backstepping based indirect adaptive controller is proposed for tracking the desired attitude angles. For each of \( i \in \{1, 2, 3\} \), we first define the error system

\[
z_{6i} = \varphi_i - \varphi_{di} \in \mathbb{R}, \quad \tilde{z}_{7i} = w_{\varphi_i} - \alpha_{7i} \in \mathbb{R}.
\]

where \( \alpha_7 \) is to be selected in the sequel.

From (4.40) and (4.6), we have

\[
\dot{z}_{6i} = \dot{\varphi}_i - \dot{\varphi}_{di} = \tilde{z}_{7i} + \alpha_{7i} - \dot{\varphi}_{di}.
\]

To stabilize (4.41), \( \alpha_7 \) is chosen as

\[
\alpha_{7i} = -k_6 z_{6i} + \dot{\varphi}_{di}, \quad k_6 > 0,
\]
so that
\[ \dot{z}_{6i} = \ddot{z}_{7i} - k_6 z_{6i}. \] (4.43)

At the second step, from (4.40), we define
\[ z_{7i} = b_i^{-1} \ddot{z}_{7i} = b_i^{-1} w_{\varphi i} - b_i^{-1} \alpha_{7i}, \] (4.44)
then from (4.34), (4.35) and (4.44), we obtain
\[ \dot{z}_{7i} = b_i^{-1} \dot{w}_{\varphi i} - b_i^{-1} \dot{\alpha}_{7i} = b_i^{-1} a_{1i} w_{\varphi i} + b_i^{-1} a_{2i} f_{2i}(w_{\varphi}) + T_{\varphi i} - b_i^{-1} \dot{\alpha}_{7i}. \] (4.45)

To stabilize (4.45), \( T_{\varphi i} \) is chosen as
\[ T_{\varphi i} = -\hat{a}_{1i} w_{\varphi i} - \hat{a}_{2i} f_{2i}(w_{\varphi}) + \hat{a}_{3i} \dot{\alpha}_{7i} - k_7 \hat{a}_{3i} z_{7i} - z_{6i}, \] (4.46)
where \( \dot{\alpha}_{7i} = \frac{\partial \theta_{7i}}{\partial \varphi_i} \dot{\varphi}_i + \frac{\partial \theta_{7i}}{\partial \varphi_{di}} \dot{\varphi}_{di} + \frac{\partial \theta_{7i}}{\partial \varphi_{di}} \ddot{\varphi}_{di} \) and \( k_7 > 0 \), so that
\[ \dot{z}_{7i} = -k_7 z_{7i} - z_{6i} - \hat{a}_{1i} w_{\varphi i} - \hat{a}_{2i} f_{2i}(w_{\varphi}) + \hat{a}_{3i} \dot{\alpha}_{7i}, \] (4.47)
where \( \hat{a}_{1i} = \hat{a}_{1i} - a_{1i}; \hat{a}_{2i} = \hat{a}_{2i} - a_{2i}; \hat{a}_{3i} = \hat{a}_{3i} - a_{3i}. \)

**Remark 4.4.4.** Since the actuator dynamics have been fully considered in the altitude and attitude design process, the control laws (4.32) and (4.46) can be regarded as the actual PWM control signals of the motors, which makes an easier way to application in practice. After all, the altitude \( u_z \) and the attitude \( u_{\psi} \) control inputs generated via \( T_{\varphi i} \) are mixed and converted to the actuator motor PWM inputs \( v_r(t), r = 1, \ldots, 4 \) as discussed in [90] for providing the overall motion.

### 4.5 System Stability Analysis

The stability of the overall closed-loop system is analyzed in this section, where the analysis results are summarized in the following theorem.

**Theorem 4.5.1.** Consider the adaptive control scheme (4.15), (4.20), (4.24), (4.32), (4.38), (4.39), (4.40), (4.46) applied to Problem 4.3.1. Assume that \( \Phi_{\psi ni} \) in Lemma 1 is persistently exciting. It is guaranteed that all the signals are bounded and the tracking error \( e(t) = [e_z(t), e_y(t), e_z(t)]^T \) asymptotically converges to zero. Further, pre-defined control performance condition (4.12) is satisfied.

67
Proof. For proving the stability of the lateral position control system, define the Lyapunov function

\[ V_{la}(z_1, z_2) = \frac{1}{2} \sum_{i=1}^{2} z_i^2. \]  

From (4.17) and (4.20), the time derivative of (4.48) is obtained as

\[ \dot{V}_{la}(z_1, z_2) = z_1(\Theta_l z_2 - k_1 z_1) + z_2(-\Theta_l z_1 - k_2 z_2) \]

\[ = -k_1 z_1^2 - k_2 z_2^2 \leq -2\min\{k_1, k_2\} V_{la}. \]  

Therefore, the origin \( z_1 = z_2 = 0 \) of the error system (4.15)-(4.21) is exponentially stable.

Then, by defining the Lyapunov function for the altitude as

\[ V_{al}(z_3, z_4, z_5) = \frac{1}{2} \sum_{i=3}^{5} z_i^2. \]  

From (4.26), (4.29) and (4.32), the time derivative of (4.50) is obtained as

\[ \dot{V}_{al}(z_3, z_4, z_5) = \Theta_2 z_3 z_4 - k_3 z_3^2 - k_4 z_4^2 - k_5 z_5^2 \leq -2\min\{k_3, k_4, k_5\} V_{al}. \]  

Therefore, the origin \( z_3 = z_4 = z_5 = 0 \) of the error system (4.24)-(4.33) is exponentially stable.

The stability of the attitude motion is established by defining the Lyapunov function as

\[ V_{at}(z_6, z_7_i) = \frac{1}{2} z_6^2 + \frac{1}{2} b_i z_7_i^2. \]  

From (4.42) and (4.46), the time derivative of (4.52) is obtained as

\[ \dot{V}_{at}(z_6, z_7_i) = z_6 \dot{z}_6 + z_7_i \dot{z}_7_i \]

\[ = -k_6 z_6^2 - k_7 b_i z_7_i^2 - \tilde{\theta}_{\phi i}^T [w_{\phi i} z_7_i, f_{2i}(w_{\phi i} z_7_i), -\dot{\alpha}_{7_i} z_7_i]^T, \]

where \( b_i \) is positive inertial based parameter under Assumption 4.4.1 and by Lemma 4.4.3 with the boundedness of the PI algorithm (4.38) guarantees that \( \tilde{\theta}_{\phi i} \) asymptotically converges to zero, which implies that as \( t \to \infty \)

\[ \dot{V}_{at}(z_6, z_7_i) \to -k_6 z_6^2 - k_7 b_i z_7_i^2 \leq -2\min\{k_6, k_7\} V_{at}. \]  

Hence, the origin \( z_6 = z_7_i = 0 \) of the error system (4.40)-(4.47) is asymptotically stable.

\[ \square \]
4.6 Alternative Adaptive Attitude Control Design

To design an alternative adaptive attitude control scheme that does not require persistent excitation for solving Problem 4.3.1, we redefine the Lyapunov function (4.52) as

$$V_{at}(z_{6i}, z_{7i}, \tilde{\theta}_{\psi i}) = \frac{1}{2}z_{6i}^2 + \frac{1}{2}b_i z_{7i}^2 + \frac{\gamma^{-1}}{2}\tilde{\theta}_{\psi i}^T\tilde{\theta}_{\psi i},$$

(4.55)

where \(\gamma\) is positive design parameter.

The time derivative of (4.55) is obtained, similarly to Section 4.5, as

$$\dot{V}_{at}(z_{6i}, z_{7i}, \tilde{\theta}_{\psi i}) = z_{6i}\dot{z}_{6i} + b_i z_{7i}\dot{z}_{7i} + \frac{\gamma^{-1}}{2}\tilde{\theta}_{\psi i}^T\dot{\tilde{\theta}}_{\psi i},$$

(4.56)

Choosing the update law \(\dot{\tilde{\theta}}_{\psi i} = \dot{\hat{\theta}}_{\psi i}\) in (4.56) as

$$\dot{\hat{\theta}}_{\psi i} = -\gamma[w_{\psi i}\tilde{z}_{7i}, f_{2i}(w_{\varphi})\tilde{z}_{7i}, -\dot{\tilde{\alpha}}_{7i}\tilde{z}_{7i}]^T,$$

(4.57)

we obtain

$$\dot{V}_{at}(z_{6i}, z_{7i}, \tilde{\theta}_{\psi i}) = -k_6 z_{6i}^2 - k_7 b_i z_{7i}^2,$$

(4.58)

which implies that \(z_{6i}\) and \(z_{7i}\) asymptotically converge to zero and \(\dot{\hat{\theta}}_{\psi i}\) asymptotically converges to a constant. Therefore, we have the following result.

**Theorem 4.6.1.** Consider the adaptive control scheme (4.15), (4.20), (4.24), (4.32), (4.40), (4.46), (4.57) applied to Problem 4.3.1. It is guaranteed that all the signals are bounded and the tracking error \(e(t) = [e_x(t), e_y(t), e_z(t)]^T\) asymptotically converges to zero. Further, pre-defined control performance condition (4.12) is satisfied.

4.7 Simulations and Experimental Tests

4.7.1 Testbed Platform and Benchmark Controllers

In simulations and experiments, the quadrotor UAV testbed platform is Qball-X4, developed by Quanser Inc. \[172\]. Designed controllers are implemented via MATLAB.
Table 4.1: The Qball-X4 quadrotor UAV dynamic parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$</td>
<td>Total mass</td>
<td>1.4 [Kg]</td>
</tr>
<tr>
<td>$l$</td>
<td>Distance between $O_b$ and the motor</td>
<td>0.2 [m]</td>
</tr>
<tr>
<td>$g$</td>
<td>Gravity</td>
<td>9.81 [m/s$^2$]</td>
</tr>
<tr>
<td>$K$</td>
<td>Positive armature gain</td>
<td>120 [N]</td>
</tr>
<tr>
<td>$K_\psi$</td>
<td>Thrust-to-moment gain</td>
<td>4 [Nm]</td>
</tr>
<tr>
<td>$b$</td>
<td>Actuator bandwidth</td>
<td>15 [rad/s]</td>
</tr>
<tr>
<td>$J_{\phi_0}$</td>
<td>Nominal pitch inertia</td>
<td>0.03 [Kgm$^2$]</td>
</tr>
<tr>
<td>$J_{\theta_0}$</td>
<td>Nominal roll inertia</td>
<td>0.03 [Kgm$^2$]</td>
</tr>
<tr>
<td>$J_{\psi_0}$</td>
<td>Nominal yaw inertia</td>
<td>0.04 [Kgm$^2$]</td>
</tr>
<tr>
<td>$d_{\phi_0}$</td>
<td>Nominal drag on pitch</td>
<td>-0.01 [Nms$^2$]</td>
</tr>
<tr>
<td>$d_{\theta_0}$</td>
<td>Nominal drag on roll</td>
<td>-0.012 [Nms$^2$]</td>
</tr>
<tr>
<td>$d_{\psi_0}$</td>
<td>Nominal drag on yaw</td>
<td>-0.009 [Nms$^2$]</td>
</tr>
</tbody>
</table>

Simulink® and Quanser Quarc interfaces. The dynamic parameters of the Qball-X4 are specified as shown in Table-4.1. The nominal values of inertias and drags are used for non-adaptive designs. Detailed description of the platform can be found in Section 3.8.1.

Three different benchmark controllers are used in comparison with the proposed control design as follows. Benchmark controller-1 is composed of a PID position control with $K_{pb} = 0.006$, $K_{ib} = 0.01$ and $K_{db} = 0.008$, and a LQR attitude control with $K_L = [0.0616, 0.0126, 1.1066]$. Benchmark controller-2 is composed of a full backstepping control by using backstepping gains with $k_1 = 0.1$, $k_2 = 0.4$, $k_3 = 5$, $k_4 = 20$, $k_5 = 100$, $k_6 = 5$ and $k_7 = 5$. Benchmark controller-3 is composed of a full backstepping control with guaranteed tracking error by using specified PPB parameters and same control settings as used in benchmark controller-2. The proposed adaptive backstepping control with guaranteed tracking error is designed by using PI model parameters which are specified as the forgetting factor $\beta = 0.001$, the initial covariance matrix $P_0 = 10^5I_{3\times3}$ and the initial parameter matrix $\theta_{\phi_0} = [−0.75; −0.5; 0.05]$, and same control and PPB settings as used in benchmark controller-3.
4.7.2 Simulation Tests

In simulations, a spiral trajectory is considered as
\[
p_d(t) = [10 \cos(0.04t), 10 \sin(0.04t), 0.025t]^T,
\]
with the initial position \( p(0) = [9, 1, 0.2]^T \). The PPB parameters are specified as \( \rho_{l0} = 3.58, \rho_{l\infty} = 0.08, k_l = 0.2, \delta_l = 0.7, \delta = 0.7, \rho_{z0} = 1.25, \rho_{z\infty} = 0.04, k_z = 0.3, \delta_z = 1 \) and \( \delta_z = 0.4 \).

The simulation results of the proposed controller compared with benchmark controllers are presented in Figures 4.3-4.4. The proposed design works successfully; the tracking errors remain within the intended bounds while the attitude dynamics model uncertainties are compensated. As seen in Figure 4.4, it is verified that PPB based error transformation gives flexibility in tuning transient and steady-state behaviors of the nonlinear system. The maneuvering, tracking and parameter estimation details with the proposed controller are shown in Figures 4.5-4.7. Next, the alternative direct adaptive controller in Section 4.6 is simulated and compared with the proposed control scheme in Section 4.4.2 in Figures 4.8. Both design methods work effectively to satisfy attitude tracking.

4.7.3 Experimental Tests

In experiments, a sinusoidal altitude trajectory is considered as
\[
p_d(t) = [0, 0, 0.5 + 0.2 \sin(0.05t)]^T,
\]
with the initial position \( p(0) = [0, 0, 0.2]^T \). The PPB parameters are specified as \( \rho_{l0} = 2.3, \rho_{l\infty} = 0.3, k_l = 0.2, \delta_l = 0.9, \delta = 1, \rho_{z0} = 2.3, \rho_{z\infty} = 0.3, k_z = 0.2, \delta_z = 1 \) and \( \delta_z = 0.9 \).

Transient and steady-state tracking errors with the proposed controller, compared with the benchmark controllers, are presented in Figure 4.9. The video of the experimental tests is presented in URL: https://www.youtube.com/watch?v=KsILES17DDk. The proposed controller is observed to guarantee bounded error performances with adjustable error convergence rates. The error convergence rate is clearly improved around 20 [sec] using tuning flexibility of the proposed control design. The experimental results of the proposed controller are shown in Figures 4.10-4.13, indicating high tracking performance and robustness to modeling uncertainties. While sinusoidal altitude motion is performing, lateral tracking is freely worked within the specified bound 0.3 [m] as shown in Figure 4.10.

Two different motion cases are tested at the fixed-level altitude to investigate hovering performance of the proposed indirect adaptive control design. The results of the circular
Figure 4.3: Spiral trajectory tracking errors of the proposed control design, compared with the benchmark controllers.

trajectory tracking test are shown in Figures 4.14-4.17 and the video of the circular hovering experiment is presented in URL: https://www.youtube.com/watch?v=fuTXh-JeRyk. The results of the square-waypoint tracking test are shown in Figures 4.18-4.21, and the video of the square hovering experiment is presented in URL: https://www.youtube.com/watch?v=G1x4r-hDU48. The proposed controller works successfully for both motion cases.
Figure 4.4: Transient error performances for 0-50[sec].

Figure 4.5: Spiral motion with proposed control design.
Figure 4.6: Attitude tracking of the proposed control design.

Figure 4.7: LS based estimation $\hat{\theta}_\phi$ of $\theta^*_\phi$. 
Figure 4.8: Tracking errors with the proposed indirect adaptive control design Section 4.4.2 and the direct adaptive control in Section 4.6.

Figure 4.9: Altitude tracking errors of the proposed control design, compared with the benchmark controllers.
Figure 4.10: Trajectory tracking of the proposed control design.

Figure 4.11: Attitude tracking of the proposed control design.
Figure 4.12: LS based estimation \( \hat{\theta}_\varphi \) of \( \theta^*_\varphi \).

Figure 4.13: Motor PWM inputs \( v_r \) for the proposed control design.
Figure 4.14: Circular hovering test motion.

Figure 4.15: Trajectory tracking for the circular hovering test.
Figure 4.16: Mean Square tracking error for the circular hovering test.

Figure 4.17: LS based estimation $\hat{\theta}_\varphi$ of $\theta^*_\varphi$ for the circular hovering test.
Figure 4.18: Square hovering test motion.

Figure 4.19: Waypoint tracking for the square hovering test.
Figure 4.20: Mean Square tracking error for the square hovering test.

Figure 4.21: LS based estimation $\hat{\theta}_\varphi$ of $\theta^*_\varphi$ for the square hovering test.
4.8 Summary and Remarks

Robust adaptive motion control of quadrotor UAVs with guaranteed tracking error performance has been studied and tested on a Qball-X4 quadrotor testbed. The proposed control schemes have been developed for position (lateral position and altitude) and attitude dynamics, separately. The position control schemes have been designed utilizing prescribed performance bound (PPB) based error transformation and backstepping techniques. These control schemes have ensured bounded trajectory tracking error with tunable transient and steady-state behaviors. Effects of nonlinearities and model (inertia and drag) uncertainties in attitude dynamics are compensated making the control design adaptive via use of LS based PI algorithms. The overall stability and convergence of the closed-loop system have been proved. The effectiveness of the proposed design has been verified via simulations and experiments.
Chapter 5

Adaptive Mixing Formation Control of a Multi-UAV System

5.1 Introduction

In formation, several architectures have been studied e.g., decentralized (distributed) vs. centralized, hierarchical vs. non-hierarchical and symmetric vs. asymmetric \[22, 53, 89\] as high-level designs. Hierarchical architectures are more practical and better suited for real-time implementation in multi-UAV systems without long-range sensors. In this chapter, a distributed formation control scheme is designed for a system of \(N \geq 4\) quadrotor UAVs in the asymmetric and hierarchical (leader-follower) structure with leader, first, second and ordinary followers, for a robust maintenance of predefined formation geometry, utilizing tools of rigid graph theory \[12, 163\] for performing cohesive motion in 3D.

Formation control literature works mostly use simple dynamical models such as single-integrator, point-mass (double integrator) or kinematic UAV model \[11, 56, 70\]. A single integrator model based formation control design is presented in \[82\] for a group of quadrotor UAVs in 2D. \[83\] extends the study to a global convergence in formation. In both studies, the authors do not deal with low-level control design of the quadrotor UAVs and its performance effects on the formation maintenance and robustness. In \[89\], an adaptive formation control of a multi quadrotor UAV system considering the realistic dynamic modeling is studied for parametric uncertainties. The quadrotor UAV dynamics is considered in three separated sub-models: reference angle, altitude, and attitude dynamics to make more proper and effective low-level control analyses, separately. In this chapter, the main contribution is to design an adaptive mixing controller (AMC) to enhance tracking perfor-
mance and robustness at the low-level while using rigid graph theory tools for formation maintenance of a multi quadrotor UAV system in 3D.

The quadrotor UAV dynamics consists of highly coupled states, aerodynamic coefficients, disturbances, and uncertainties. To compensate these effects, more advanced control designs are required. In the literature, there exist proportional-integral-derivative (PID) and linear quadratic (LQ) control based classical studies for simplified and linearized dynamics [30, 36] as well as advanced control schemes [25, 27, 39, 90, 103] that use nonlinear effects by utilizing nonlinear control techniques such as feedback linearization, sliding mode and backstepping control methods.

To compensate the effects of parametric uncertainties, there exist two main adaptive control approaches: direct and indirect [74]. Indirect approaches calculate controller gains using estimated system parameters at each instant time [33, 46, 89], while direct methods update controller parameters directly [46, 53]. In this chapter, we develop indirect adaptive controllers using the least squares (LS) based parameter identification (PI) algorithm to cancel the effects of inertial uncertainties in the attitude of quadrotor UAVs. This approach gives us some advantages, e.g. avoiding negative effects of adaptation time on control performances. By this approach, control performances can be adjusted easily. An integral state is also added into the controllers for dealing with disturbances which come from ignored terms, e.g. drag and Coriolis.

Although there are many theoretical accomplishments and successful applications of PI based indirect adaptive control designs, this area still needs an effort to solve difficulties in robustness performance [92]. Since PI algorithms need a time to converge uncertain parameters to actual values, indirect adaptive control schemes perform poorly for transient tracking performance with overshoot/undershoot and more chattering during the convergence time. By this motivation, as the main contribution of the chapter, the AMC scheme is designed to improve the individual tracking of each quadrotor UAV by providing smoother control action as well as formation maintenance performance. In the proposed scheme, the multiple model adaptive control approach [16, 32, 33, 92] is used for quadrotor UAVs. Based on inertial changes, the proposed scheme is aimed to blend outputs of a set of controllers, each of which is pre-designed to provide desirable stability and performance properties for a certain parametric setting of the system environment. Since mixing strategies provide a smooth transition between control gains based on the estimated parameters, the AMC scheme aims to increase the robustness of tracking performances and formation maintenances. The proposed AMC is compared with the adaptive linear quadratic controller (ALQC) design. In both cases, PD controller is used for reference angle generation and PID controller is used for the altitude model to complete the motion of the multi quadrotor UAV system.
5.2 Quadrotor UAV Dynamics

The full nonlinear dynamic model of quadrotor UAV motion dynamics (2.10) is presented in Chapter 2. Then, the nonlinear dynamics is parted into the lateral position (reference angle), the altitude and the attitude dynamics. In this chapter, we have reconsidered the sub-models to obtain their separate linear models as following:

a) Reference Angle Dynamics: The reference angle $\phi_d$ is defined for the quadrotor UAV to drive on the desired lateral position $\dot{p}_d = [p_{xd}, p_{yd}]^T$ by using the dynamics as

$$\dot{p}_l = v_l, \quad \dot{v}_l = \frac{4T_z}{m} f_{ul}, \quad f_{ul} \triangleq \begin{bmatrix} \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \end{bmatrix} \approx \begin{bmatrix} \dot{\theta} \\ -\dot{\phi} \end{bmatrix} \approx \begin{bmatrix} \dot{\theta}_d \\ \dot{\phi}_d \end{bmatrix},$$

where the approximation is valid for small angles $\theta, \phi, \psi$.

b) Altitude Dynamics: The altitude dynamics is

$$\dot{p}_z = v_z, \quad \dot{v}_z = \frac{4}{m} (\cos \phi \cos \theta) T_z - g, \quad \dot{T}_z = -b T_z + K b u_z.$$  

Since the rotational angles are close to zero by the small angle approximation, the linearized altitude model is obtained as

$$\ddot{p}_z = \left( \frac{4Kb}{m(s+b)} \right) u_z - g.$$  

(5.6)

c) Attitude Dynamics: Consider the attitude dynamics from (2.10), and for further design, let us redefine the attitude dynamics properly for controllable canonical form as

$$\dot{\phi} = w_\phi, \quad \dot{w}_\phi = \varsigma_\phi, \quad \dot{\varsigma}_\phi = -b \varsigma_\phi + K b \frac{\sigma_\phi}{f_\phi} u_\phi, \quad \phi \in \{\phi, \theta, \psi\}. $$  

(5.9)
where $ς_ϕ$ is the fictitious state. For the disturbance rejection, we also augment an integrator to the attitude dynamics. Then, we write the state-space form of the attitude dynamics as follows:

$$\dot{x} = Ax + Bu_ϕ + d, \quad (5.10)$$

where $x = \left[\int (ϕ - ϕ_d), (ϕ - ϕ_d), \dot{ϕ}, ς_ϕ\right]^T$,

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -b \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \sigma_ϕ K_b \end{bmatrix}, \text{ and } d = \begin{bmatrix} 0 \\ -\dot{ϕ}_d \\ 0 \\ 0 \end{bmatrix}$$

with assuming that $\dot{ϕ}_d : \mathbb{R} \rightarrow \mathbb{R}$ is bounded since there exists a desired angular velocity limit $\varepsilon$ such that $|\dot{ϕ}_d(t)| \leq \varepsilon$.

### 5.3 Rigid Graph Modeling of the Multi-UAV System

In this chapter, a 3D asymmetric, hierarchical (leader-follower) formation architecture [20, 163] is considered as illustrated in Figure 5.1 for a multi quadrotor UAV system $S$ that
consists of $N$ vehicles indexed as $V_1, \ldots, V_N$, where the position of each vehicle $V_i$ is denoted as $p_i(t) = [x_i(t), y_i(t), z_i(t)]^T \in \mathbb{R}^3$. The sensing and distance constraint network within the system $S$ is represented by a directed underlying graph $G_S = (V_S, E_S)$ [20, 53, 163], where each quadrotor UAV $V_i$ is represented by a vertex $i \in V_S$ and each directed edge $(i, j) \in E_S$ represents a sensing and distance constraint link from $V_i$ to $V_j$, indicating that $V_i$ senses its distance $d_{ij} = \|p_i - p_j\|$ from $V_j$ and is required to keep this distance at a pre-defined desired value of $d_{ij}^*$. The 3D formation is represented by $F_S = (S, G_S, D_S)$ as a combination of the multi quadrotor UAV system $S$, the underlying directed graph $G_S = (V_S, E_S)$ and the desired distance set $D_S = \{d_{ij}^* | (i, j) \in E_S\}$.

In the high-level (formation) control design of this chapter, the notion of persistence [20, 163] is used to accommodate cohesive motion. A 3D formation is called rigid if the distance $d_{ij}$ between corresponding quadrotor UAV pair $(V_i, V_j)$ remains constant during any formation scenario. If each quadrotor UAV satisfies the distance constraint in the formation, it is called constraint consistent. If a formation satisfies both rigidity and constraint consistence, it is called persistent. For the formation $F_S = (S, G_S, D_S)$ to be persistent in 3D, the underlying graph $G_S = (V_S, E_S)$ needs to have at least $|E_S| = 3|V_S| - 6$ edges, in which case $F_S$ is called minimally persistent. Formal definitions of rigidity, constraint consistence and persistence are presented with more details in [163].

5.4 Problem Statement

Given a system $S$ of $N \geq 4$ quadrotor UAVs in a predefined persistent formation $F_S$ and a desired trajectory to follow, the high-level (formation) control objective is to maintain the predefined formation during trajectory maneuvering, and, for each quadrotor UAV $V_i$, the low level (individual) control objective is to guarantee accurate trajectory tracking, robustly to inertial uncertainties and disturbances, under the following assumptions:

**Assumption 5.4.1.** Reference way-points are known for the leader $V_1$ as a sequence of $M$ way-points:

$$p^k_r \in \{p^1_r, \ldots, p^M_r\}. \quad (5.11)$$

**Assumption 5.4.2.** Each quadrotor UAV $V_i$ can measure the position $p_i$ of itself in $O_g$ and the relative position $R_{ij} = (p_j - p_i)$ of each neighbor agent $V_j$, for which $(i, j) \in E_S$ and sensing $d_{ij}$.
Assumption 5.4.3. Desired relative positions of first $V_2$, second $V_3$ and ordinary $V_i$ followers are known as $R_{(1,2)}^*$, $R_{(1,3)}^*$, $R_{(2,3)}^*$ and $R_{(j,i)}^*$, $j \in \{i-1, i-2, i-3\}$ for establishing $d_{ij}^*$ and formation geometry.

Next, the corresponding formal control problem is stated.

Problem 5.4.1. Consider a system $S$ of $N \geq 4$ quadrotor UAVs moving in $\mathbb{R}^3$, with agent motion dynamics (5.2), (5.6), (5.10) and let $F_S = (S,G_S,D_S)$ be a 3D minimally persistent formation in leader-follower structure. Design a distributed formation control scheme in the two-level as following:

1. At the formation level, design an on-line distributed supervisor $P_s$, under Assumptions 5.4.1-5.4.3, to generate the desired positions $p_{di}(t)$. Then, design a PD controller to generate desired angles $\varphi_{di} = (\phi_{di}, \vartheta_{di}, \psi_{di})$.

2. At the individual level, for attitude and altitude dynamics, design control laws to generate the control signals $\hat{u}_{zi}^*(t)$ and $u_{zi}(t)$ such that the position $p_i(t)$ tracks the desired trajectory $p_{di}(t)$ generated at the formation level while maintaining desired distances $d_{ij}^*$ and keeping the predefined formation shape without deforming during trajectory maneuvering.

5.5 Distributed Control Design

High (formation) and low (individual) levels of the control scheme to address Problem 5.4.1 are designed as described in the following two subsections.

5.5.1 High-level Control Design

The high-level module $H_i$ of each agent $V_i$’s controller within the proposed distributed control scheme consists of two submodules as shown in Figure 5.2:

On-line Distributed Formation Supervisor

Desired trajectory $p_{di}(t)$ of $V_i$ is generated on-line by this submodule, depending on whether $V_i$ is the leader ($i = 1$), first or second follower ($i = 2$ or $i = 3$), or ordinary follower ($i \geq 4$), as follows:
Figure 5.2: The Overall Control Structure for $i$th quadrotor UAV.
**Leader:** $V_1$ is responsible for tracking the predefined way-points defined in (19). To address this task, Algorithm 1 is used to generate $p_{d1}(t)$ at each time instant $t$.  

**Algorithm 1**  Leader Way-point Based Trajectory Generation

1. Set $p_{d1}(0) = p^1_r$; $k = 1$
2. At each time instant $t > 0$:
   - $p_{d1}(t) = p^k_r$.
   - if $k < M$ and $\|p_1(t) - p^k_r\| \leq \delta$,
   - $k = k + 1$.

**First Follower:** $V_2$ only follows the leader. Its desired trajectory $p_{d2}(t)$ is generated as

$$p_{d2}(t) = [p_1(t) + R^*_r(1,2)]. \quad (5.12)$$

**Second Follower:** $V_3$ follows the leader and first follower. Its desired trajectory $p_{d3}(t)$ is generated as

$$p_{d3}(t) = \frac{1}{2} [p_1(t) + R^*_r(1,3) + p_2(t) + R^*_r(2,3)]. \quad (5.13)$$

**Ordinary Followers:** $V_i, \ i = 4,..,N$ follows previous three neighbors. The desired trajectory $p_{di}(t)$ of $V_i$ is obtained as

$$p_{di}(t) = \frac{1}{3} \sum_j [p_j(t) + R^*_r(j,i)]. \quad (5.14)$$

Note that the sensed and the desired distance sets among quadrotor UAVs are obtained by the relative and the pre-defined relative position knowledge under Assumption 5.4.2-5.4.3. To achieve cohesive motion for the leader-follower formation as discussed in Section 5.3, first, second and ordinary followers need to sense leader, leader and first follower, and previous three neighbors, respectively. Thus, these required interactions are provided via first (5.12), second (5.13) and ordinary (5.14) follower formation rules. Moreover, the above formation rules are derived to increase the robustness of the leader - follower formation when the number of ordinary followers enlarges.
Reference Angle Generation

Using desired position information, desired angles \( \phi_{di} = [\psi_{di}, \varphi_{di}, \psi_{di}]^T \) is generated by a PD controller before the low-level scheme as

\[
f_{li} = K_{Pli}(p_{ldi} - p_{li}) + K_{Dli}(\dot{p}_{ldi} - \dot{p}_{li}).
\]  

(5.15)

It is considered that yaw motion does not affect directly the lateral motion of quadrotor UAVs. Thus, assumed that the desired yaw angle is \( \psi_{di} = 0 \) for \( \forall t \). Then, \( \phi_{di} = [f_{li}, \psi_{di}]^T = [\vartheta_{di}, \phi_{di}, \psi_{di}]^T \). Since the derivative term of (5.15) has high frequency, this action can generate large control input variations in high-frequency error signals. To limit high-frequency gain, a low pass filter \( (F(s) = 1/Ts + 1) \) is used as seen in Figure 5.3 where \( T \) is the filter time constant.

5.5.2 Low-level Control Design

The low-level control scheme \( (L_i) \) is considered in two subparts which are the altitude \( (C_{Pi}) \) and the attitude \( (C_{Ai}) \) as seen in Figure 5.2. The low-level control design is responsible for tracking and stabilizing of \( i \)th quadrotor UAV in the altitude by PID and the attitude by ALQC / AMC. These sub-control modules generate the effective control inputs of \( u_{zi} \) and \( \hat{u}_{\phi_i} \), and then by mixing the control inputs via (2.5), the motor control input \( v_{ri} \) is produced. Both operating frameworks are stated as follows.
Altitude Control Design (PID)

Altitude controller \(C_{Pi}\) is designed to succeed longitudinal tracking of \(i\)th quadrotor UAV. By the altitude model (5.6), a PID control law is used for \(V_i\) as

\[
u_{zi} = K_{Pzi}e_{zi} + K_{Izi} \int_0^t e_{zi} dt + K_{Dzi} \dot{e}_{zi}.
\]

(5.16)

where \(e_{zi} = (p_{zdi} - p_{zi})\).

Attitude Control Design

Adaptive attitude control schemes are designed based on the indirect approach. First, an LS based PI algorithm is separately developed to estimate the unknown inertias as shown in 5.4. Then, a mixing-based adaptive controller is designed to improve individual tracking and robustness as well as the formation performance of \(i\)th quadrotor UAV. An ALQC scheme is also designed to compare the formation performances. These designs are analyzed with more details in the sections 5.6 and 5.7.

5.6 On-line Parameter Identification

To overcome the inertial uncertainties in the attitude (5.10), an on-line LS based PI algorithm is utilized with ALQC and AMC schemes as shown in Figure 5.4. Following [74], we first form a linear parametric model, then design the LS algorithm. Consider (5.10) for \(i\)th quadrotor UAV in the form of transfer function as

\[
s^2 \varphi_i = \frac{\sigma_{\varphi} K b}{J_{\varphi i} (s + b)} u_{\varphi i},
\]

(5.17)

**Parametric Model:** The parametric model is defined for (5.17) avoiding need for signal differentiation and the associated noise sensitivity issue by use of the stable filter \(\frac{1}{(s+\lambda)^2}, \lambda > 0\), as

\[
z_{\varphi i} = \theta^*_i \varphi_i,
\]

\[
z_{\varphi i} = \frac{s^2}{(s + \lambda)^2} \varphi_i, \quad \theta^*_i = \frac{1}{J_{\varphi i}}, \quad \Phi_{\varphi i} = \frac{\sigma_{\varphi} K b}{(s + \lambda)^2(s + b)} u_{\varphi i},
\]

92
where Euler angles \( \phi_i \) and control signals \( u_{\phi_i} \) are measurable, and \( K, b, \) and \( \sigma_\phi \) are constant parameters, respectively.

**Recursive LS:** To generate the estimate \( \hat{\theta}_i \) of the uncertain inertia parameter \( \theta_i^* \), we apply the recursive LS algorithm \([74]\) with forgetting factor to (5.18) as

\[
\dot{\hat{\theta}}_i(t) = Pr(\Phi_{\phi_i}P(t)e(t)) = \begin{cases} P(t)e(t)\Phi_{\phi_i}(t), & \text{if } \theta_i < \hat{\theta}_i < \theta_i^* \\ 0, & \text{otherwise} \end{cases},
\]

\[
\dot{P}(t) = \begin{cases} \beta P(t) - \frac{\Phi_{\phi_i}(t)^2}{m^2_i(t)} P^2(t), & \text{if } \theta_i < \hat{\theta}_i < \theta_i^* \\ 0, & \text{otherwise} \end{cases},
\]

\[
e(t) = \frac{z_{\phi_i}(t) - \hat{z}_{\phi_i}(t)}{m^2_i(t)}, \quad m^2_i(t) = 1 + \Phi_{\phi_i}(t)^2,
\]

where \( P(t) \in \mathbb{R} \) is the positive covariance term with \( P(0) > 0 \), \( m_\phi(t) \) is the normalizing signal, \( \beta \) is the forgetting factor and \( e(t) \) is the estimation error. Assumed that the upper and lower bounds of \( \theta_i^* \) are known, i.e. \( \theta_i \geq \theta_i^*(t) \geq \theta_i > 0 \). \( Pr(.) \) is the projection operator which maintains \( \hat{\theta}_i(t) \in [\theta_i, \theta_i^*] \), for all \( t \geq 0 \).

**Lemma 5.6.1 (Stability and Convergence).** Consider the LS based PI algorithm, applied to the attitude dynamics (5.17). It is guaranteed that all the signals in (5.19), including \( P \) and \( P^{-1} \), are bounded and \( \hat{\theta}_i(t) \in [\theta_i, \theta_i^*] \). Further, if \( \Phi_{\phi_i}^2 = \frac{\Phi_{\phi_i}^2}{m^2_i} \) persistently exciting, i.e. if \( \frac{1}{T} \int_0^T \Phi_{\phi_i}^2 d\tau \geq \alpha_0 \) for all \( t > 0 \) and some \( T, \alpha_0 > 0 \), then (5.19) ensures that \( \theta_i(t) \to \theta_i^* \) as \( t \to \infty \). The convergence of \( \theta_i(t) \to \theta_i^* \) is exponential for \( \beta > 0 \).
Proof. The result is a direct corollary of the more general Theorem 3.7.4 and 3.10.1 in [74].

5.7 Adaptive Attitude Control Laws

Two particular adaptive control laws, which are ALQC and AMC schemes, are studied based on the designed PI algorithm (5.19) and the infinite-time LQR [10] for the attitude model (5.10). Both control laws base on the LQR design as a nominal controller. Hence, the nominal LQR procedure is firstly presented as follows:

Let consider the quadrotor UAV dynamics from (5.10) as
\[
\dot{x} = Ax + Bu_\varphi,
\]
where \( A \in \mathbb{R}^{4 \times 4} \) is a matrix, and \( B \in \mathbb{R}^4 \) is a matrix. Then, design a nominal LQR for minimizing performance measurement of (5.20) with the cost function
\[
J = \int (x^T Q x + R u_\varphi^2) dt,
\]
where \( Q \in \mathbb{R}^{4 \times 4} \) is a positive definite matrix and \( R \in \mathbb{R} \) is a positive scalar. We calculate the optimal gain \( K_c \in \mathbb{R}^{1 \times 4} \) as
\[
K_c = R^{-1} B^T P,
\]
where \( P = P^T > 0 \in \mathbb{R}^{4 \times 4} \) is an auxiliary matrix calculated by solving the Riccati equation as
\[
A^T P + PA - PBR^{-1} B^T P + Q = 0.
\]
After obtaining \( K_c \), the state-feedback control law is
\[
u_\varphi = -K_c x.
\]
So that the closed-loop nominal LQR control law becomes
\[
\dot{x} = (A - BK_c)x.
\]

Lemma 5.7.1 (Nominal LQR Stability). Consider the nominal LQR design procedure (5.21)-(5.24) for the dynamics (5.20). Then, the closed-loop system (5.25) is asymptotically stable.
Proof. Under Assumption 3.2.1 from (pp.44-45) of [10], Lemma from (pp.46) of [10] guarantees that $P$ is positive definite since $(A, D)$, where $D$ is any matrix such that $DD' = Q$, is completely observable and $(A, B)$ is stabilizable for the parameters: $b, K, \sigma, J_\phi > 0$. So that the closed-loop nominal LQR control law (5.25) is asymptotically stable.

5.7.1 Adaptive Linear Quadratic Control (ALQC)

In this section, adaptive linear quadratic control (ALQC) is designed for $i$th quadrotor UAV as seen in Figure 5.4 based on the nominal LQR design with the PI algorithm. Since the input matrix $B_i$ includes inertial uncertainties, the LS based PI algorithm (5.19) is derived to estimate $\hat{\theta}_i$ of $\theta_i^*$ for all $t \geq 0$. By combining the estimate $\hat{\theta}_i$ with the LQR design (5.21)-(5.24), the time-varying state-feedback control law is obtained as

$$\hat{u}_{\phi i}^* = -\hat{K}_{ci}x_i. \quad (5.26)$$

where $\hat{K}_{ci}$ is the on-line optimal control gain matrix.

Lemma 5.7.2 (ALQC Stability). Consider the attitude dynamics (5.20) for $i$th quadrotor UAV and design ALQC by following the nominal LQR steps (5.21)-(5.24) with the estimation $\hat{\theta}_i(t)$ (5.19) of $\theta_i^*(t)$. The adaptive optimal control law (5.26) is asymptotically stable.

Proof. The control law (5.26) is asymptotically stable by Lemma 5.6.1 and Lemma 5.7.1 with guaranteed $\bar{\theta}_i \geq \hat{\theta}_i(t) \geq \theta_i > 0$, for all $t \geq 0$.

5.7.2 Proposed Adaptive Mixing Control (AMC)

This section presents an adaptive mixing control (AMC) approach applied the attitude of $i$th quadrotor UAV. The proposed AMC scheme uses a blending strategy between candidate controllers calculated based on the nominal LQR while it deals with overcoming inertial uncertainties as well as disturbance and sensor noises. The detail of the AMC approach is discussed in earlier studies [16, 33, 92]. In this section, the AMC procedure [92] is firstly introduced for a linear SISO plant which contains unknown parameters, disturbance and
sensor noise as,

\[ y = G(s, \theta^*)u + d, \quad (5.27) \]
\[ G(s, \theta^*) = G_N(s, \theta^*) (1 + \triangle_m(s)), \quad (5.28) \]
\[ G_N(s, \theta^*) = \frac{N(s, \theta^*)}{D(s, \theta^*)}, \quad (5.29) \]
\[ y_m = y + \nu, \quad (5.30) \]

where \( y_m \) is measured system output of \( y \), \( \theta^* \) is unknown plant parameter, \( G_N(s, \theta^*) \) is the transfer function of the nominal plant with unknown parameters, \( \triangle_m(s) \) is the multiplicative model uncertainty, \( d \) is the disturbance and \( \nu \) is sensor noise on the system. The control objective is to regulate the output \( y \) to zero. In the AMC design, to accomplish the objective, there are four conditions that have to be satisfied \([16, 92]\) as follows:

1. Interval of the unknown parameter \( \theta^* \) is known,
2. \( G_N(s, \theta^*) \) is strictly proper,
3. \( \triangle_m(s) \) is proper and analytic in \( \text{Re}(s) \geq -\frac{\delta_0}{2} \),
4. \( D(s, \theta^*) \) is a monic polynomial with known degree.

Then, we consider the state-space realization of (5.28) for further state-space based control design

\[
\dot{x}_m = A_m x_m + B_m u_m, \quad (5.31) \\
y_m = C_m x_m + d + \nu, \quad (5.32)
\]

where \((A_m, B_m)\) is stabilizable and \((C_m, A_m)\) is detectable to satisfy the control objective. Thus, under conditions 1-4, the proposed AMC scheme is developed considering (5.20) in the form of (5.31) for \( i \)th quadrotor UAV model. The AMC scheme consists of three main steps as illustrated in Figure 5.4: Adaptive law (LS scheme), Candidate controllers (developed off-line sets and used in multi-controller), and Mixing scheme as follows.

**Adaptive law:** Considering the adaptive mixing control literature, the gradient algorithm has been used in \([16, 92]\) with parameter projection. Another method, the LS based adaptive law has been studied in \([33]\). Since the LS algorithm is less affected by the noise and inaccuracies in the observed data \([74]\), the LS algorithm with parameter projection is used in the proposed AMC design. The LS based PI scheme has already designed in
Therefore, it performs separately and serves the estimation $\hat{\theta}_i$ to candidate controllers and mixing strategy.

**Candidate controllers:** Composing candidates and transition regions between each other, we define $n$ subsets based on a priori knowledge on bounds of the uncertain parameter $\theta_i^*$

$$\Omega_i = (\underline{\theta}_i, \bar{\theta}_i] = \{\theta_i^* \in \Omega_i \mid \underline{\theta}_i < \theta_i^* \leq \bar{\theta}_i\},$$  \hspace{1cm} (5.33)

where assumed that $\underline{\theta}_i$ and $\bar{\theta}_i$ are known. Then, $\Omega_i$ is separated into common $n$ subsets and let us define the subset range $\Omega_i^h$ for candidate controllers as

$$\Omega_i^h = [\theta_i^h, \bar{\theta}_i^h] \in \{\Omega_1^i, \Omega_2^i, \ldots, \Omega_n^i\}, \quad h = 1, 2, \ldots, n$$  \hspace{1cm} (5.34)

where $\Omega_i^h$, $\theta_i^h$ and $\bar{\theta}_i^h$ are the pre-defined range, the lower and upper bounds for each subset, respectively. $\Omega_i^h$ is chosen that the intersection of each transition subset pair is non-empty $\Omega_i^h \cap \Omega_i^{h+1} \neq \emptyset$. Hence, transition subsets satisfy the condition $\theta_i^{(h+1)} < \bar{\theta}_i^h$ where $h = [1, 2, \ldots, n - 1]$.

For each subset $\Omega_i^h$, a candidate controller is designed off-line as

$$u_i^h = C_i^h, \quad h = 1, 2, \ldots, n$$  \hspace{1cm} (5.35)

where $C_i^h$ is the candidate controller which meets the control objectives for $\Omega_i^h$. Therefore, the state-feedback controller law is obtained based on the nominal LQR steps (5.20)-(5.24) as

$$C_i^h: u_i^h = -K_{ci}^h x_i,$$  \hspace{1cm} (5.36)

where $K_{ci}^h$ is the state feedback gain vector.

Hence, the set of candidate controllers are considered as

$$\Lambda \triangleq \{C_i^h\}_{h \in \{1, 2, \ldots, n\}}.$$  \hspace{1cm} (5.37)

**Lemma 5.7.3** (Candidate Stability). Consider the set $\Omega_i$ of the known bounds of $\theta_i^*$ which consists of the union of $n$ subsets as

$$\Omega_i = \bigcup_{h=1}^n \Omega_i^h \subset (\underline{\theta}_i, \bar{\theta}_i].$$  \hspace{1cm} (5.38)

Choose a fixed control parameter as $\theta_i^{ho} = \left(\frac{\theta_i^h + \bar{\theta}_i^h}{2}\right)$ for each candidate subset where $h \in \{1, 2, \ldots, n\}$ as

$$\theta_i^{ho} \in \Omega_i^h \triangleq \{\theta_i^{ho} \in \mathbb{R} \mid \theta_i^h < \theta_i^{ho} < \bar{\theta}_i^h\}.$$  \hspace{1cm} (5.39)

97
Design each closed-loop control law (5.36), which is asymptotically stable for the fixed control parameter $\theta_i^{ho}$, in (5.37).

**Proof.** The stability of each control law (5.36) in (5.37) is directly satisfied based on the Lemma 5.7.1 and the fixed $\theta_i^{ho} > 0$. 

**Mixing scheme:** We now select and employ bump functions to provide a smooth switching between the candidate controllers. Bump functions are represented as $\eta^h_i \in \{\eta^1_i, \eta^2_i, \eta^3_i, ..., \eta^n_i\}$. A mixing function is selected for each candidate range as

$$\varrho_i(\hat{\theta}_i) = \left( \frac{\hat{\theta}_i - a^h_i}{b^h_i} \right), \quad (5.40)$$

where $a^h_i$ and $b^h_i$ are bump function parameters, and (5.40) defines a bump function as

$$\eta^h_i(\hat{\theta}_i) = \begin{cases} e^{\frac{-1}{1-(\varrho_i^h)^2}}, & \text{if } |\varrho_i^h| < 1 \\ 0, & \text{otherwise.} \end{cases} \quad (5.41)$$

Then, mixing performance gains are calculated as

$$\kappa_i^h(\hat{\theta}_i) = \frac{\eta_i^h(\hat{\theta}_i)}{\sum_{h=1}^n \eta_i^h(\hat{\theta}_i)}, \quad (5.42)$$

where $\kappa_i^h(\hat{\theta}_i) = 0$ if $\hat{\theta}_i \notin \Omega_i^h$.

Therefore, by the set of candidate controllers (5.37) and the mixing performance gains (5.42), the adaptive mixing control scheme is obtained as the following formula

$$\hat{u}^*_{\varphi h} = \sum_{h=1}^n \kappa_i^h(\hat{\theta}_i) u_i^h. \quad (5.43)$$

**Lemma 5.7.4 (AMC Stability).** Consider the AMC design (5.40)-(5.43) working with the PI (5.19). Select bump function parameters $a^h_i$ and $b^h_i$ for each properly selected candidate subset $\Omega_i^h$. So that the mixing performance gains are only activated for intersection of two neighbor candidate subsets as

$$\hat{\theta}_i \in \Omega_i^h \cap \Omega_i^{h+1} = [\theta_i^h, \theta_i^{h+1}] \cap [\theta_i^{h+1}, \theta_i^h] = [\theta_i^{h+1}, \theta_i^h]. \quad (5.44)$$
Then, the mixing control scheme becomes

\[ \hat{u}_{\varphi i}(\hat{\theta}_i) = -\left[ \kappa_i^h(\hat{\theta}_i)K_{ci}^h + \kappa_i^{h+1}(\hat{\theta}_i)K_{ci}^{h+1} \right] x_i \]  

(5.45)

where it is designed that \( \kappa_i^h + \kappa_i^{h+1} = 1 \). Thus, guarantee that the scheme (5.45) is Hurwitz stable when \( \hat{\theta}_i \in \Omega_i^h \cap \Omega_i^{h+1} \).

**Proof.** Consider the model (5.20) in form of the general controller canonical form where \( A \) and \( B \) matrices can be written as

\[
A = \begin{bmatrix}
0 & 1 & 0 & \ldots & 0 \\
0 & 0 & 1 & \ddots & \vdots \\
\vdots & \vdots & \ddots & \ddots & 0 \\
0 & 0 & \ldots & 0 & 1 \\
-a_m & -a_{m-1} & \ldots & -a_2 & -a_1
\end{bmatrix},
B = \begin{bmatrix}
0 \\
\vdots \\
0 \\
b_0
\end{bmatrix}.
\]

(5.46)

Then, characteristic equation of the closed-loop controller of (5.46) for \( i \)th quadrotor UAV with the estimation \( \hat{\theta}_i \) can be found as

\[ c_i(s) = (sI - A_i + \hat{B}_i K_{ci}) \]  

(5.47)

\[ c_i(s) = (s^m + a_1s^{m-1} + \cdots + a_{m0} + \hat{b}_{i0}(K_{ci}^m s^{m-1} + \cdots + K_{ci}^1)) \]  

(5.48)

By (5.45) and (5.48), we have the mixing characteristic equation \( C(s) \) when \( \hat{\theta}_i \in \Omega_i^h \cap \Omega_i^{h+1} \) as

\[ C(c_i^h, c_i^{h+1}) = (\kappa_i^h c_i^h(s) + \kappa_i^{h+1} c_i^{h+1}(s)) \]  

(5.49)

By Lemma 5.6.1 guaranteeing \( \hat{b}_{i0} > 0 \) \( \forall t \) and following Theorem 2.3 of [28], \( C(c_i^h, c_i^{h+1}) \) is Hurwitz stable since \( c_i^h \) and \( c_i^{h+1} \) are Hurwitz stable and if and only if

\[ eig(W) = eig(H^{-1}(c_i^h)H(c_i^{h+1})) \in (0, \infty], \]  

(5.50)

where Hurwitz matrix \( H(\cdot) \) associated with the polynomial \( h(s) \in C(c_i^h, c_i^{h+1}) \). The detail proofs are given in [28].
5.8 Real-time Testbed and Simulations

5.8.1 Real-time Testbed System

In the simulations, we consider that the multi quadrotor UAV testbed system is composed of Qball-X4 quadrotors developed by Quanser Inc. [172]. The Qball-X4 quadrotor system has been introduced in section 3.8.1

5.8.2 Adding the noise effect into the simulation

For more realistic simulation, we add Gaussian noises characterized by measuring Qball-X4 testbed’s sensors. The Qball-X4 is off-line switched on idle running for 20 [sec] to measure pitch rate from IMU as shown in Figure 5.5. By the measured data, the variance and mean value are calculated as

$$\rho = \frac{1}{r} \sum_{i=1}^{r} (w_\phi), \quad \text{and} \quad Var(w_\phi) = \frac{1}{r} \sum_{i=1}^{r} (w_\phi - \rho)^2$$

(5.51)

where $Var(w_\phi)$ is the variance value, and $\rho$ is the average of the noisy data. Then, the characterized noises are added to the attitude dynamics.
Table 5.1: The off-line calculated candidate controller gains

<table>
<thead>
<tr>
<th>Sets $\Omega_i^h$</th>
<th>Gains ($K_i^h$)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$k_1$</td>
<td>$k_2$</td>
<td>$k_3$</td>
<td>$k_4$</td>
</tr>
<tr>
<td>$\Omega_1^i$</td>
<td>0.018257418</td>
<td>0.066549024</td>
<td>0.029999841</td>
<td>0.641136268</td>
</tr>
<tr>
<td>$\Omega_2^i$</td>
<td>0.018257418</td>
<td>0.063712188</td>
<td>0.019879852</td>
<td>0.805576366</td>
</tr>
<tr>
<td>$\Omega_3^i$</td>
<td>0.018257418</td>
<td>0.062640956</td>
<td>0.016173044</td>
<td>0.92352064</td>
</tr>
<tr>
<td>$\Omega_4^i$</td>
<td>0.018257418</td>
<td>0.062034229</td>
<td>0.014101452</td>
<td>1.019855195</td>
</tr>
<tr>
<td>$\Omega_5^i$</td>
<td>0.018257418</td>
<td>0.061629867</td>
<td>0.012732009</td>
<td>1.102934544</td>
</tr>
<tr>
<td>$\Omega_6^i$</td>
<td>0.018257418</td>
<td>0.061335083</td>
<td>0.011739312</td>
<td>1.176831973</td>
</tr>
<tr>
<td>$\Omega_7^i$</td>
<td>0.018257418</td>
<td>0.06107543</td>
<td>0.010976320</td>
<td>1.243882949</td>
</tr>
<tr>
<td>$\Omega_8^i$</td>
<td>0.018257418</td>
<td>0.060924803</td>
<td>0.010365603</td>
<td>1.305577123</td>
</tr>
<tr>
<td>$\Omega_9^i$</td>
<td>0.018257418</td>
<td>0.060773698</td>
<td>0.009861993</td>
<td>1.362933702</td>
</tr>
<tr>
<td>$\Omega_{10}^i$</td>
<td>0.018257418</td>
<td>0.060645928</td>
<td>0.009372189</td>
<td>1.416686371</td>
</tr>
<tr>
<td>$\Omega_{11}^i$</td>
<td>0.018257418</td>
<td>0.060535963</td>
<td>0.009072189</td>
<td>1.467384069</td>
</tr>
</tbody>
</table>

5.8.3 Control Design Parameters

All control parameters are set for real-time based simulations and the Qball-X4’s parameters are taken as presented in Section 3.8.1. Sensor noises are characterized as $Var(w_\phi) = 0.0003$ and $\rho = 0.000721$ by (5.51). Reference angle PD control parameters are $K_{Pli} = 0.7$ and $K_{Dli} = 0.4$ and altitude PID control parameters are $K_{Pzi} = 0.006$, $K_{Izi} = 0.008$ and $K_{Dzi} = 0.002$. In the LS, forgetting factor, initial covariance and initial unknown parameter value are chosen $\beta = 0.1$, $P(0) = 10^5$ and $\theta_i(0) = 5$, respectively. In the nominal LQR, the ideal cost is determined by weightings as $Q = \text{diag}(150 \ 0 \ 20000 \ 25)$ and $R = 30000$. In the AMC, the number of candidates is selected as $n = 11$. The minimum and maximum bounds of $\theta_i^*$ are considered $\theta_i = 5 < \theta_i^* < \bar{\theta}_i = 77$. Then, the interval of subsets is taken $\Omega_i = [5, 77]$. $n$ subset candidates are assigned as $\Omega_1^i = [5, 10]; \Omega_2^i = [7, 17]; \Omega_3^i = [14, 24]; \Omega_4^i = [21, 31]; \Omega_5^i = [28, 38]; \Omega_6^i = [35, 45]; \Omega_7^i = [42, 52]; \Omega_8^i = [49, 59]; \Omega_9^i = [56, 66]; \Omega_{10}^i = [63, 73]; \Omega_{11}^i = [70, 77]$. Bump functions parameters selected for each subset as $a_i^h \in \{5, 12, 19, 26, 33, 40, 47, 54, 61, 68, 75\}$ and $b_i^h = 5$. The off-line calculated candidate gains are given in Table 5.1.
5.8.4 Simulation Results

Simulation results present the formation performances of the multi quadrotor UAV system with \( N=5 \) vehicles in leader-follower structure. The initial positions of quadrotor UAVs are \( p_1(0) = (9, 1, 0.2)^T \); \( p_2(0) = (8, -1, 0.2)^T \); \( p_3(0) = (8, 1, 0.2)^T \); \( p_4(0) = (7, 0, 0.2)^T \); \( p_5(0) = (7, -1, 0.2)^T \). Spiral motion way-points are generated by

\[
p_k(t) = (10 \cos(0.04t), 10 \sin(0.04t), 0.025t)^T
\]

where \( t = T_s k \) and the sampling time \( T_s = 0.05 \). Spiral motion scenarios are performed to maintain desired formation distance \( d_{ij}^* = 5 \; [m] \) and \( 5 \sqrt{3} \; [m] \) based on the desired relative positions:

\[
\begin{align*}
R_{(1,2)}^* &= (-\frac{5}{2}, -\frac{5}{2}\sqrt{3}, 0)^T, \\
R_{(1,3)}^* &= (-5, 0, 0)^T, \\
R_{(2,3)}^* &= (-\frac{5}{2}, \frac{5}{2}\sqrt{3}, 0)^T, \\
R_{(1,4)}^* &= (-\frac{15}{2}, -\frac{5}{2}\sqrt{3}, 0)^T, \\
R_{(2,4)}^* &= (-5, 0, 0)^T, \\
R_{(3,4)}^* &= (-\frac{5}{2}, -\frac{5}{2}\sqrt{3}, 0)^T, \\
R_{(2,5)}^* &= (-\frac{15}{2}, \frac{5}{2}\sqrt{3}, 0)^T, \\
R_{(3,5)}^* &= (-5, 0, 0)^T, \\
R_{(4,5)}^* &= (-\frac{5}{2}, \frac{5}{2}\sqrt{3}, 0)^T.
\end{align*}
\]

The formation performances of both control schemes are satisfied in terms of formation requirements as seen in Figures 5.6 and 5.7 as well as satisfying individual tracking performances of all quadrotor UAVs. Figure 5.13 presents \( \hat{\theta}_{\phi_1} \) of the leader for both scenarios. The PI algorithm works well and both \( \theta_{\phi_1} \) approach to ideal value after 50 [sec], but they do not converge to ideal value because of the noise. Although we can able to adjust the convergence time period for the estimation which is chosen a long time for this case study, we have designed the PI algorithm by selecting the parameters with small initial values of \( \theta_{\phi_1} \), forgetting factor \( \beta \), covariance \( \theta_0 \) to show the effectiveness of the proposed AMC design during the poor transient convergence period. Therefore, both control strategies work poorly at the convergence time period than their steady-state performances because of the poor transient estimation performance and initial positions of the quadrotor UAVs. After the estimation is converged around the ideal value after 50 [sec], the formation schemes work satisfactorily with small errors for maintaining the desired rigid distances as seen in Figure 5.8 and 5.9.

During the poor estimation performances at the beginning of simulations between 0 [sec] - 50 [sec], the ALQC design has bigger transient formation distance errors with overshoot, undershoot and chattering as seen clearly in Figures 5.8 and 5.10. This behavior is also based on the ALQC scheme’s control gains change sharply at each instant time. On the other hand, the proposed AMC scheme compensates these negative effects as seen in 5.9 and 5.11 and it eliminates chattering effects and large transient errors on the control design because of providing smooth blending for calculation of control gains at each instant time. During the AMC test, Figure 5.12 presents how to generate the smooth blending by using the bump function with the on-line estimation.
Figure 5.6: Spiral formation motion in 3D for ALQC.

Figure 5.7: Spiral formation motion in 3D for AMC.
Figure 5.8: Formation distances among quadrotor UAVs for ALQC.

Figure 5.9: Formation distances among quadrotor UAVs for AMC.
Figure 5.10: Mean Square tracking errors for ALQC.

Figure 5.11: Mean Square tracking errors for AMC.
Figure 5.12: Bump functions of candidate subsets for the estimation $\hat{\theta}_{\phi_1}$ of the leader quadrotor UAV.

Figure 5.13: $\hat{\theta}_{\phi_1}$ estimation of $\theta^*_{\phi_1}$ for the leader quadrotor UAV.
5.9 Summary and Remarks

In this chapter, a two-level, distributed formation scheme has been designed to keep persistently the formation shape of the multi quadrotor UAV system for the realistic quadrotor UAV dynamics. At the individual control, we have developed the AMC and compared with ALQC. In order to suppress the negative effects of inertial uncertainties in the attitude model, an online PI model is developed for both control schemes. The proposed AMC scheme have used to increase formation maintenance and robustness by using the switching method under uncertainties, disturbances, and noises. The ALQC has also been designed to compare both controllers and to show the effectiveness of the proposed AMC by the real-time based simulations. Performances of both adaptive formation control algorithms have been evaluated on realistic quadrotor UAV model by simulation tests. Both simulations are based the realistic model of the quadrotor UAV (developed from first principle dynamics as well as using data collected from real-time experiments), and we witness the high performance of the proposed AMC scheme. Successful formation results show the efficiency of the proposed AMC scheme.
Chapter 6

Optimal Tracking and Formation Control of Fixed-wing UAVs

In Chapters 3 and 5, we have proposed a linear quadratic tracking (LQT) control law for optimal attitude tracking of quadrotor UAVs and a distributed control law for formation maintenance of multi-quadrotor UAV systems. In this chapter, we extend these designs to lateral motion control of (Piccolo-controlled) small fixed-wing UAVs. We design LQT control schemes for trajectory tracking of the small fixed-wing UAVs and extend these designs to distributed formation control of a multiple fixed-wing UAV system. The surveillance tasks are simulated, and the main results are presented for the two cases: tracking and formation.

6.1 Lateral Motion Model of Small Fixed-wing UAV

The simplified nonlinear motion model (2.26) of fixed-wing UAVs is presented in Chapter 2. In this section, we simplify and partition (2.26) to obtain separate linear lateral model which is considered in the two sub-models as following:

a) Linear position model:

\[ \dot{x}_l(t) = A_l x_l(t) + B_l v_{ld}(t), \quad l \in \{x, y\}, \]
\[ p_l(t) = C_l x_l(t), \] 

(6.1)
where \( A_l = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{\alpha_v} \end{bmatrix} \), \( B_l = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \), \( C_l = \begin{bmatrix} 1 \end{bmatrix}^T \), and \( x_l = \begin{bmatrix} p_l \\ v_l \end{bmatrix} \). \( x_l \), \( v_l \), \( p_l \), \( v_l \) and \( \alpha_v \) represent states, control input, position, velocity, and inertial related dynamic parameter for the lateral model, respectively.

b) Linear heading model:

\[
\begin{align*}
\dot{x}_\psi(t) &= A_\psi x_\psi(t) + B_\psi w_c(t), \\
\psi(t) &= C_\psi x_\psi(t),
\end{align*}
\]

(6.2)

where \( A_\psi = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{\alpha_\psi} \end{bmatrix} \), \( B_\psi = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \), \( C_\psi = \begin{bmatrix} 1 \end{bmatrix}^T \), and \( x_\psi = \begin{bmatrix} \psi \\ w \end{bmatrix} \). \( x_\psi \), \( w_c \), \( \psi \), \( w \) and \( \alpha_\psi \) represent states, control input, heading, angular velocity and inertial related dynamic parameter for the heading model, respectively.

**Remark 6.1.1.** The dynamic systems have input constraints such that \( 0 < v_{\text{min}} < v_c < v_{\text{max}} \) and \( -w_{\text{min}} < w_c < w_{\text{max}} \) since the real-time fixed-wing UAV works in limited velocity and angular velocity.

### 6.2 Problem Statement for Fixed-level Motion

Consider a fixed-wing UAV and a system \( S \) of \( N \geq 3 \) fixed-wing UAVs moving in \( \mathbb{R}^2 \) with the lateral motion model (6.1) and (6.2). For an equilateral formation case of the UAV system \( S \) in 2D as discussed in section 5.3, let \( F_S = (S, G_S, D_S) \) be a 2D minimally persistent formation in leader-follower structure. Before the problem definition, the following assumptions are made.

**Assumption 6.2.1.** The states of the fixed-wing UAVs are measurable.

**Assumption 6.2.2.** The fixed-wing UAVs are held at the constant altitude for planar motion tasks. The desired path is known for tracking task of a single UAV and formation task of leader UAV \( V_1 \).

**Assumption 6.2.3.** Desired relative positions of first \( V_2 \) and ordinary \( V_i \) followers are known as \( R_{(1,2)}^* \) and \( R_{(i,j)}^* \), \( j \in \{i-1, i-2\} \) to establish \( d_{ij}^* \) and formation geometry.

**Assumption 6.2.4.** The fixed-wing UAVs are equipped with the autopilot avionic devices whose dynamics can be modeled as the simplified forms presented in Section 6.1.
Design a distributed formation control scheme and LQT control schemes as follows:

1. At the formation level, design an on-line distributed supervisor \((P_S)\) and a reference heading generator \((P_H)\) under Assumptions 6.2.1-6.2.3 to generate \(p_{di}(t) = [p_{xdi}(t), p_{ydi}(t)]^T\) and \(\psi_{di}(t)\) of \(V_i\).

2. Design the lateral position control unit to generate the command signal \(v_{ci}\) for feeding the autopilot under Assumption 6.2.4. Design an infinite-horizon LQT controller to generate an optimal control signal \(v_{ci}^*(t)\) so that \(p_{li}(t)\) tracks its desired trajectory \(p_{ldi}(t)\), minimizing the cost function for predefined quadratic performance optimal tracking and energy consumption

\[
J_{vi} = \frac{1}{2} \int_0^\infty (Qe_{li}^2(t) + Rv_{ci}^2(t))dt,
\]

where \(Q\) and \(R\) are positive constant weighting terms and

\[
e_{li}(t) = p_{ldi}(t) - p_{li}(t),
\]

is the lateral position tracking error.

3. Design the heading control unit to generate the command signal \(w_{ci}\) for feeding the autopilot under Assumption 6.2.4. Design an infinite-horizon LQT controller to generate the optimal attitude control signal \(w_{ci}(t) = w_{ci}^*(t)\) so that \(\psi_i(t)\) tracks its desired trajectory \(\psi_{di}(t)\), minimizing the cost function for predefined quadratic performance optimal tracking and energy consumption

\[
J_{wi} = \frac{1}{2} \int_0^\infty (Qe_{wi}^2(t) + Rw_{ci}^2(t))dt,
\]

where \(Q\) and \(R\) are positive constant weighting terms and

\[
e_{wi}(t) = \psi_{di}(t) - \psi_i(t)
\]

is the heading tracking error.

**Remark 6.2.1.** Note that we follow the leader UAV design \(i = 1\) of Problems 1-3 for trajectory tracking control of a single fixed-wing UAV.
6.3 High-level Control: Desired Trajectory and Heading Derivation

We now derive a distributed, hierarchical, asymmetric high-level controller structure to solve the control problems. The high-level module $H_i$ of each agent $V_i$’s controller within the proposed distributed control scheme consists of two submodules which generate the desired position and heading via the formation supervisor ($P_S$) and reference heading generator ($P_H$) as shown in Figure 6.1.

**On-line Distributed Formation Supervisor**

Desired trajectory $p_{di}(t) = [p_{xdi}(t), p_{ydi}(t)]^T$ of $V_i$ is generated on-line by this submodule, depending on whether $V_i$ is the leader ($i = 1$), first follower ($i = 2$), or ordinary follower ($i \geq 3$), as follows:

**Leader:** $V_1$ is responsible to track the pre-defined trajectory $p_{d1}(t)$ at each time instant $t$. 

Figure 6.1: Fixed-wing UAV formation control block diagram.
**First Follower:** $V_2$ only follows leader. Its desired trajectory $p_{d2}(t)$ is generated as

$$p_{d2}(t) = [p_1(t) + R_{(1,2)}^*]. \quad (6.7)$$

**Ordinary Followers:** $V_i$, $i = 3,..,N$ follows previous two neighbors. The desired trajectory $p_{di}(t)$ of $V_i$ is obtained as

$$p_{di}(t) = \frac{1}{2} \sum_j [p_j(t) + R_{(j,i)}^*]. \quad (6.8)$$

Note that the above formation rules are designed by following Section 5.5.1.

**Reference Heading Generation**

For $i$th fixed-wing UAV, we write the following control rule which always directs the heading angle to the desired position as

$$\psi_{di}(t) = \tan^{-1}\left(\frac{p_{ydi}(t) - p_{yi}(t)}{p_{xdi}(t) - p_{xi}(t)}\right). \quad (6.9)$$

### 6.4 Low-level Control: Optimal Linear Quadratic Tracking (LQT) Control Design

In this section, the LQT control scheme for lateral tracking of $i$th fixed-wing UAV is developed following generic LQT control design as established in Section 3.6.2. The approach is to apply the control laws (3.28), (3.30), (3.31) to the system models (6.1) and (6.2). Implementation of the control law (3.30) requires $\bar{P}$ from (3.28) and $\bar{g}(t)$ from (3.31). Note that a generic control analysis is presented for both the models (6.1) and (6.2) since their system matrices $A, B, C$ are same. Therefore, the RE, the approximate vector signal $\bar{g}(t)$ and the optimal control signal are obtained, respectively as

$$-\bar{P}A - A^T\bar{P} + \bar{P}BR^{-1}B^T\bar{P} - C^TQC = 0, \quad (6.10)$$

$$\bar{g}(t) = [A^T - PBR^{-1}B^T]^{-1}[-C^TQz(t)], \quad (6.11)$$

$$u^*(t) = -R^{-1}B^T\bar{P}x(t) + R^{-1}B^T\bar{g}(t), \quad (6.12)$$
where \( A = \begin{bmatrix} 0 & 1 \\ 0 & -\alpha \end{bmatrix} \), \( B = \begin{bmatrix} 0 \\ \alpha \end{bmatrix} \), \( C = \begin{bmatrix} 1 \\ 0 \end{bmatrix}^T \).

Solving (6.10) for \( \bar{P} = \begin{bmatrix} \bar{P}_1 & \bar{P}_2 \\ \bar{P}_2 & \bar{P}_3 \end{bmatrix} \in \mathbb{R}^{2\times2} \), we obtain

\[
0 = -\left(\alpha^2 \bar{P}_2^2 / R\right) + Q, \tag{6.13a}
\]
\[
0 = -\bar{P}_1 + \alpha \bar{P}_2 + \left((\alpha^2 \bar{P}_2 \bar{P}_3) / R\right), \tag{6.13b}
\]
\[
0 = 2 \bar{P}_2 - 2\alpha \bar{P}_3 - \left((\alpha^2 \bar{P}_3^2) / R\right), \tag{6.13c}
\]

Solving (6.11) for \( \bar{g}(t) = [\bar{g}_1(t) \; \bar{g}_2(t)]^T \in \mathbb{R}^2 \), we obtain

\[
\bar{g}_1(t) = \left[(\alpha + (\bar{P}_3 \alpha^2) / R)\right][((RQ)/(\bar{P}_2 \alpha^2))] z(t), \tag{6.14a}
\]
\[
\bar{g}_2(t) = \left[(RQ)/((\bar{P}_2 \alpha^2))] z(t). \tag{6.14b}
\]

### 6.5 Calculation of LQT Control Parameters

For both position and heading control models, the inertial related dynamics parameter, the error and the control weighting parameters are chosen same as \( \alpha = \alpha_l = \alpha_p = 100 \), \( Q = 200 \) and \( R = 1 \). Hence, the LQT control design as explained in detail in Section 6.4 is followed for a generic calculation of both models as follows.

The entry \( \bar{P}_2 \) of \( \bar{P}(t) \) is calculated by (6.13a) as

\[
\bar{P}_2 = \sqrt{(QR)/\alpha} = 1.4142. \tag{6.15}
\]

Then, by using (6.15) in (6.13c), the quadratic polynomial \( 10000\bar{P}_3 + 200\bar{P}_3 - 2.8284 = 0 \) is obtained as a function of \( \bar{P}_3 \), and solving the quadratic polynomial, the entry \( \bar{P}_3 \) is found as

\[
\bar{P}_3 = 0.0096. \tag{6.16}
\]

The remaining entry \( \bar{P}_1 \) of \( \bar{P} \) is calculated by using (6.15) and (6.16) in (6.13b) as

\[
\bar{P}_1 = 276.71. \tag{6.17}
\]
Table 6.1: Small fixed-wing UAV specifications used in simulations

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_{\text{min}}$ for all UAVs</td>
<td>Minimum angular velocity</td>
<td>-1.2 [rad/sec]</td>
</tr>
<tr>
<td>$w_{\text{max}}$ for all UAVs</td>
<td>Maximum angular velocity</td>
<td>1.2 [rad/sec]</td>
</tr>
<tr>
<td>$v_{\text{min}}$ for all UAVs</td>
<td>Minimum velocity</td>
<td>5 [m/sec]</td>
</tr>
<tr>
<td>$v_{\text{max}}$ for leader UAV</td>
<td>Maximum velocity</td>
<td>7 [m/sec]</td>
</tr>
<tr>
<td>$v_{\text{max}}$ for first follower UAV</td>
<td>Maximum velocity</td>
<td>7.5 [m/sec]</td>
</tr>
<tr>
<td>$v_{\text{max}}$ for ordinary follower UAV</td>
<td>Maximum velocity</td>
<td>8 [m/sec]</td>
</tr>
</tbody>
</table>

Therefore, for both models, $\bar{P}$ and the vector signal $\bar{g}(t)$ are obtained as follows:

$$
\bar{P} = \begin{bmatrix} 276.71 & 1.4142 \\ 1.4142 & 0.0096 \end{bmatrix}
\quad \text{and} \quad
\bar{g}(t) = \begin{bmatrix} 276.71 z(t) \\ 1.4142 z(t) \end{bmatrix},
$$

(6.18)

where $\bar{g}(t)$ represents $\bar{g}_i(t)$ for $z(t) = p_{id}(t)$ of the position control and $\bar{g}_\psi(t)$ for $z(t) = \psi_d(t)$ of the heading control.

**Remark 6.5.1.** After all, using (6.18) in (6.12), optimal position $v^*_{ch}(t)$ and heading $w^*_{ci}(t)$ control inputs for $i$th fixed-wing UAV are obtained as following:

$$
v^*_{ch}(t) = -R^{-1}B^T \bar{P}x_{hi}(t) + R^{-1}B^T \bar{g}_{hi}(t),
$$

(6.19)

$$
w^*_{ci}(t) = -R^{-1}B^T \bar{P}x_{\psi i}(t) + R^{-1}B^T \bar{g}_{\psi i}(t),
$$

(6.20)

### 6.6 Simulations and Results

The small fixed-wing UAV specifications used in simulations are presented in Table 6.1. After setting all control parameters from Section 6.5 with the sampling rate 200 [Hz] by using MATLAB/ Simulink®, we have simulated the proposed schemes for tracking control of the single fixed-wing UAV and formation maintenance control of the multiple fixed-wing UAV system. For both cases, surveillance missions by following the spiral reference trajectory generated by $p_{di}(t) = (3t \cos(\frac{t}{200} \cos(2\pi)), 3t \cos(\frac{t}{200} \cos(2\pi)))^T$ are simulated as follows.
Figure 6.2: Spiral surveillance motion of a fixed-wing UAV.

### 6.6.1 Simulation results of a single fixed-wing UAV

A numerical simulation is employed for a fixed-wing UAV with the initial position $p(0) = (-0.5, 0.1)^T$ and a spiral trajectory. All corresponding results are presented in Figures 6.2-6.6. The performances of the proposed control scheme satisfy the tracking requirements as seen in Figures 6.2-6.4.

### 6.6.2 Simulation results of a multiple fixed-wing UAV system

A numerical simulation is performed for a multi fixed-wing UAV system with $N=3$ vehicles in *leader-follower* structure. The initial positions of UAVs are $p_1(0) = (-0.5, 0.1)^T$; $p_2(0) = (-10, -60)^T$; $p_3(0) = (50, -30)^T$. Spiral motion scenarios are employed to maintain desired formation distance $d_{ij}^* = 100$ [m] based on the desired relative positions: $R_{(1,2)}^* = (-50, -50\sqrt{3})^T$, $R_{(1,3)}^* = (50, -50\sqrt{3})^T$, $R_{(2,3)}^* = (100, 0)^T$.

All corresponding results are presented in Figures 6.7-6.11. The formation performances of the proposed distributed control scheme satisfy the formation requirements as seen in Figures 6.7 and 6.8 as well as the individual tracking performances of all UAVs at the low-level.
Figure 6.3: Lateral trajectory tracking performances.

Figure 6.4: Heading tracking performance.
Figure 6.5: Commanded and actual velocities.

Figure 6.6: Commanded and actual angular velocities.
Figure 6.7: Spiral surveillance motion of a multiple fixed-wing UAV system in formation.

Figure 6.8: Formation maintenance performances for $d_{ij}$.
Figure 6.9: Heading tracking performances for $V_i$.

Figure 6.10: Commanded and actual velocities for $V_i$. 

119
6.7 Summary and Remarks

The linear quadratic tracking (LQT) schemes have been developed to control and stabilize the lateral motion of a small fixed-wing UAV system. Then, the two-level, hierarchical, distributed formation controller has been designed for a multiple fixed-wing UAV system with three vehicles. All analytical analyses and designs are verified by surveillance mission simulations for both tracking and formation maintenance control cases. It is witnessed that the LQT schemes in the low (individual) level work successfully in terms of the optimal tracking performances. It is also observed that at the high (formation) level, the distributed laws are practical and maintain successfully desired formation shape by associating with the low (individual) level controllers.

Figure 6.11: Commanded and actual angular velocities for $V_i$. 
Chapter 7

Concluding Remarks

This thesis has studied robust, optimal and nonlinear control design for trajectory tracking of single-UAV systems and formation maintenance of multi-UAV systems. These control problems have been addressed for both quadrotor and small fixed-wing UAVs.

For trajectory tracking of single-UAV systems, Chapter 3 has presented a novel infinite-horizon ALQT control design for the attitude of the quadrotor UAV in an optimal sense. ALQT is selected for this purpose because it is real-time implementable and robust to effect of modeling uncertainties. In the experiments, the attitude measurement noises, which come from IMU sensors, are compensated using a Kalman filter to obtain a more reliable attitude estimation and compared with a complementary filter. In Chapter 4, a backstepping based robust adaptive controller with guaranteed tracking errors has been studied and tested in case of under-actuated dynamics, nonlinearities and model uncertainties. The proposed design is capable of transferring the constrained tracking error to an unconstrained form for adjusting transient and steady-state behaviors within prescribed bounds. It also compensates effects of model nonlinearities and uncertainties. We have concluded that the tracking objectives in Chapters 3 and 4 are achieved and experimentally validated utilizing the proposed controllers even in the existence of under-actuation, nonlinearities, uncertainties and sensor noises in the quadrotor UAV motion dynamics.

For formation maintenance of multi-UAV systems, a two-level control structure is introduced for constructing high (formation) and low (individual) level controllers, separately. This separation helps efficient and systematic control synthesis addressing robustness to effects of uncertainties and disturbances as well as optimality at both levels, independently. In Chapter 5, using the proposed two-level control, at the high-level, a distributed hierarchical formation control scheme has been designed in leader-follower structure. Formation
maintenance is ensured by utilizing tools of rigid graph theory for performing cohesive motion in 3D. At the low-level, the proposed adaptive mixing based controller is designed to compensate the effect of real dynamics issues for enhancing tracking performance and robustness by providing smoother control action. We have concluded that the multi quadrotor UAV system achieves the formation control objectives with the proposed controller even in the existence of parametric uncertainties in the quadrotor UAV motion dynamics. Especially, the proposed control scheme compensates the negative effects of the indirect adaptive approach which come from poor transient estimation time of the parameter identification algorithm. Furthermore, in Chapter 6, the proposed single-UAV linear quadratic tracking control and multi-UAV two-level control designs have been extended and studied to fixed-wing UAV systems. Hence, the proposed two-level control approach is modular and practical in the sense that it can be employed easily to various types of multiple UAV systems which have different motion characteristics.

Some of the potential future works and shortcomings that can be studied as a continuation of this thesis are as follows:

(i) In Chapter 3, a potential future direction is to extend optimal linear quadratic attitude tracking control design for altitude and yaw dynamics to perform fully autonomous motion tasks. It is also observed that the heading (yaw) estimation is affected by magnetic disturbances in experiments. Yaw estimation can be improved by following the solution methods in the literature such as those in [50].

(ii) The proposed position control scheme with guaranteed tracking performance designed in Chapter 4 can be extended to formation tasks for guaranteeing formation maintenance performances. One of the shortcomings of the proposed control design is as it lacks analysis for external disturbances such as wind gust. This can be addressed with disturbance rejection approaches, and the robustness of the proposed controller can be improved against real-world disturbances during flight missions.

(iii) For Chapter 5, it would be a good extension to analyze the formation maintenance and robustness of the multi quadrotor UAV system for actuator failures, and study of other cooperative tasks such as cooperative surveillance and joint tasks with ground vehicles. Another future direction is to implement the proposed formation control scheme on a real multiple quadrotor UAV system for an experimental validation. As a future solution for shortcoming of Chapter 5, the nominal LQR control design can be studied combining with a disturbance rejection method to eliminate internal disturbance effects of ignored modeling terms and real-world disturbances such as wind gust.
Bibliography


tors,” in Proc. IEEE International Conference on Robotics and Automation, pp. 5376-5381, 
Karlsruhe, Germany, May 2013.


application,” in Recent advances in learning and control, Ser. Lecture Notes in Control and 

for autonomous formations,” IEEE Control Systems Magazine, vol. 28, no. 6, pp. 48-63, 
2008.

“Real-time integral backstepping with sliding mode control for a quadrotor UAV,” in Proc. 

autonomous helicopter,” in Proc. IEEE International Conference on Intelligent Engineering 


[16] S. Baldi, P. Ioannou and E. Mosca, “Multiple model adaptive mixing control: the discrete-


[18] J.D. Barton, “Fundamentals of small unmanned aircraft flight,” Johns Hopkins APL tech-

control for MIMO uncertain non-linear quadrotor helicopter systems,” Transactions of the 


and M.A. Goodrich, “Autonomous vehicle technologies for small fixed-wing UAVs,” Journal


128


