Spring 2015

Linear matrix inequality-based nonlinear adaptive robust control with application to unmanned aircraft systems

David William Kun
Purdue University, davidkun4@gmail.com

Graduate School Form 30
Updated 1/15/2015

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By David William Kun

Entitled
Linear Matrix Inequality-based Nonlinear Adaptive Robust Control with Application to Unmanned Aircraft Systems

For the degree of Master of Science in Aeronautics and Astronautics

Is approved by the final examining committee:

Inseok Hwang
Chair

Martin Corless

Dengfeng Sun

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): Inseok Hwang

Approved by: Tom Shih 04/17/2015
Head of the Departmental Graduate Program Date

LINEAR MATRIX INEQUALITY-BASED
NONLINEAR ADAPTIVE ROBUST CONTROL
WITH APPLICATION TO UNMANNED AIRCRAFT SYSTEMS

A Thesis

Submitted to the Faculty

of

Purdue University

by

David William Kun

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Aeronautics and Astronautics

May 2015

Purdue University

West Lafayette, Indiana

I dedicate this to Kasia, for her continuous love and encouragement, and to my family, for helping me achieve my dreams.

ACKNOWLEDGMENTS

I would like to thank my adviser, Professor Inseok Hwang, for his guidance and suggestions that made this research work possible. I would also like to thank professors Martin Corless and Dengfeng Sun for their suggestions that helped shape and finalize my thesis. I would like to acknowledge Professor Bin Yao, whose course on Nonlinear Control inspired my thesis work, and Ricardo Arteaga, whose mentorship helped me form my research objectives.

I would like to acknowledge the financial support from Purdue University's Chappelle Fellowship and the School of Aeronautics and Astronautics' Teaching Assistantships.

I extend my thanks to my good friends Ted Sieffert and Nick Kowalczyk, who helped me through my B.S. and M.S. degrees with their moral and technical support. I am grateful for my family's love and encouragement, and for my amazing wife Kasia, who has helped me in every aspect of life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

SYMBOLS

Mathematical Notation

$\mathbb{R}$	Field of real numbers
■	End of proof (Q.E.D.)
$:=$ and $\triangleq$	Defined as
$\text{diag}(a_1, ..., a_n)$	Diagonal matrix with a_i in the diagonal
$\mathbf{A}^T$ and $\mathbf{A}'$	Transpose of $\mathbf{A}$
$\lambda_{max}(\mathbf{A})$, $\lambda_{min}(\mathbf{A})$	Maximum, minimum eigenvalue of $\mathbf{A}$

Controller Notation

$x(t) \in \mathbb{R}^n$	State vector
$x_d(t) \in \mathbb{R}^n$	Desired state vector
$\boldsymbol{\varphi}(x,t) \in \mathbb{R}^p$	Known basis functions
$\boldsymbol{\theta}$, $\hat{\boldsymbol{\theta}}(t)$, $\tilde{\boldsymbol{\theta}}(t) \in \mathbb{R}^p$	Uncertain parameters, estimates, and errors
$\Delta(x,t)$	Disturbances and unmodeled dynamics

Quadrotor Notation

ϕ, θ, ψ	Roll, pitch, yaw angles (inertial frame), [rad]
N, E, h	North, East, altitude positions (inertial), [m]
p, q, r	Roll, pitch, yaw rates (body frame), [rad/s]
u, v, w	x, y, z velocities (body frame), [m/s]
I_{xx}, I_{xx}, I_{xx}, I_{xx}	Moments of inertia, [kg·m^2]
b	Rotor drag factor, kg·m^2
k_t	Rotor thrust factor, [kg·m]
k_m	Motor gain, [1/s]
Ω_i	i-th Rotor's angular velocity, [rad/s^2]
T	Total thrust force, [N]
τ_ϕ,τ_θ,τ_ψ	Roll, pitch, yaw torques, [N·m]

ABBREVIATIONS

AC	Adaptive Control
ARC	Adaptive Robust Control
DRC	Deterministic Robust Control
EKF	Extended Kalman Filter
FAA	Federal Aviation Administration
FBL	Feedback Linearization
GNC	Guidance Navigation and Control
GPS	Global Positioning System
HIL	Hardware-in-the-Loop
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
KF	Kalman Filter
LMI	Linear Matrix Inequality
UAS	Unmanned Aerial (or Aircraft) System
UAV	Unmanned Aerial Vehicle

ABSTRACT

Kun, David William MSAA, Purdue University, May 2015. Linear Matrix Inequality-based Nonlinear Adaptive Robust Control with Application to Unmanned Aircraft Systems. Major Professor: Inseok Hwang.

Unmanned aircraft systems (UASs) are gaining popularity in civil and commercial applications as their lightweight on-board computers become more powerful and affordable, their power storage devices improve, and the Federal Aviation Administration addresses the legal and safety concerns of integrating UASs in the national airspace. Consequently, many researchers are pursuing novel methods to control UASs in order to improve their capabilities, dependability, and safety assurance. The nonlinear control approach is a common choice as it offers several benefits for these highly nonlinear aerospace systems (e.g., the quadrotor). First, the controller design is physically intuitive and is derived from well known dynamic equations. Second, the final control law is valid in a larger region of operation, including far from the equilibrium states. And third, the procedure is largely methodical, requiring less expertise with gain tuning, which can be arduous for a novice engineer.

Considering these facts, this thesis proposes a nonlinear controller design method that combines the advantages of adaptive robust control (ARC) with the powerful design tools of linear matrix inequalities (LMI). The ARC-LMI controller is designed with a discontinuous projection-based adaptation law, and guarantees a prescribed transient and steady state tracking performance for uncertain systems in the presence of matched disturbances. The norm of the tracking error is bounded by a known function that depends on the controller design parameters in a known form. Furthermore, the LMI-based part of the controller ensures the stability of the system while overcoming polytopic uncertainties, and minimizes the control effort. This can reduce the number of parameters that require adaptation, and helps to avoid control input saturation.

These desirable characteristics make the ARC-LMI control algorithm well suited for the quadrotor UAS, which may have unknown parameters and may encounter external disturbances such as wind gusts and turbulence. This thesis develops the ARC-LMI attitude and position controllers for an X-configuration quadrotor helicopter. The inner-loop of the autopilot controls the attitude and altitude of the quadrotor, and the outer-loop controls its position in the earth-fixed coordinate frame. Furthermore, by intelligently generating a smooth trajectory from the given reference coordinates (waypoints), the transient performance is improved. The simulation results indicate that the ARC-LMI controller design is useful for a variety of quadrotor applications, including precise trajectory tracking, autonomous waypoint navigation in the presence of disturbances, and package delivery without loss of performance.

1. INTRODUCTION

This thesis introduces a novel linear matrix inequality-based adaptive robust control framework for uncertain dynamic systems in the presence of bounded external disturbances. The proposed controller design methodology is applied to an autonomous unmanned aircraft system, i.e. a quadrotor helicopter, in order to demonstrate the performance of the controller and its advantages. The following two sections provide concise literature surveys of relevant previous research. Section 1.1 describes related works in adaptive robust control and linear matrix inequalities, as well as the motivation for the proposed controller architecture. Section 1.2 outlines the various approaches to modeling and controlling the quadrotor helicopter, and provides the motivation for studying this platform.

1.1 Nonlinear Control: Related Works & Motivation

Nonlinear control of second-order dynamic systems such as electro-hydraulic and electro-mechanical systems has been actively researched by control engineers for several decades. The advances in manufacturing and computing have enabled these systems to perform at high speeds in varying and hostile environments with unprecedented levels of accuracy. This naturally comes at the cost of requiring more complex controller designs. Nonlinear control is a powerful approach that can take into account hard nonlinearities (such as Coulomb friction, dead-zones, and hysteresis) and model uncertainties (such as slow or abrupt parameter changes). Furthermore, it can handle the system's nonlinearities in a broad range of operation, which enables modern systems to operate at higher speeds while maintaining the same desired accuracy (see [46] for a thorough analysis of applied nonlinear control).

Traditionally, two classes of nonlinear controllers have been used on these dynamic systems: robust controllers and adaptive controllers. A newer approach introduced in [55], named adaptive robust control (ARC), merges the competing strategies of deterministic robust control (DRC) and adaptive control (AC) via a cleverly designed adaptation law, and achieves fast robust response with a good parameter estimation ability. ARC has been demonstrated on several high-performance industrial applications with excellent results, e.g. precision motion control of linear motors [52]. In this thesis, we use the results from ARC in combination with the linear matrix inequalities (LMI)-based controller design approach [7]. For uncertain nonlinear systems in a special polytopic form, an LMI-based optimization problem generates a static feedback controller that guarantees global exponential stability.

The motivation behind using the LMI-based part of the controller is several-fold. First, we can guarantee the convergence of the nominal system by selecting the desired convergence rate and solving the LMI convex optimization problem offline (Chapter 7 of [7]). The result yields a conservative minimum-norm feedback gain matrix, which, in turn, leads to smaller controller effort and helps to avoid actuator saturation. Second, the LMI-based control input also guarantees the convergence in the presence of polytopic uncertainties and nonlinearities [37], which can eliminate the need for additional parameter estimates in the adaptation law. This reduces computation time and avoids potential performance loss when the reference trajectory is not persistently exciting. Third, if a polytopic nonlinearity is associated with a noisy measurement, the LMI-based controller performance will not be degraded by the nonlinearity, as only the predetermined bounds are considered in the controller design.

There are several recent works that integrate LMI-based control [29] with other existing control approaches in order to overcome time delays, uncertain parameters, and/or external disturbances. In [10], an LMI-based sliding mode control is introduced for nonlinear systems in a special form, but no online learning scheme is considered. In [51], an adaptive sliding mode control scheme is derived using LMI theory to compute the sliding surfaces, but is limited to uncertain *linear* systems. The authors in [32] develop a discontinuous controller with LMI-based adaptive sliding surfaces for nonlinear systems with time delays. A model reference robust adaptive controller is developed in [49] for a class of uncertain switched linear systems, and stability is proved with LMIs. The authors of [38] examine a robust feedback linearization control scheme based on fuzzy models for uncertain special systems with structured uncertainty by casting the system into a Lur'e system form. An adaptive robust controller design based on LMIs is considered in [61] for uncertain *linear* systems with time delays. This research work sets itself apart by considering a more general class of second-order uncertain *nonlinear* systems, and smoothly integrating LMI-based feedback with the well-developed ARC approach [53].

1.2 Quadrotor Control: Related Works & Motivation

The quadrotor helicopter, more commonly referred to as a 'quadcopter' or simply 'quadrotor', is a very maneuverable rotorcraft with four rotors of equal size. The first documented quadrotors, from as early as the 1920's, were very large (over 1,600 kg) and each of the rotors had collective pitch control for differential thrust [15]. Since then, quadrotors have changed considerably, with an emphasis on lightweight designs and acrobatic capabilities. For example, a typical quadrotor measures approximately 0.5 meters diagonally, weighs approximately 1 kg, and is capable of flying between 15 and 30 minutes. Quadrotors have gained popularity over the last decade in both research works and commercial and civil applications. The advances in electro-mechanical, and power storage systems, as well as on-board computation capabilities, have rendered the quadrotor a more dependable and maneuverable aerial vehicle with many applications. For example, the Federal Aviation Administration (FAA) approved the oil and gas company BP to inspect their Alaskan oil fields with quadrotors [22], the German firm DHL has launched a quadrotor package-delivery service in 2014 [18], and the Israeli company Bladeworx, which specializes in drone aerial photography, is working towards a drone surveillance system to protect the

Jerusalem light rail from riots and vandalism [45]. As the FAA works towards legislating the civil and commercial use of unmanned aircraft systems in the national airspace (UAS in the NAS), the developments of UAS applications are expected to greatly expand.

Many controller design approaches have been considered for the quadrotor helicopter, including PID [31], [6], [16], LQR [6], [42]), model-predictive control [5], adaptive and robust adaptive control (RAC) [33], [36], [3], [13], and robust sliding-mode control [25], [39], [44], to name a few. Some of these approaches are focused on acrobatic and aggressive maneuvers [31], some aim to achieve robust hover stability during wind gusts [50], and others are interested in adaptive fault-tolerant control and health monitoring [13], [40].

In this thesis, we develop a linear matrix inequality-based nonlinear adaptive robust controller (ARC-LMI) for the quadrotor's attitude and position. The new ARC-LMI controller approach is well-suited for the quadrotor system since, in addition to the guaranteed transient tracking performance, the projection-based adaptation law improves the steady state performance. Furthermore, the bound on the norm of the error is prescribed by the controller design parameters in known form, assuring a safe performance in the presence of bounded external disturbances (e.g. wind gusts). It should be noted that the ARC-LMI approach is fundamentally different from the RAC approach in [14]. First, as in the traditional ARC design [55], the ARC-LMI controller emphasizes the robust feedback term, which accounts for parametric uncertainties in addition to external disturbances. Moreover, by switching off the adaptation law, the result is a deterministic robust controller that maintains the prescribed transient tracking performance [57].

The objective of the ARC-LMI controller is to enable the quadrotor to carry and deliver payloads of uncertain mass in the presence of bounded environmental disturbances. We require only information about the bounds of the uncertain parameters and external disturbances in order to guarantee the controller performance. Therefore, in the quadrotor dynamic equations, we include the environmental forces, such as constant wind and sporadic wind gusts. These considerations are crucial for precise trajectory tracking in a range of external conditions. The controller is in a cascaded structure, with an outer-loop position controller and an inner-loop attitude-altitude controller. Furthermore, the performance of the trajectory tracking is improved via a carefully constructed exogenous system for trajectory generation.

Irrespective of the particular objective, most nonlinear controller designs require an analytical or experimental model of the quadrotor system. The two main quadrotor structural configurations are the ‘X-shape’ and the ‘plus-shape’. Plus-shape quadrotors are more commonly researched as they are simpler to model and control, given that they are symmetrical, with a diagonal inertia matrix, and that their rotors are aligned with the body-centered coordinate frame. On the other hand, X-configuration quadrotors have a more complex dynamic model (depending on the fidelity of the model), but are considered to be more stable [17]. In this thesis, we focus on the X-configuration quadrotor, with the model based on the IRIS quadrotor [1]. We construct the analytical model using Newtonian mechanics and some relatively mild assumptions about the rotor and motor dynamics.

1.3 Outline of the Thesis

The remainder of this thesis is organized in the following manner. **Chapter 2** is dedicated to the proposed ARC-LMI controller design method and results. It presents the problem statement, notation, and assumptions, and reviews the theoretical formulation and lemmas from LMI-based control and ARC design. Section 2.2, specifically, develops the nonlinear ARC-LMI controller algorithm and provides a stability and robustness theorem with its corresponding proofs. The chapter concludes with a demonstration of the ARC-LMI controller methodology for an example dynamic system, and presents the illustrative simulation results. **Chapter 3** is dedicated to applying the ARC-LMI controller to a quadrotor. It derives the detailed dynamic model of the quadrotor, provides an overview of the controller architecture, constructs the attitude-altitude and position controllers, and proposes a trajectory generation method to further improve the transient tracking performance. The chapter concludes with simulation results for two illustrative examples, which demonstrate the adaptive and robust performance of the controller. **Chapter 4** presents a conclusion to the findings in this thesis. It provides a summary of the contributions from both chapters and details the results of the ARC-LMI controller approach. The chapter ends with proposed directions for future research, both theoretical and application-oriented.

2. LINEAR MATRIX INEQUALITY-BASED NONLINEAR ADAPTIVE ROBUST CONTROL

This chapter develops a novel approach to nonlinear control of uncertain second-order systems in the presence of matched disturbances, assuming basic knowledge about the bounds of the uncertain parameters and disturbances. The approach combines the advantages of adaptive robust control (ARC) with the powerful design tools of linear matrix inequalities (LMI). Similar to the results from traditional ARC, the discontinuous projection-based ARC-LMI controller guarantees predetermined transient and steady state tracking performance in the presence of bounded external disturbances and parameter uncertainties. The norm of the error is bounded by a known function that depends on the controller design parameters in a known form, allowing one to prescribe the acceptable tracking error with certainty. Furthermore, the LMI-based part of the controller ensures the stability of the system while overcoming polytopic uncertainties, and minimizes the control effort. This can reduce the number of parameters that require adaptation and helps to avoid control input saturation. We apply the ARC-LMI control algorithm to an inverted single-link manipulator in the presence of unstructured uncertainty and drag force to demonstrate the ARC-LMI procedure and performance.

2.1 Problem Statement and Preliminaries

Consider a dynamic system described by the following second-order differential equation.

$$\ddot{x} = f(\bar{x}, t) + g(\bar{x}, t)\, u \tag{2.1}$$

where $f(\cdot)$ and $g(\cdot)$ are nonlinear functions of the state and time, $\bar{x} = [x(t),\, \dot{x}(t)]^T$ is the state vector (e.g. position and velocity or angle and angular rate), and u is the control input (e.g. force or torque). The nonlinear function $f(\cdot)$ may contain uncertain parameters and nonlinear disturbances (e.g. unmodeled dynamics and external disturbances), but it is assumed that their bounds are known. Additionally, we assume that $g^{-1}(\cdot)$ is bounded for $\bar{x} \in \mathbb{R}^2$ and $t > 0$. These assumptions are relatively mild and are stated formally later in this section. The system (2.1) has the following state-space representation when $x_1 := x(t)$ and $x_2 := \dot{x}(t)$

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f(\bar{x}, t) + g(\bar{x}, t)\, u \end{aligned} \tag{2.2}$$

The objective is to design a bounded control input u such that the system is stable and the state x_1 tracks a desired trajectory $x_d(t)$ as closely as possible. To achieve this, first define the tracking error $z_1 := x_1 - x_d$ and $z_2 := x_2 - \dot{x}_d$, which yields the following error dynamics

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= f(\bar{z}, t) - \ddot{x}_d(t) + g(\bar{z}, t)\, u \end{aligned} \tag{2.3}$$

where the error vector is $\bar{z} = [z_1, z_2]^T$. In general, we can decompose $f(\bar{z}, t)$ into three parts:

$$f(\bar{z}, t) = \boldsymbol{\varphi}_p(\bar{z}, t)^T \boldsymbol{\theta}_p + \boldsymbol{\varphi}_{np}(\bar{z}, t)^T \boldsymbol{\theta}_{np} + \Delta(\bar{z}, t) \tag{2.4}$$

Here $\boldsymbol{\varphi}_p(\bar{z}, t) \in \mathbb{R}^{p_1}$ is a vector of known basis functions that are linearly parametrized by unknown weights $\boldsymbol{\theta}_p \in \mathbb{R}^{p_1}$ and can be written in a special polytopic form:

$$\begin{bmatrix} 0 \\ \boldsymbol{\varphi}_p(\bar{z}, t)^T \boldsymbol{\theta}_p \end{bmatrix} = A(\bar{z}, t) \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

$A(\bar{z}, t)$ has the following structure

$$A(\bar{z}, t) = A_0 + \psi_1(\bar{z}, t)\Delta A_1 + \cdots + \psi_l(\bar{z}, t)\Delta A_l$$

$$a_i \leq \psi_i(\bar{z}, t) \leq b_i$$

where a_i, b_i are known bounds of the scalar function $\psi(\bar{z},t)$ and $A_0, \Delta A_i \in \mathbb{R}^{(2\times 2)}$ are known constant matrices for $i = 1, \ldots, l$. $\boldsymbol{\varphi}_{np}(\bar{z},t) \in \mathbb{R}^{p_2}$ is a vector of known basis functions that are linearly parametrized by unknown weights $\boldsymbol{\theta}_{np} \in \mathbb{R}^{p_2}$ but are non-polytopic, and $\Delta(\bar{z},t) \in \mathbb{R}$ is a nonlinear function which captures external disturbances, unmodeled dynamics, and terms that cannot be linearly parametrized.

Combining (2.3) and (2.4) we have the following second-order error dynamics.

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= \boldsymbol{\varphi}_p(\bar{z},t)^T\boldsymbol{\theta}_p + \boldsymbol{\varphi}_{np}(\bar{z},t)^T\boldsymbol{\theta}_{np} + \Delta(\bar{z},t) - \ddot{x}_d(t) + g(\bar{z})\,u \end{aligned} \tag{2.5}$$

As in [56], we assume some knowledge of the bounds of parameters and disturbances in the system in (2.5).

Assumption 2.1.1 *The uncertain parameters $\boldsymbol{\theta} = [\boldsymbol{\theta}_p^T, \boldsymbol{\theta}_{np}^T]^T$ lie in a known bounded region Ω_θ and the disturbances Δ are bounded by a known function $\delta(\bar{z},t)$, that is,*

$$\boldsymbol{\theta} \;\in\; \Omega_\theta \triangleq \{\boldsymbol{\theta} : \boldsymbol{\theta}_{min} < \boldsymbol{\theta} < \boldsymbol{\theta}_{max}\} \tag{2.6}$$

$$\Delta \;\in\; \Omega_\Delta \triangleq \{\Delta : |\Delta(\bar{z},t)| \leq \delta(\bar{z},t)\} \tag{2.7}$$

where the lower and upper bounds of the parameters $\boldsymbol{\theta}_{min} = (\theta_{1,min}, \ldots, \theta_{(p_1+p_2),min})^T$ and $\boldsymbol{\theta}_{max} = (\theta_{1,max}, \ldots, \theta_{(p_1+p_2),max})^T$ and the function $\delta(t,x)$ are known.

Before continuing with the controller design, we review some preliminary theory of linear matrix inequality (LMI) control of nonlinear polytopic systems and adaptive robust control (ARC) methodology, which will be used to support the proposed ARC-LMI control approach.

2.1.1 Controlling Polytopic Uncertain/Nonlinear Systems with Linear Matrix Inequalities

The use of LMIs to analyze the stability of dynamic systems began in the 1890s, after Aleksandr Lyapunov showed that the system $\dot{x} = A\,x$ is stable if and only if there exists a positive definite matrix P such that $A^TP + PA < 0$, which is a special form of an LMI on P (Chapter 1 of [7]). Since then, much progress has been made in the field and it has been found that LMIs appearing in control theory can be expressed as convex optimization problems, which, in turn, can be solved by very efficient numerical algorithms. The MATLAB software package, for example, offers an LMI Control Toolbox for LMI-based stability analysis and controller design. Of the many applications of LMIs in control theory (see [7]), we are particularly interested in controlling polytopic uncertain nonlinearities, which frequently appear in models of dynamic systems. More specifically, consider a second-order polytopic uncertain nonlinear system in the form

$$\dot{x} = A(x,t)x + B(x,t)u \tag{2.8}$$

where $x \in \mathbb{R}^2$ is the state vector and the matrices A and B have the following structure:

$$\begin{aligned} A(x,t) &= A_0 + \psi_1(x,t)\Delta A_1 + \cdots + \psi_l(x,t)\Delta A_l \\ B(x,t) &= B_0 + \psi_1(x,t)\Delta B_1 + \cdots + \psi_l(x,t)\Delta B_l \end{aligned} \tag{2.9}$$

and $A_0,\ \Delta A_i \in \mathbb{R}^{(2\times 2)}$ and $B_0,\ \Delta B_i \in \mathbb{R}^{(2\times 1)}$ are constant matrices for $i = 1, \ldots, l$, and ψ_i are scalar-valued functions that satisfy

$$a_i \leq \psi_i(x,t) \leq b_i \tag{2.10}$$

for $i = 1, \ldots, l$ and with known bounds, a_i and b_i. Note that ψ_i can be a bounded nonlinear function *or* an uncertain parameter that lies within known bounds. Then, for all $t \in \mathbb{R}$ and $x \in \mathbb{R}^2$, the matrices $A(x,t), B(x,t)$ satisfy

$$\begin{bmatrix} A(x,t) & B(x,t) \end{bmatrix} \in \mathbf{Co}\left\{ \begin{bmatrix} A_1 & B_1 \end{bmatrix}, \ \ldots\ , \begin{bmatrix} A_N & B_N \end{bmatrix} \right\} \tag{2.11}$$

that is, the matrix pair remains in a convex hull with a finite number of vertices [37].

Lemma 2.1.1 *The state feedback controller*

$$u = LS^{-1}x \tag{2.12}$$

where the symmetric positive definite (s.p.d.) matrix S and the matrix L are solutions to the following optimization problem:

$$\text{Minimize } \beta \text{ subject to: } \quad \begin{cases} AS + BL + SA^T + L^TB^T + 2\alpha S \leq 0 \\ S \geq I \\ \begin{pmatrix} \beta I & L^T \\ L & S \end{pmatrix} \geq 0 \end{cases} \tag{2.13}$$

for all (A, B) in $\mathcal{AB}$

where the 2^l pairs of (A, B) in $\mathcal{AB}$ are defined below:

$$\mathcal{AB} = \{(A_0 + \delta_1 A_1 + \cdots + \delta_l A_l, \quad B_0 + \delta_1 B_1 + \cdots + \delta_l B_l) : \delta_i = a_i \text{ or } b_i \text{ for } i = 1, \ldots, l\}$$

makes the system (2.8) globally uniformly exponentially stable about the origin, with a convergence rate of α, and with the Lyapunov matrix $P = S^{-1}$. In addition, the parameter β determines the upper bound on the feedback gain $||LS^{-1}||^2$.

For a proof of Lemma 2.1.1, see Section 2.4.

2.1.2 Adaptive Robust Control Methodology

ARC was proposed in [55] as a high-performance nonlinear control approach that integrates deterministic robust control (DRC) and adaptive control (AC). By assuming that bounds on disturbances and uncertain parameters are known, the ARC approach maintains the benefits of these competing design approaches while overcoming their drawbacks [58]. It should be noted that ARC is different from the robust adaptive control approach, e.g. [20], [21].

ARC has been implemented on electro-mechanical systems and has proven to be a powerful control approach (see for example [62]). In recent years, ARC has been developed to be less sensitive to noise (Desired Compensation ARC), provide accurate parameter estimates (Indirect ARC), and be faster than the previous indirect ARC (Integrated Direct/Indirect ARC) [54]. For clarity, we consider the traditional direct

ARC design, however it is relatively straightforward to extend the work in this paper to the more sophisticated ARC methods. In order to concisely convey the ARC methodology, consider the simple first order system in (2.14).

$$\dot{x} = \boldsymbol{\varphi}^T \boldsymbol{\theta} + \Delta + u \tag{2.14}$$

As before, $\boldsymbol{\varphi}$ is a regressor of known basis functions, $\boldsymbol{\theta}$ are the uncertain parameters that lie within known bounds, Δ represents the disturbances and un-modeled dynamics bounded by a known function, and u is the control input. ARC starts with feedback linearization and a backstepping approach (for higher-order systems) in order to track a desired trajectory $x_d(t)$, i.e. the initial control input is

$$u = \dot{x}_d - \boldsymbol{\varphi}^T \hat{\boldsymbol{\theta}} - kz + u_r \tag{2.15}$$

where $z = x - x_d$ is the tracking error, $\hat{\boldsymbol{\theta}}$ are the parameter estimates, and u_r is the DRC feedback term. This input yields the closed-loop system

$$\dot{z} + kz = u_r + [\boldsymbol{\varphi}^T \tilde{\boldsymbol{\theta}} + \Delta] \tag{2.16}$$

where $\tilde{\boldsymbol{\theta}}(t) = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(t)$ is the error in parameter estimates. ARC combines the DRC and AC approaches by designing a DRC controller to account for disturbances *and* parameter uncertainties. The left hand side of (2.16) is the stable nominal system when $k > 0$, and the right hand side in brackets represents all uncertainties and disturbances. The ideal sliding mode control law can be used to overcome these terms.

$$u_r = -\rho\,\mathrm{sgn}(z) = \begin{cases} -\rho\,\frac{z}{|z|} & \text{, if } z \neq 0 \\ 0 & \text{, if } z = 0 \end{cases} , \quad \rho \geq \left| \boldsymbol{\varphi}^T \tilde{\boldsymbol{\theta}} + \Delta \right| \tag{2.17}$$

Note that the ideal sliding mode control law leads to control chattering, so smoother approximating functions $\mathcal{S}\left(\rho\,\mathrm{sgn}(z)\right)$ can be used instead (such as the saturation or hyperbolic tangent function). This yields a prescribed transient tracking performance as well as steady state tracking accuracy within a specified radius. The final control law, then, is given by

$$u = \dot{x}_d - \boldsymbol{\varphi}^T \hat{\boldsymbol{\theta}} - kz - \mathcal{S}\left(\rho\,\mathrm{sgn}(z)\right) \tag{2.18}$$

An adaptation law is used to update the parameter estimates $\hat{\boldsymbol{\theta}}(t)$ as in AC, which improves the steady state performance and yields asymptotic tracking in the absence of disturbances (i.e. with only parametric uncertainties). However, to merge DRC and AC in this way, it is necessary to keep the parameter estimates bounded and, in turn, the right hand side of (2.16). The traditional integral adaptation law can lead to unbounded parameter estimates when there are disturbances, so the discontinuous projection adaptation law from [55] is implemented,

$$\dot{\hat{\boldsymbol{\theta}}} = \text{Proj}_{\hat{\theta}}(\boldsymbol{\Gamma}\boldsymbol{\varphi}z) \tag{2.19}$$

where $\boldsymbol{\Gamma}$ is a diagonal positive definite matrix, and $\text{Proj}_{\theta}(\bullet)$ is defined component-wise for vector $\bullet$ such that $\text{Proj}_{\hat{\theta}}(\bullet) = \left[\text{Proj}_{\hat{\theta}_1}(\bullet_1), \ldots, \text{Proj}_{\hat{\theta}_p}(\bullet_p)\right]^T$, and

$$\text{Proj}_{\hat{\theta}_i}(\bullet_i) := \begin{cases} 0 & \text{if} \begin{cases} \hat{\theta}_i = \hat{\theta}_{i,max} \text{ and } \bullet_i > 0 \\ \hat{\theta}_i = \hat{\theta}_{i,min} \text{ and } \bullet_i < 0 \end{cases} \\ \bullet_i & \text{otherwise} \end{cases} \tag{2.20}$$

which guarantees that $\hat{\boldsymbol{\theta}}(t) \in \Omega_{\theta}$ for all t, and therefore $\tilde{\boldsymbol{\theta}}(t)$ is bounded.

Lemma 2.1.2 *The ARC law* (2.18) *implemented with the adaptation law* (2.19) *on the system* (2.14) *guarantees that the tracking error is bounded above by a known function and it converges to a ball of prescribed radius with a convergence rate of at least k. Additionally, if the disturbances are eliminated for all $t > t_0 \geq 0$, the tracking error asymptotically converges to zero.*

For a proof of Lemma 2.1.2, please refer to [53].

In the following section, we use the previous problem statement and preliminary theory to design a control input $u(z, t)$ that employs adaptive robust control techniques together with LMI-based feedback in order to guarantee fast and accurate tracking of the dynamic system in (2.5).

2.2 ARC-LMI Controller Design

In this section, we present a nonlinear adaptive robust controller that incorporates linear matrix inequalities (ARC-LMI) with the objective of tracking a time-varying trajectory in a second-order uncertain dynamic system with bounded disturbances. The traditional ARC method is modified and combined with an LMI-based feedback for stabilization and overcoming polytopic uncertainties and nonlinearities in the system. We describe a continuous approximation to the signum function for the robust feedback control input and a modified adaptive control law. Furthermore, we prove the guaranteed properties of the transient performance and steady state tracking with and without external disturbances.

Let us return to the uncertain second-order system in (2.1) under Assumption 2.1.1, which has the following error dynamics, from (2.5).

$$\dot{\bar{z}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \bar{z} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{ \boldsymbol{\varphi}_p(\bar{z},t)^T \boldsymbol{\theta}_p + \boldsymbol{\varphi}_{np}(\bar{z},t)^T \boldsymbol{\theta}_{np} + \Delta(\bar{z},t) - \ddot{x}_d(t) + g(\bar{z})\, u \right\} \tag{2.21}$$

For now, suppose that $g(\bar{z},t)$ is known (this assumption will be relaxed at a later point). We begin by decomposing the control law into three parts: u_s, a stabilizing state feedback and compensation for polytopic uncertainties, u_m, a dynamic model compensation term, and u_r, a robust control law. Then the complete control law can be written as

$$u = g(\bar{z})^{-1}(u_s + u_m + u_r) \tag{2.22}$$

It has been shown (e.g. [48]) that using feedback linearization prior to additional control approaches can be advantageous when there are severe nonlinearities. Therefore, we first apply a dynamic model compensation term u_m using parameter estimates $\hat{\boldsymbol{\theta}}_p$ and $\hat{\boldsymbol{\theta}}_{np}(t)$, which is updated online via an adaptation law, to be specified later.

$$u_m = \ddot{x}_d(t) - \boldsymbol{\varphi}_p(\bar{z},t)^T \hat{\boldsymbol{\theta}}_p - \boldsymbol{\varphi}_{np}(\bar{z},t)^T \hat{\boldsymbol{\theta}}_{np}(t) \tag{2.23}$$

Thus, the closed-loop system is given by

$$\dot{\bar{z}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \bar{z} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{ u_s + \boldsymbol{\varphi}_p(\bar{z},t)^T \tilde{\boldsymbol{\theta}}_p \right\} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{ u_r + \boldsymbol{\varphi}_{np}(\bar{z},t)^T \tilde{\boldsymbol{\theta}}_{np}(t) + \Delta(\bar{z},t) \right\} \tag{2.24}$$

where $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}$ is the error in the parameter estimates. Next, we design the stabilizing feedback u_s, which is robust to polytopic uncertainty. In the traditional ARC method, a backstepping approach would be used to stabilize the nominal second-order system, by selecting two feedback gains and designing a virtual control input for z_2. In contrast, the ARC-LMI algorithm computes the static feedback term

$$u_s = K\bar{z} = LS^{-1}\bar{z} \tag{2.25}$$

where the s.p.d. matrix S and the matrix L are solutions to the optimization problem in (2.13), with a desired convergence rate α, and β as an upper bound on $||K||^2$ for the system

$$\dot{\bar{z}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \bar{z} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \boldsymbol{\varphi}_p(\bar{z},t)^T \tilde{\boldsymbol{\theta}}_p + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_s \tag{2.26}$$

which can be written in the form of system (2.8), with

$$A(\bar{z},t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \bar{z} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \boldsymbol{\varphi}_p(\bar{z},t)^T \tilde{\boldsymbol{\theta}}_p \quad , \qquad B(\bar{z},t) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.27}$$

This yields an optimal feedback term for the nominal system *with* polytopic uncertainty based on the desired convergence rate, rather than by an arbitrary choice of backstepping gains. From this we also obtain the s.p.d Lyapunov matrix $P = S^{-1}$. Now we design the robust feedback term u_r to overcome the parametric uncertainties associated with non-polytopic terms and the disturbances, that is

$$u_r = -\boldsymbol{\mathcal{S}}\left(\rho(\bar{z},t)\,\mathrm{sgn}(\eta)\right) \tag{2.28}$$

where

$$\eta = p_{12}z_1 + p_{22}z_2 \tag{2.29}$$

with p_{ij} from Lyapunov matrix P, and the bounding function, $\rho(\bar{z},t)$ satisfies

$$\rho(\bar{z},t) \geq \left|\boldsymbol{\varphi}_{np}(\bar{z},t)^T\tilde{\boldsymbol{\theta}}_{np} + \Delta(\bar{z},t)\right| \tag{2.30}$$

An example of such a function $\rho(\bar{z},t)$, given our knowledge of the system bounds, is

$$\rho(\bar{z},t) = \left|\boldsymbol{\varphi}_{np}(\bar{z},t)\right|^T \; |\boldsymbol{\theta}_{np,max} - \boldsymbol{\theta}_{np,min}| + \delta(\bar{z},t) \tag{2.31}$$

where $|\bullet|$ is defined component-wise for vector $\bullet$, and $\delta(\bar{z},t) \geq |\Delta(\bar{z},t)|$, as in Assumption 2.1.1. A continuous approximation $\mathcal{S}(\bullet)$ of the signum function, for example, is the hyperbolic tangent,

$$\mathcal{S}\left(\rho\,\mathrm{sgn}(\eta)\right) = \rho\;\tanh\left(\frac{\kappa\rho}{\epsilon}\eta\right) \tag{2.32}$$

where $\kappa = 0.2785$, which satisfies the following two properties [53]:

$$\begin{array}{lll} \text{i.} & \eta\,\mathcal{S}\left(\rho\,\mathrm{sgn}(\eta)\right) & \geq 0 \\ \text{ii.} & \eta\left[\rho\,\mathrm{sgn}(\eta) - \mathcal{S}\left(\rho\,\mathrm{sgn}(\eta)\right)\right] & \leq \epsilon \end{array} \tag{2.33}$$

These properties will be used later to prove the stability and convergence accuracy in the steady state. Then, from (2.22), (2.23), (2.25), and (2.28), we have the following complete control law.

$$\begin{aligned} &u = g(\bar{z})^{-1}\,\bar{u} \\ &\bar{u} = \ddot{x}_d(t) - \boldsymbol{\varphi}_p(\bar{z},t)^T\hat{\boldsymbol{\theta}}_p - \boldsymbol{\varphi}_{np}(\bar{z},t)^T\hat{\boldsymbol{\theta}}_{np}(t) - \rho(\bar{z},t)\tanh\left(\frac{\kappa\rho(\bar{z},t)}{\epsilon}\eta\right) + LS^{-1}\bar{z} \end{aligned} \tag{2.34}$$

The discontinuous projection-based adaptation law that completes the ARC-LMI algorithm is taken from (2.19)-(2.20) and modified as follows.

$$\dot{\hat{\boldsymbol{\theta}}}_{np} = \mathrm{Proj}_{\hat{\theta}_{np}}(\boldsymbol{\Gamma}\boldsymbol{\varphi}_{np}\eta) \tag{2.35}$$

Note that the parameter estimates associated with the polytopic functions, $\hat{\boldsymbol{\theta}}_p$, do not need to be updated as u_s overcomes those errors at an exponential rate. This is an additional advantage of the ARC-LMI approach. A schematic of the ARC-LMI controller architecture is shown in Figure 2.1.

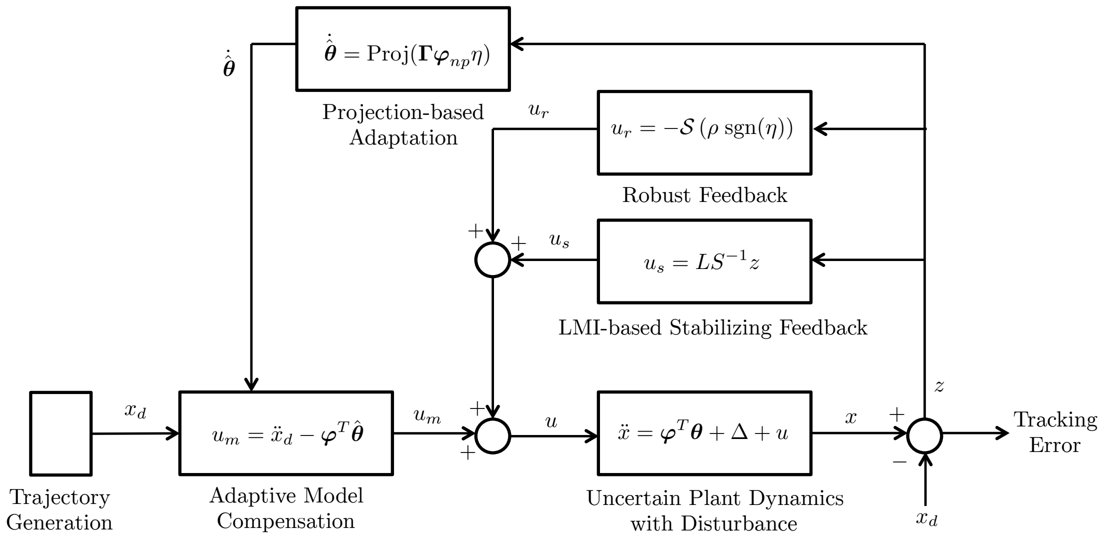

$\dot{\hat{\boldsymbol{\theta}}} = \mathrm{Proj}(\boldsymbol{\Gamma}\boldsymbol{\varphi}_{np}\eta)$
$\dot{\hat{\boldsymbol{\theta}}}$
Projection-based Adaptation
u_r
$u_r = -\mathcal{S}\left(\rho\ \mathrm{sgn}(\eta)\right)$
Robust Feedback
u_s
$u_s = LS^{-1}z$
LMI-based Stabilizing Feedback
+
+
+
+
+
−
x_d
$u_m = \ddot{x}_d - \boldsymbol{\varphi}^T\hat{\boldsymbol{\theta}}$
u_m
u
$\ddot{x} = \boldsymbol{\varphi}^T\boldsymbol{\theta} + \Delta + u$
x
z
Tracking Error
x_d
Trajectory Generation
Adaptive Model Compensation
Uncertain Plant Dynamics with Disturbance

Figure 2.1.. The ARC-LMI controller architecture

Theorem 2.2.1 summarizes the results obtained from the ARC-LMI controller design. As in traditional ARC, we are able to guarantee the following transient and steady state performance properties.

Theorem 2.2.1 (ARC-LMI) *Given the uncertain second-order dynamic system* (2.1) *with the error dynamics shown in* (2.21) *and under Assumption 2.1.1, the nonlinear ARC-LMI control input* (2.34) *together with the adaptation law* (2.35) *yield the following results:*

I. *All signals are bounded and the tracking error is guaranteed to exponentially converge to a ball of constant radius at a rate of convergence no less than α. The transient performance is prescribed and can be improved by increasing α and decreasing ϵ.*

II. *If the system is only subject to parametric uncertainties after some time t_0, i.e. $\Delta(\bar{z}, t) = 0,\ \forall t \geq t_0$, then asymptotic tracking is guaranteed in addition to the results in Part I.*

Proof : Theorem 2.2.1 Part I

Let the error vector $z = [z_1, z_2]^T$ and the Lyapunov matrix from the LMI-based optimization problem be $P = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix}$. Now, consider the positive definite Lyapunov function

$$V(z) = \frac{1}{2} z^T P z \tag{2.36}$$

Taking the derivative of $V(z)$ and substituting the closed-loop error dynamics $\dot{z}$ that results from the ARC-LMI control input (2.34) applied to the original system (2.1) yields

$$\begin{aligned} \dot{V} &= z^T P \dot{z} \\ &= z^T P \left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{ \boldsymbol{\varphi}_p^T \tilde{\boldsymbol{\theta}}_p + u_s \right\} \right) + z^T P \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{ \boldsymbol{\varphi}_{np}^T \tilde{\boldsymbol{\theta}}_{np} + \Delta + u_r \right\} \\ &= z^T P \left(A(z,t) + B(z,t) L S^{-1} \right) z + z^T P \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{ \boldsymbol{\varphi}_{np}^T \tilde{\boldsymbol{\theta}}_{np} + \Delta - \mathcal{S}(\rho \operatorname{sgn}(\eta)) \right\} \end{aligned}$$

From Lemma 2.1.1, we can show that $z^T P \left(A(z,t) + B(z,t) L S^{-1} \right) z \leq -\alpha z^T P z$ for all $z \in \mathbb{R}^2$ and $t \geq 0$, as derived in Appendix 2.4. Therefore,

$$\begin{aligned} \dot{V} &\leq -\alpha z^T P z + z^T P \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{ \boldsymbol{\varphi}_{np}^T \tilde{\boldsymbol{\theta}}_{np} + \Delta - \mathcal{S}(\rho \operatorname{sgn}(\eta)) \right\} \\ &= -2\alpha V + \eta \left\{ \boldsymbol{\varphi}_{np}^T \tilde{\boldsymbol{\theta}}_{np} + \Delta - \mathcal{S}(\rho \operatorname{sgn}(\eta)) \right\} \\ &\leq -2\alpha V + |\eta| \left| \boldsymbol{\varphi}_{np}^T \tilde{\boldsymbol{\theta}}_{np} + \Delta \right| - \eta \mathcal{S}(\rho \operatorname{sgn}(\eta)) \end{aligned}$$

Note that $z^T P \begin{bmatrix} 0 \\ 1 \end{bmatrix} = z_1 p_{12} + z_2 p_{22} = \eta$ as defined in (2.29). Using (2.30) and the second property of (2.33), we arrive at

$$\begin{aligned} \dot{V} &\leq -2\alpha V + \eta \left\{ \rho \operatorname{sgn}(\eta) - \mathcal{S}(\rho \operatorname{sgn}(\eta)) \right\} \\ &\leq -2\alpha V + \epsilon \end{aligned} \tag{2.37}$$

The Comparison Principle (page 102 of [24]) allows us to find the bounds of the tracking error as follows. Since $\dot{V} \leq -2\alpha V + \epsilon$, then

$$V(t) \leq V(0)e^{-2\alpha t} + \frac{\epsilon}{2\alpha}\left(1 - e^{-2\alpha t}\right) \tag{2.38}$$

Using trajectory initialization, i.e. ensuring that $x_d(0) = x_1(0)$ and $\dot{x}_d(0) = x_2(0)$, we can satisfy $z_1(0) = z_2(0) = 0$ and therefore $V(0) = 0$, which results in

$$V(t) \leq \frac{\epsilon}{2\alpha}\left(1 - e^{-2\alpha t}\right) \tag{2.39}$$

Since $V = \frac{1}{2}z^T P z = \frac{\lambda}{2}||z||^2$, where $\lambda \in (0, 1]$ is an eigenvalue of P (from $S \geq I$ in (2.13)), we can write

$$||z|| = \sqrt{\frac{2V}{\lambda}} \leq \sqrt{\frac{\epsilon}{\lambda\alpha}\left(1 - e^{-2\alpha t}\right)} \leq \sqrt{\frac{\epsilon}{\lambda_{min}\alpha}} \tag{2.40}$$

where λ_{min} is the minimum eigenvalue of P. This shows that the error norm $||z||$ is bounded by the design parameters ϵ and α, which can by freely adjusted to predetermine the transient and steady state tracking performance. With the discontinuous projection-based adaptation law, the parameter estimates are bounded and, therefore, the control input is bounded, which implies that the state $x(t)$ is bounded. This completes the proof of Theorem 2.2.1 Part I. ■

Proof : Theorem 2.2.1 Part II

Consider the following positive definite Lyapunov function

$$V(z, \tilde{\boldsymbol{\theta}}_{np}) = \frac{1}{2}z^T P z + \frac{1}{2}\tilde{\boldsymbol{\theta}}_{np}^T \boldsymbol{\Gamma}^{-1}\tilde{\boldsymbol{\theta}}_{np} \tag{2.41}$$

Since the uncertain parameter vector $\boldsymbol{\theta}_{np}$ is assumed to be constant, $\dot{\tilde{\boldsymbol{\theta}}}_{np} = -\dot{\hat{\boldsymbol{\theta}}}_{np}$. The derivative of $V(z, \tilde{\boldsymbol{\theta}}_{np})$ with adaptation law (2.35), then, is given by

$$\dot{V} = z^T P \dot{z} - \tilde{\boldsymbol{\theta}}_{np}^T \boldsymbol{\Gamma}^{-1}\dot{\hat{\boldsymbol{\theta}}}_{np} \tag{2.42}$$

From the closed-loop system in (2.21) resulting from the ARC-LMI control input (2.34), when the system is subject only to parametric uncertainties, i.e. $\Delta = 0$, $\dot{V}$ can be written as

$$\begin{aligned}
\dot{V} &= z^T P \left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{ \boldsymbol{\varphi}_p^T \tilde{\boldsymbol{\theta}}_p + u_s \right\} \right) + z^T P \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{ \boldsymbol{\varphi}_{np}^T \tilde{\boldsymbol{\theta}}_{np} + u_r \right\} \\
&\quad - \tilde{\boldsymbol{\theta}}^T \boldsymbol{\Gamma}^{-1} \mathrm{Proj}_{\hat{\theta}_{np}}(\boldsymbol{\Gamma} \boldsymbol{\varphi}_{np} \eta) \\
&\leq -2\alpha z^T P z + z^T P \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{ \boldsymbol{\varphi}_{np}^T \tilde{\boldsymbol{\theta}}_{np} + u_r \right\} - \tilde{\boldsymbol{\theta}}^T \boldsymbol{\Gamma}^{-1} \mathrm{Proj}_{\hat{\theta}_{np}}(\boldsymbol{\Gamma} \boldsymbol{\varphi}_{np} \eta) \\
&= -2\alpha z^T P z + \eta \left\{ \tilde{\boldsymbol{\theta}}_{np}^T \boldsymbol{\varphi}_{np} + u_r \right\} - \tilde{\boldsymbol{\theta}}^T \boldsymbol{\Gamma}^{-1} \mathrm{Proj}_{\hat{\theta}_{np}}(\boldsymbol{\Gamma} \boldsymbol{\varphi}_{np} \eta) \\
&\leq -2\alpha z^T P z + \tilde{\boldsymbol{\theta}}_{np}^T \left\{ \boldsymbol{\varphi}_{np} \eta - \boldsymbol{\Gamma}^{-1} \mathrm{Proj}_{\hat{\theta}_{np}}(\boldsymbol{\Gamma} \boldsymbol{\varphi}_{np} \eta) \right\} && (2.43) \\
&\leq -2\alpha z^T P z && (2.44)
\end{aligned}$$

Let $k = 2\alpha\bar{\lambda}$, where $\bar{\lambda}$ is the maximum eigenvalue of P. By integrating (2.44),

$$\int_0^t ||z||^2 \leq -\frac{1}{k} \int_0^t \dot{V}(\nu) \mathrm{d}\nu = -\frac{1}{k}[V(t) - V(0)] \leq \frac{1}{k} V(0) \tag{2.45}$$

it is clear that $z \in \mathcal{L}_2^2[0, \infty)$. Additionally, it is straightforward to see that $\dot{z} \in \mathcal{L}_\infty^2$, thus z is uniformly continuous. Then, by Barbalat's lemma (page 323 of [24]), we can conclude that $z \to 0$ as $t \to \infty$, which proves the asymptotic tracking claim in Theorem 2.2.1 Part II.

In (2.43), to see that $\tilde{\boldsymbol{\theta}}_{np}^T \left\{ \boldsymbol{\varphi}_{np} \eta - \boldsymbol{\Gamma}^{-1} \mathrm{Proj}_{\hat{\theta}_{np}}(\boldsymbol{\Gamma} \boldsymbol{\varphi}_{np} \eta) \right\} \leq 0$, recall that $\boldsymbol{\Gamma}$ is a diagonal p.d. matrix $\boldsymbol{\Gamma} = \mathrm{diag}(\gamma_1, \ldots, \gamma_{p_2})$, and note that this vector product can be written as the sum

$$\sum_{i=1}^{p_2} \tilde{\theta}_{np,i} \left[\varphi_{np,i} \eta - \frac{1}{\gamma_i} \mathrm{Proj}_{\theta_{np,i}}(\gamma_i \varphi_{np,i} \eta) \right] \tag{2.46}$$

The adaptation law in (2.19) has three possible cases. The first condition dictates that if $\hat{\theta}_i = \hat{\theta}_{i,max}$ and $\gamma_i \varphi_i \eta > 0$, then $\mathrm{Proj}_{\theta_i}(\gamma_i \varphi_i \eta) = 0$. This also signifies that $\tilde{\theta}_i = \theta_i - \theta_{i,max} \leq 0$, and since $\gamma_i > 0$, the i-th term in (2.46) is $\tilde{\theta}_{np,i} \varphi_{np,i} \eta \leq 0$. Similarly, the second condition dictates that if $\hat{\theta}_i = \hat{\theta}_{i,min}$ and $\gamma_i \varphi_i \eta < 0$, then $\mathrm{Proj}_{\theta_i}(\gamma_i \varphi_i \eta) = 0$. Since $\tilde{\theta}_i = \theta_i - \theta_{i,min} \geq 0$, the i-th term in the summation is also $\tilde{\theta}_{np,i}\varphi_{np,i}\eta \leq 0$. Lastly, from the third condition of (2.19), $\text{Proj}_{\theta_i}(\gamma_i\varphi_i\eta) = \gamma_i\varphi_i\eta$. Therefore the i-th term in (2.46) is simply $\tilde{\theta}_{np,i}(\varphi_{np,i}\eta - \varphi_{np,i}\eta) = 0$. This proves that $\tilde{\boldsymbol{\theta}}_{np}^T \left\{\boldsymbol{\varphi}_{np}\eta - \boldsymbol{\Gamma}^{-1}\text{Proj}_{\hat{\theta}_{np}}(\boldsymbol{\Gamma}\boldsymbol{\varphi}_{np}\eta)\right\} \leq 0$.

■

Remark 2.2.1 *Suppose that the scalar function $g(\bar{z})$ is known but has an uncertain gain on the input (with known bounds). Then we can write $g(\bar{z}) = g_u(\bar{z})\theta_u$, where θ_u is the uncertain parameter on the function g_u. The control law in* (2.34) *is modified to*

$$u = \left(g_u(\bar{z})\hat{\theta}_u(t)\right)^{-1} \bar{u}$$

and the estimate $\hat{\theta}_u$ is included in the vector $\hat{\boldsymbol{\theta}}_{np}^{\star} = [\hat{\boldsymbol{\theta}}_{np}^T, \hat{\theta}_u]^T$ and the adaptation law (2.35)*. Then it is simple to verify that the results from Theorem 2.2.1 hold.*

2.3 Design Example and Simulation Results

To illustrate the ARC-LMI algorithm and its performance, we demonstrate the controller design and simulation results for a single link manipulator. The control of a single link manipulator (or inverted pendulum) with a torque actuator is a well-known problem (e.g. [12], [41], [27]) involving a second-order dynamic system that may have uncertain parameters, such as mass and arm length, and complex nonlinearities, such as Coulomb friction and drag force. These factors make it an excellent case study for the ARC-LMI controller, both for exemplifying the design methodology and for demonstrating the performance of the controller.

Consider the single link manipulator of mass m, length l, drag coefficient c_d, and moment of inertia J. The dynamics of the arm in terms of the angle ϕ can be expressed as

$$J\ddot{\phi} = mgl\sin\phi - c_d\dot{\phi}\left|\dot{\phi}\right| + d + u \tag{2.47}$$

where d is an external disturbance and u is the control torque. For simplicity, assume that the inertia $J = 1.0\ kg{\cdot}m^2$ is known. Then (2.47) can be written in the state-space form, with $x_1 := \phi$ and $x_2 := \dot{\phi}$, as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \theta_1 \sin x_1 - \theta_2 x_2 \left|x_2\right| + d + u \end{aligned} \tag{2.48}$$

where θ_i are the uncertain parameters. Suppose we want to track a desired angle trajectory $x_d(t)$. The error dynamics, with $z_1 := x_1 - x_d$ and $z_2 := \dot{x}_1 - \dot{x}_d$, are given by

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= \theta_1 \sin x_1 - \theta_2 x_2 \left|x_2\right| + d - \ddot{x}_d + u \end{aligned} \tag{2.49}$$

The control input, $u = u_s + u_m + u_r$, with

$$u_m = \ddot{x}_d - \hat{\theta}_1 \sin x_1 + \hat{\theta}_2 x_2 \left|x_2\right| \tag{2.50}$$

leads to the following closed-loop error dynamics for z_2

$$\dot{z}_2 = \tilde{\theta}_1 \sin x_1 - \tilde{\theta}_2 x_2 \left|x_2\right| + d + u_s + u_r \tag{2.51}$$

The first term, for example, can be partially written in the polytopic form (2.9), (2.10) in terms of the error state z_1. To see this, note that $x_1 = z_1 + x_d$, and write

$$\tilde{\theta}_1 \sin x_1 = \tilde{\theta}_1 \left[\sin(z_1 + x_d) - \sin(x_d)\right] + \tilde{\theta}_1 \sin(x_d)$$

Then the bounded uncertain/nonlinear function ψ_1 is given by

$$\psi_1(\bar{z}, t) = \begin{cases} \tilde{\theta}_1 \left[\sin(z_1 + x_d) - \sin(x_d)\right] / z_1 \ , & \text{if } z_1 \neq 0 \\ \tilde{\theta}_1 \ , & \text{if } z_1 = 0 \end{cases} \tag{2.52}$$

$$-\tilde{\theta}_{1,max} \leq \psi_1(\bar{z}, t) \leq \tilde{\theta}_{1,max}$$

where the maximum error in parameter estimate $\tilde{\theta}_{1,max} = \theta_{1,max} - \theta_{1,min}$. The term $\sin(x_d)$ cannot be written in the polytopic form, so it is added to $\boldsymbol{\varphi}_{np}(\bar{z}, t)$. Note that this term cannot be corrupted by measurement noise, as it depends on the known trajectory $x_d(t)$, and not the state. The closed-loop error dynamics can now be written as follows.

$$\dot{\bar{z}} = A(\bar{z},t)\bar{z} + B(\bar{z},t)u_s + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{ \boldsymbol{\varphi}_{np}(\bar{z},t)^T \tilde{\boldsymbol{\theta}}_{np} + d + u_r \right\} \tag{2.53}$$

where,

$$\begin{aligned} A(\bar{z},t) &= A_0 + \psi_1(\bar{z},t)\Delta A_1 \ , \quad A_0 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \ , \quad \Delta A_1 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \\ B(\bar{z},t) &= B_0 + \psi_1(\bar{z},t)\Delta B_1 \ , \quad B_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \ , \quad \Delta B_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{aligned} \tag{2.54}$$

with ψ_1 from (2.52). Then $u_s = LS^{-1}\bar{z}$ can be computed by solving the LMI optimization problem (2.13). If $\tilde{\theta}_{1,max} = 1$, for a convergence rate $\alpha = 10.0$, we obtain $LS^{-1} = K = [-152.9, \ -20.2]$, with the corresponding Lyapunov matrix,

$$P = S^{-1} = \begin{bmatrix} 0.996 & 0.066 \\ 0.066 & 0.007 \end{bmatrix}$$

It emphasized that this static feedback term u_s is optimal for the chosen α with respect to minimizing $||K||^2$, and it stabilizes the nominal system *with* the uncertain polytopic nonlinearity (2.52). These are key advantages of the ARC-LMI controller. Continuing with the controller design, in (2.53) we have

$$\boldsymbol{\varphi}_{np}(\bar{z},t) = \begin{bmatrix} \sin x_d \\ -x_2|x_2| \end{bmatrix} \ , \ \tilde{\boldsymbol{\theta}}_{np} = \begin{bmatrix} \tilde{\theta}_1 \\ \tilde{\theta}_2 \end{bmatrix} \tag{2.55}$$

Therefore, the robust feedback term $u_r = -\rho \tanh\left(\frac{\kappa\rho}{\epsilon}\eta\right)$, where $\rho(\bar{z},t)$ can be written explicitly, using (2.31), as

$$\rho = \tilde{\theta}_{1,max}|\sin x_d| + \tilde{\theta}_{2,max}x_2^2 + \delta \tag{2.56}$$

where $\tilde{\theta}_{i,max} = \theta_{i,max} - \theta_{i,min}$ and $\delta(\bar{z},t) \geq |d(\bar{z},t)|$ for all $\bar{z} \in \mathbb{R}^2, t \geq 0$.

For this example, suppose $\theta_1 = 5$ and $\theta_2 = 1$, and assume that we know the parameters within 10% of their true values, that is

$$\boldsymbol{\theta}_{np} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}, \ \boldsymbol{\theta}_{np,min} = \begin{bmatrix} 4.5 \\ 0.9 \end{bmatrix}, \ \boldsymbol{\theta}_{np,max} = \begin{bmatrix} 5.5 \\ 1.1 \end{bmatrix}$$

To test the robustness to disturbances, let $d = (-1)^{round(t^2)}$, then the upper bound can be chosen as $\delta = 1 \geq |d(t)|$, and therefore $\rho = |\sin x_d| + 0.2x_2^2 + 1$. We begin the simulation with 5% error in each parameter estimate, i.e. $\hat{\boldsymbol{\theta}}_{np}(0) = [5.25, 0.95]^T$, and we remove the disturbance after 15 seconds, i.e. $d(t) = 0$ for $t > 15$. The adaptation rate matrix $\boldsymbol{\Gamma} = \text{diag}([500, 100])$, and the trajectory is selected to be $x_d(t) = 1 - \cos(2\pi t/3)$. The tracking errors are shown in Figure 2.2, as well as the theoretical bound on the norm of the tracking error vector $||z||$. The parameter estimation errors are shown in Figure 2.3.

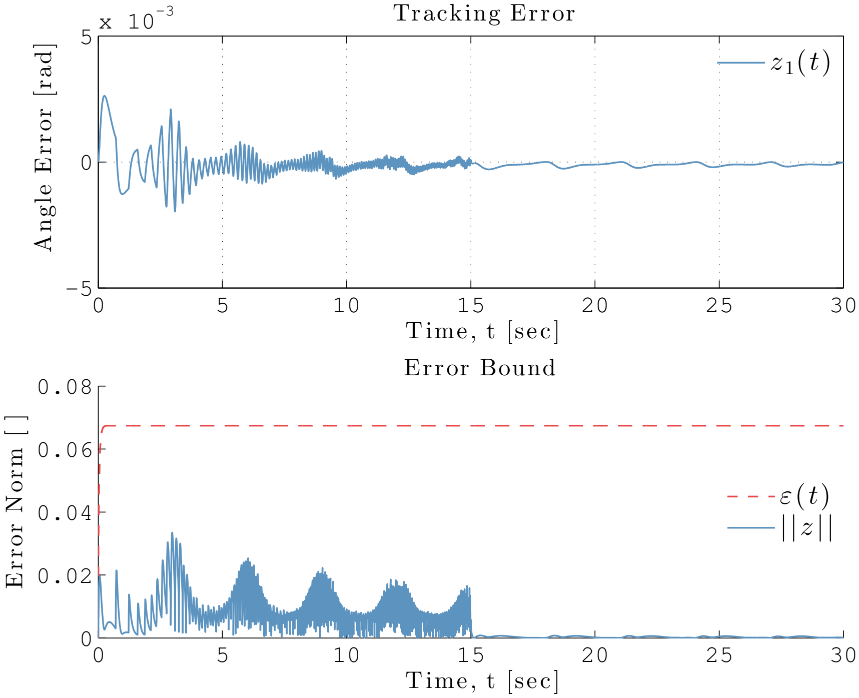


Figure 2.2.. Angle tracking errors $z_1(t)$, and the error vector norm $||z||$ together with the theoretical bound $\varepsilon(t) = \sqrt{\frac{\epsilon}{\lambda\alpha}\left(1 - e^{-2\alpha t}\right)}$

Evidently, the disturbance from 0 to 15 seconds does not damage the transient performance as the tracking error norm remains within the theoretical bound $\varepsilon(t) = \sqrt{\frac{\epsilon}{\lambda\alpha}\left(1 - e^{-2\alpha t}\right)}$ from (2.40). Furthermore, the steady state performance improves, even in the presence of the external disturbance, due to the online adaptation. After 15 seconds, the disturbance is removed and, as expected, near-asymptotic tracking is achieved (within accuracy of the simulation). The control input history is shown in Figure 2.4, together with the previously described external disturbance.

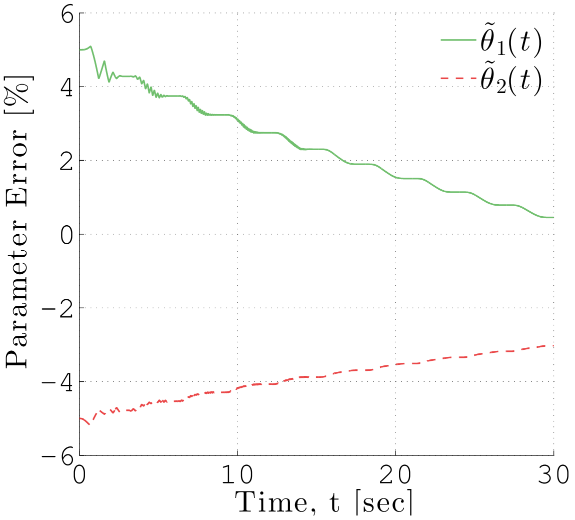


Figure 2.3.. Parameter estimation percentage errors

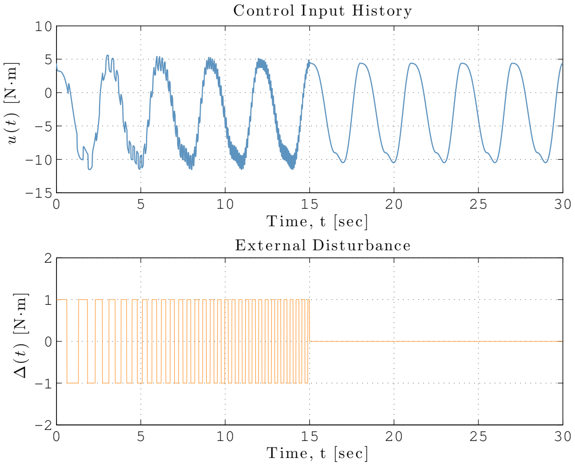


Figure 2.4.. Control input torque in the presence of a disturbance $\Delta(t)$

2.4 Additional Proofs

Proof : Lemma 2.1.1. To see that the control law (2.12) stabilizes the system in (2.8), observe that the closed-loop dynamics are given by $\dot{x} = [A(x,t) + B(x,t)LS^{-1}]x$. Let

$$\begin{aligned} N(x,t) &:= A(x,t)S + B(x,t)L + SA(x,t)^T + L^T B(x,t)^T \\ &= N_0 + \psi_1(x,t)\Delta N_1 + \cdots + \psi_l(x,t)\Delta N_l \end{aligned}$$

where

$$\begin{aligned} N_0 &= A_0 S + B_0 L + SA_0^T + L^T B_0^T \\ \Delta N_i &= \Delta A_i S + \Delta B_i L + S\Delta A_i^T + L^T \Delta B_i^T \ , \ \ \text{for } i = 1, \ldots, l \end{aligned}$$

Since $a_i \le \psi_i(x,t) \le b_i$, then $N(x,t) \le -2\alpha S$ for all t and x if

$$N_0 + \delta_1 \Delta N_1 + \cdots + \delta_l \Delta N_l \le -2\alpha S \ , \qquad \text{for } \delta_i = a_i \text{ or } b_i \ , \ i = 1, \ldots, l$$

which is the first equation in (2.13), for all pairs (A, B) in $\mathcal{AB}$. Then, if $N(x,t) \le -2\alpha S$ for all t and x, we have:

$$A(x,t)S + B(x,t)L + SA(x,t)^T + L^T B(x,t)^T \le -2\alpha S$$

Pre- and post-multiplying the previous inequality by $P = S^{-1}$ yields

$$PA(x,t) + PB(x,t)LS^{-1} + A(x,t)^T P + S^{-1} L^T B(x,t)^T P \le -2\alpha P$$

Using the fact that $P = P^T$, the previous inequality can be arranged as follows:

$$P\,[A(x,t) + B(x,t)LS^{-1}] \ + \ [A(x,t) + B(x,t)LS^{-1}]^T \ P \le -2\alpha P$$

Notice that this corresponds to the closed-loop system, $\dot{x} = [A(x,t) + B(x,t)LS^{-1}]x$. Consider the Lyapunov function $V(x) = x^T P x$, then

$$\begin{aligned} \dot{V} &= 2x^T P \dot{x} \\ &= 2x^T P[A(x,t) + B(x,t)LS^{-1}]x \\ &= x^T P[A(x,t) + B(x,t)LS^{-1}] + [A(x,t) + B(x,t)LS^{-1}]^T P x \\ &\le -2\alpha x^T P x = -2\alpha V(x) \end{aligned}$$

Therefore, the closed-loop system is globally exponentially stable, with the Lyapunov matrix $P = S^{-1}$ and a convergence rate of α.

In addition, the final LMI in the optimization problem (2.13),

$$\begin{pmatrix} \beta I & L^T \\ L & S \end{pmatrix} \geq 0 \tag{2.57}$$

restricts the size of $K := LS^{-1}$ as follows. From the Schur Complement results on Hermitian matrix inequalities (see Theorem 1.12 of [60] and the Appendix of [11]), we can show that (2.57) is true if and only if

$$S > 0 \quad \text{and} \quad \beta I - LS^{-1}L^T \geq 0 \tag{2.58}$$

The second inequality in (2.58) can be written as $KSK^T \leq \beta I$. Since we also require in (2.13) that $S \geq I$, we obtain

$$KK^T \leq KSK^T \leq \beta I$$

which explicitly shows that β serves as an upper bound on $||K||^2$.

■

3. ARC-LMI CONTROLLER DESIGN FOR A QUADROTOR

In this chapter, we implement a novel linear matrix inequality-based nonlinear adaptive robust control algorithm to design the attitude and position controllers for an X-configuration quadrotor helicopter. The inner-loop of the autopilot controls the attitude and altitude of the quadrotor, and the outer-loop controls its position in the earth-fixed coordinate frame. By assuming knowledge of the bounds of the quadrotor's uncertain parameters (e.g. mass and moments of inertia) and predetermined bounds on the external and unstructured disturbances (e.g. wind gusts and unmodeled dynamics), we attain a guaranteed transient and steady state tracking performance. We demonstrate the performance of the controllers via two illustrative examples. In the first example mission, the quadrotor follows an altitude reference trajectory in the presence of wind gusts and delivers a package of uncertain mass mid-flight. The second example presents an autonomous waypoint flight in the presence of constant wind and wind gusts. The results of our simulations indicate that our controller design is useful for a variety of quadrotor applications, including precise trajectory tracking, autonomous waypoint navigation in the presence of disturbances, and package delivery without loss of performance.

3.1 Modeling the Quadrotor

In order to design an effective adaptive and robust controller, an accurate dynamic model of the system is required. In this paper, we consider the 3D Robotics IRIS quadrotor which has an X-configuration [1] for algorithm development and performance validation. However, the proposed algorithms are general, and can be applied to other types of UAS. The dynamic model of the quadrotor is derived based on the following assumptions: (1) the structure is rigid and has roll symmetry, (2) the origin of the body frame is at the center of mass of the quadrotor, and (3) the rotors are rigid in the plane perpendicular to the body z-axis.

Reference Frames & Transformations. The position of the quadrotor in the inertial frame $\mathscr{F}^I$ is defined by the vector $\boldsymbol{\xi} = (N, E, D)^T$, and the attitude in the inertial frame is defined by the vector $\boldsymbol{\eta} = (\phi, \theta, \psi)^T$, where $\phi(t)$ is the roll angle (about the N-axis), $\theta(t)$ is the pitch angle (about the E-axis), and $\psi(t)$ is the yaw angle (about the D-axis), in a right-handed coordinate frame. The linear velocities in the body frame $\mathscr{F}^b$ are defined by the vector $\mathbf{V} = (u, v, w)^T$, and the angular velocities in the body frame are defined by the vector $\boldsymbol{\omega} = (p, q, r)^T$, where $u(t) = \dot{x}$, $v(t) = \dot{y}$, $w(t) = \dot{z}$, $p(t)$ is the roll rate (about the x-axis), $q(t)$ is the pitch rate (about the y-axis), and $r(t)$ is the yaw rate (about the z-axis). The body and inertial reference frames, and their corresponding positive rotations, are shown in Figure 3.1.

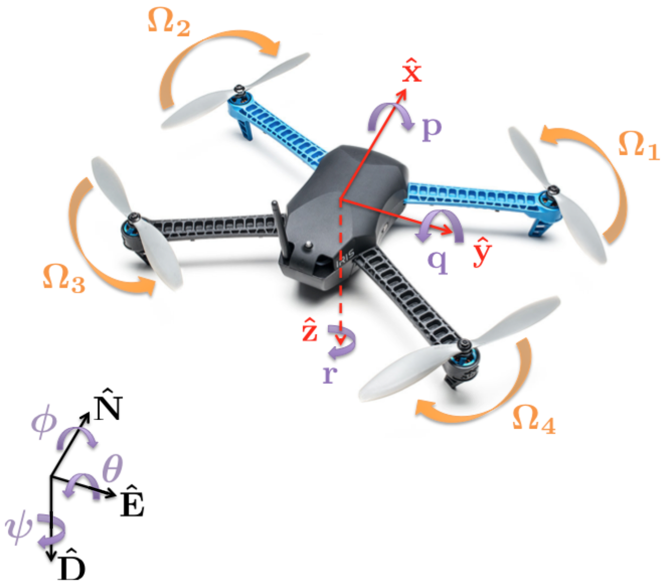


Figure 3.1.. Quadrotor reference frames and positive rotations *(obtained from [2] and annotated)*

We denote the rotation matrix that transforms the a-frame states to the b-frame states as $\mathbf{R}_b^a : \mathscr{F}^a \to \mathscr{F}^b$. The rotation matrix from the earth-fixed inertial frame (N, E, D) to the body-fixed frame (x, y, z) by successive rotations of yaw $(\psi) \to$ pitch $(\theta) \to$ roll (ϕ) is given by,

$$\mathbf{R}_b^I = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{bmatrix} \tag{3.1}$$

where $c\alpha \triangleq \cos(\alpha)$, $s\alpha \triangleq \sin(\alpha)$, $t\alpha \triangleq \tan(\alpha)$. For a derivation of the rotation matrix in (3.1), see Appendix A. The rotation matrix is orthogonal and has its determinant $\det(\mathbf{R}_b^I) = 1$. Therefore, $(\mathbf{R}_b^I)^{-1} = (\mathbf{R}_b^I)^T$. From these properties, we can directly derive the rotation matrix from the body frame (x, y, z) to the inertial frame (N, E, D), $\mathbf{R}_I^b : \mathscr{F}^b \to \mathscr{F}^I$, as follows.

$$\mathbf{R}_I^b = (\mathbf{R}_b^I)^{-1} = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{3.2}$$

The transformations of angular velocities from the inertial frame to the body frame and vice versa are given by,

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & c\theta s\phi \\ 0 & -s\phi & c\theta c\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi / c\theta & c\phi / c\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.3}$$

respectively. Here we assume that the pitch angle $\theta(t) \in (-\pi/2, \pi/2)$ for all t, to avoid the singularity in the second transformation of (3.3).

Force Equations. Using the Newton-Euler equations to describe the dynamics of the quadrotor in the inertial frame, the product of the mass and the inertial acceleration is equal to the sum of the forces acting on the quadrotor, i.e. the weight of the quadrotor mg, the total thrust of the four rotors T, and the drag forces F_w.

$$\begin{aligned} m\ddot{\boldsymbol{\xi}} &= m\mathbf{G} + \mathbf{R}_I^b\mathbf{T} + \mathbf{F_w} \ , \\ \begin{bmatrix} \ddot{N} \\ \ddot{E} \\ \ddot{D} \end{bmatrix} &= g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - \frac{T}{m}\mathbf{R}_I^b \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - \frac{1}{m} \begin{bmatrix} k_s(\dot{N} - w_N)|\dot{N} - w_N| \\ k_s(\dot{E} - w_E)|\dot{E} - w_E| \\ k_u(\dot{D} - w_D)|\dot{D} - w_D| \end{bmatrix} \end{aligned} \tag{3.4}$$

where m is the mass of the quadrotor, g is the gravitational acceleration, and T is the thrust in the $(-z)$ direction. The wind disturbances in the inertial frame are $(w_N, w_E, w_D)^T$ and the constants $k_s, k_u > 0$ are used to model the aerodynamic drag on the side and upper/lower faces of the quadrotor, respectively. Note that some works assume that the drag is linearly proportional to the freestream velocity, i.e. $\mathbf{F_w} \propto \mathbf{V}_\infty$ [16], or regard the wind conditions as unstructured disturbances [6], [5], or simply neglect drag forces altogether [31], [8], [23]. In general, the drag force is proportional to the square of the freestream velocity (as shown in (3.4)), and the aerodynamic coefficients k_s and k_u will depend on the air density and viscosity, and the vehicle orientation, that is $k_s, k_u = f(\rho, \nu, \boldsymbol{\xi})$. In this paper, we use the more accurate representation of the drag force, but we assume that these aerodynamic coefficients are constant for a typical flight.

Given that the on-board sensors measure accelerations in the body frame, it is useful to derive the force equations in this local coordinate frame. The basic kinematic equation (BKE) gives the time rate of change of the body-frame velocities relative to the inertial frame.

$$\frac{\mathrm{d}^I}{\mathrm{d}t}\mathbf{V} = \dot{\mathbf{V}} + {}^I\boldsymbol{\omega}^b \times \mathbf{V}$$

where ${}^{I}\boldsymbol{\omega}^{b}$ is the rotation of the quadrotor in the inertial frame and '×' denotes the standard vector cross-product. Therefore, the force equations in the body frame of the quadrotor are derived as follows.

$$m\dot{\mathbf{V}} = -m(\boldsymbol{\omega} \times \mathbf{V}) + \mathbf{R}_b^I\mathbf{G} + \mathbf{T} + \mathbf{F_w} \ ,$$
$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \mathbf{R}_b^I \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} - \frac{T}{m} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - \frac{1}{m} \begin{bmatrix} k_s(u - w_x)|u - w_x| \\ k_s(v - w_y)|v - w_y| \\ k_u(w - w_z)|w - w_z| \end{bmatrix} \tag{3.5}$$

Moment Equations. The moment equations are expressed in the body frame, as the angular accelerations are measured solely in this frame. From the conservation of angular momentum and Newton's laws of motion, the inertial time derivative of the quadrotor's angular momentum ($\mathbf{H} = \mathbf{I}\boldsymbol{\omega}$) is equal to the external moments on the system, $\boldsymbol{\tau}_{total}$. These moments include the torques and gyroscopic moments due to the rotors, $\boldsymbol{\tau}$ and $\boldsymbol{\Gamma}$ respectively, and the disturbance torques due to wind, $\boldsymbol{\tau_w}$. We apply the BKE when taking the derivative of $\mathbf{H}$, as follows.

$$\frac{\mathrm{d}^I}{\mathrm{d}t}\mathbf{H} = \dot{\mathbf{H}} + {}^{I}\boldsymbol{\omega}^{b} \times \mathbf{H} = \boldsymbol{\tau}_{total}$$
$$\mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}) = \boldsymbol{\tau} + \boldsymbol{\Gamma} + \boldsymbol{\tau_w} \tag{3.6}$$

where the symmetric inertia matrix $\mathbf{I}$ and the gyroscopic moments due to the rotors are given by,

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{xz} & 0 & I_{zz} \end{bmatrix} , \qquad \boldsymbol{\Gamma} = I_r \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Omega_r = \begin{bmatrix} I_r\Omega_r q \\ -I_r\Omega_r p \\ 0 \end{bmatrix} \tag{3.7}$$

respectively. I_r is the rotational moment of inertia about the motor axis and the relative rotor speed, Ω_r, is defined as $\Omega_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4$. The external torques from the rotors are $\boldsymbol{\tau} = (\tau_\phi, \tau_\theta, \tau_\psi)^T$ and wind disturbances are denoted as $\boldsymbol{\tau_w} =$

$(\tau_{w,\phi}, \tau_{w,\theta}, \tau_{w,\psi})^T$. Substituting (3.7) in (3.6) and solving for rotational accelerations yields,

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \mathbf{I}^{-1} \left(- \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx}p - I_{xz}r \\ I_{yy}q \\ I_{zz}r - I_{xz}p \end{bmatrix} + \begin{bmatrix} I_r\Omega_r q \\ -I_r\Omega_r p \\ 0 \end{bmatrix} + \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} + \begin{bmatrix} \tau_{w,\phi} \\ \tau_{w,\theta} \\ \tau_{w,\psi} \end{bmatrix} \right) \tag{3.8}$$

where,

$$\mathbf{I}^{-1} = \gamma^{-1} \begin{bmatrix} I_{zz} & 0 & I_{xz} \\ 0 & \frac{\gamma}{I_{yy}} & 0 \\ I_{xz} & 0 & I_{xx} \end{bmatrix}, \quad \gamma = I_{xx}I_{zz} - I_{xz}^2 \tag{3.9}$$

Navigation Equations. The body frame velocities are related to the inertial frame velocities by the rotation matrix as follows,

$$\begin{bmatrix} \dot{N} \\ \dot{E} \\ \dot{D} \end{bmatrix} = \mathbf{R}_I^b \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{3.10}$$

A more intuitive inertial frame is North-East-Up, where 'Up' is the height of the quadrotor in the flat-Earth inertial frame, denoted by h. The transformation between this frame and the body frame is similar, but given that $\dot{D} = -\dot{h}$, we change the sign of the third row of $\mathbf{R}_I^b$. This yields,

$$\begin{bmatrix} \dot{N} \\ \dot{E} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ s\theta & -c\theta s\phi & -c\theta c\phi \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{3.11}$$

Control Inputs. The quadrotor is an underactuated mechanical system, in which the four motor inputs are used to control the system's six degrees of freedom [23]. The attitude of the quadrotor is coupled with its position, as a roll or pitch angle is required in order to move in the (x, y)-plane. The IRIS quadrotor has an X-configuration, which is symmetric about the x-axis (roll symmetry). We can derive the forces and moments from the dimensions in Figure 3.2 and Newtonian mechanics.

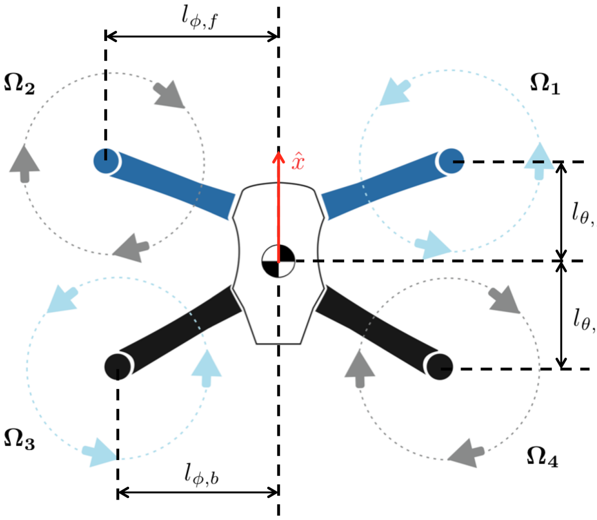


Figure 3.2.. IRIS top-view diagram *(taken from [1] and annotated)*

The control inputs for guiding and stabilizing the quadrotor system are mapped from the four independent motor thrusts to one force and three torques: total thrust force, roll torque, pitch torque, and yaw torque ($T, \tau_\phi, \tau_\theta$, and τ_ψ respectively). As shown in Figure 3.3(a), the total thrust T in the $(-z)$ direction is given by the sum of thrusts from the four rotors.

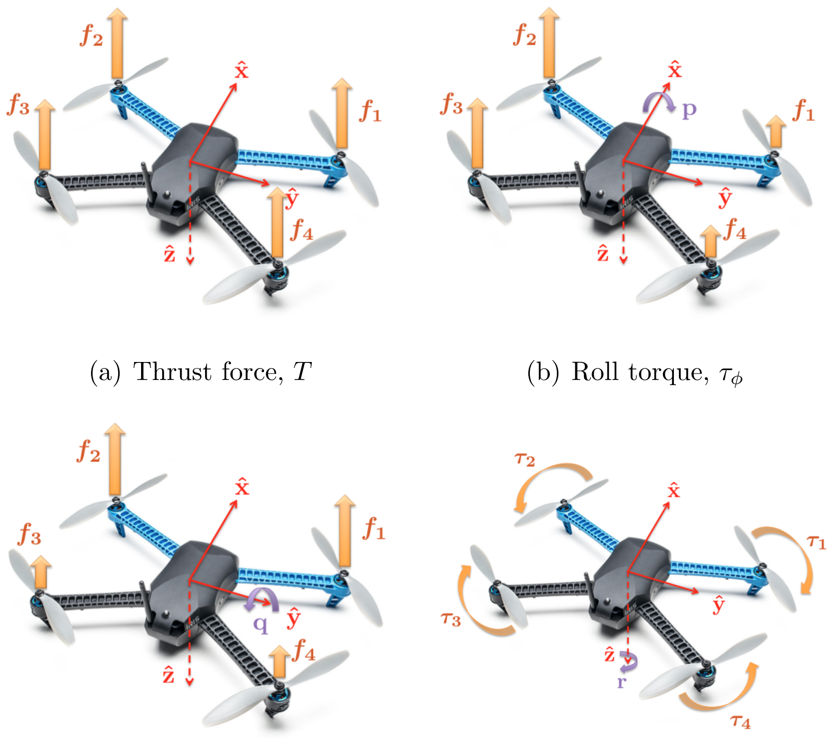
$$T = f_1 + f_2 + f_3 + f_4 \tag{3.12}$$

From Figure 3.3(b), a positive roll torque (roll right) is generated by increasing the thrust in the left motors (motors 2 and 3) and/or decreasing the thrust in the right motors (motors 1 and 4).

$$\tau_\phi = l_{\phi,f}(-f_1 + f_2) + l_{\phi,b}(f_3 - f_4) \tag{3.13}$$

As shown in Figure 3.3(c), a positive pitch torque (‘pitch up’) is generated by increasing thrust in the front motors (motors 1 and 2) and/or decreasing thrust in the back motors (motors 3 and 4).

$$\tau_\theta = l_{\theta,f}(f_1 + f_2) - l_{\theta,b}(f_3 + f_4) \tag{3.14}$$



(a) Thrust force, T

(b) Roll torque, τ_ϕ

(c) Pitch torque, τ_θ

(d) Yaw torque, τ_ψ

Figure 3.3.. Mapping motor inputs to control inputs

The drag force on the rotors produces a yawing torque on the quadrotor in the opposite direction of the rotor's rotation. Therefore, a positive yaw torque (clockwise from top-view) is generated by increasing the first and third motor speeds and/or decreasing the second and fourth motor speeds (as seen in Figure 3.3(d)).

$$\tau_\psi = \tau_1 + \tau_2 + \tau_3 + \tau_4 \tag{3.15}$$

Rotor Dynamics. The relationship between the thrust force and the rotor angular velocity is complex and varies with the angle of attack and the quadrotor velocity relative to free stream [19]. It is further affected by blade flapping, where the leading blade experiences a higher relative velocity than the retreating blade, a phenomenon which essentially tilts the thrust vector away from its original $(-z)$ direction. These effects are difficult to model and have little influence during near-hover flight conditions, i.e. small angles of attack and low speeds. Therefore, as in most research papers that study the modeling and control of quadrotors [23], [34], the following simplified relation between the i-th rotor's thrust f_i and the rotor's angular velocity Ω_i is made.

$$f_i = k_t \Omega_i^2 \tag{3.16}$$

In general, the thrust constant $k_t > 0$ depends on the density of air, rotor radius, blade shape, blade flapping, and flight regime [28]. Here we assume that it remains nearly constant throughout the flight. Similarly, the relationship between the rotor's angular velocity and the counter-torque due to drag on the propeller is quite complex (see [16]), but if we assume near-hover conditions, it can be simplified as follows.

$$\tau_i = b\Omega_i^2 \tag{3.17}$$

where $b > 0$ is the drag factor on the propeller, assumed to be nearly constant for the flight duration. Note that the robust feedback in the ARC-LMI controller overcomes the simplifications in (3.16) and (3.17), as explained in Section 3.2.1.

Motor Mixing. By substituting the approximations from (3.16) and (3.17) into the force and torque equations (3.12)-(3.15), we have the following system of equations that relates the virtual commands of force or torque to equivalent rotor angular velocities.

$$\begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} k_t & k_t & k_t & k_t \\ -l_{\phi,f}k_t & l_{\phi,f}k_t & l_{\phi,b}k_t & -l_{\phi,b}k_t \\ l_{\theta,f}k_t & l_{\theta,f}k_t & -l_{\theta,b}k_t & -l_{\theta,b}k_t \\ b & -b & b & -b \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{3.18}$$

The transformation from commanded forces and torques to the motor control inputs is obtained by inverting (3.18), and is given by,

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} l_{\theta,b}/c_1 & -1/c_2 & 1/c_1 & l_{\phi,b}/c_3 \\ l_{\theta,b}/c_1 & 1/c_2 & 1/c_1 & -l_{\phi,b}/c_3 \\ l_{\theta,f}/c_1 & 1/c_2 & -1/c_1 & l_{\phi,f}/c_3 \\ l_{\theta,f}/c_1 & -1/c_2 & -1/c_1 & -l_{\phi,f}/c_3 \end{bmatrix} \begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \tag{3.19}$$

where

$$c_1 = 2k_t(l_{\theta,f} + l_{\theta,b}) \tag{3.20}$$

$$c_2 = 2k_t(l_{\phi,f} + l_{\phi,b}) \tag{3.21}$$

$$c_3 = 2b(l_{\phi,f} + l_{\phi,b}) \tag{3.22}$$

Since this transformation produces the desired motor velocities from the virtual control inputs, it is sometimes referred to as 'motor mixing' [43, 26]. The control inputs $T, \tau_\phi, \tau_\theta$, and τ_ψ are designed in order to stabilize the attitude and position of the quadrotor. From the rotor dynamics in (3.16)-(3.17) and the static motor mixing matrix in (3.19), the control inputs are translated into commanded angular velocities for each motor/rotor pair. Note that for the purpose of the controller design, we assume that the motor dynamics are fast enough to be neglected. However, for a more accurate simulation of the complete quadrotor dynamics, the motor dynamics are also included. From the system identification performed in [31], the rotor speed Ω_i is related to the desired rotor speed $\Omega_{i,d}$ by the following first-order differential equation.

$$\dot{\Omega}_i = k_m \left(\Omega_{i,d} - \Omega_i\right) \tag{3.23}$$

where the motor gain k_m depends on the inertia and drag of the propeller, as well as the mechanical and electrical characteristics of the motor. This completes the full dynamic model of the quadrotor. For a concise summary of the quadrotor equations of motion, see Appendix B. In the following section, we present the nonlinear adaptive and robust controller architecture.

3.2 Controller Architecture

In this section, we describe the controller design for the autonomous quadrotor. We begin with an overview of controller architecture, and, in Section 3.2.1, define the linear matrix inequality-based nonlinear adaptive robust control (ARC-LMI) theory. In Sections 3.2.2 and 3.2.3, we provide the detailed control input designs for the attitude-altitude inner-loop and the position tracking outer-loop systems, respectively. In Section 3.2.4, we develop the trajectory generation technique for improved position and altitude tracking.

The proposed controllers are based on a cascaded structure, such that the inner-loop controls the faster attitude dynamics, and the outer-loop controls the slower position dynamics by generating commanded angles to the inner-loop controller. From a practical point of view, the on-board accelerometers and gyroscopes that make up the inertial measurement unit (IMU) of the quadrotor provide local measurements at a very fast rate (100-400 kHz), and are used in the inner-loop for stabilizing the attitude and altitude. The inertial navigation system (INS) receives GPS measurements at a slower rate (5 Hz), corrects the IMU measurement error with an extended Kalman filter algorithm, and provides inertial position measurements. These measurements, then, are used in the outer-loop controllers for trajectory tracking. A schematic of the controller architecture is shown in Figure 3.4.

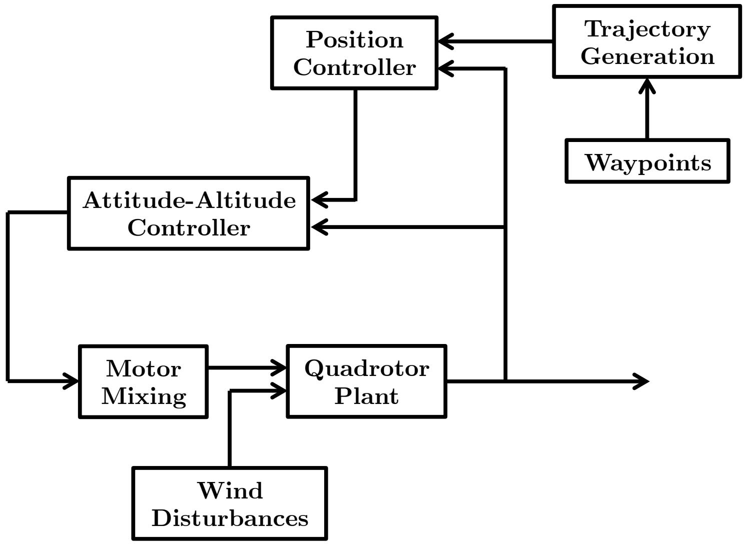


Figure 3.4.. The cascaded controller architecture

The controllers are designed by the linear matrix inequality-based adaptive robust control (ARC-LMI) approach, which guarantees a robust transient performance and improved steady state tracking via an online learning scheme. In addition, the trajectory generator transforms the discrete set of waypoints into a smooth three-dimensional path by an exogenous system that improves the trajectory tracking performance and enables autonomous waypoint navigation.

3.2.1 ARC-LMI Control

The ARC-LMI control algorithm from Section 2.2 is concisely re-stated here for reference. Consider the second-order uncertain system given by,

$$\ddot{x} = f(\bar{x}, t) + g(\bar{x}, t)\, u \tag{3.24}$$

and define the tracking errors $z = [z_1,\, z_2]^T$, with $z_1 = x - x_d(t)$ and $z_2 = \dot{x} - \dot{x}_d(t)$, where $x_d(t)$ is the desired trajectory. Then the system in (3.24) has the following error dynamics.

$$\dot{z} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{ \boldsymbol{\varphi}_p^T \boldsymbol{\theta}_p + \boldsymbol{\varphi}_{np}^T \boldsymbol{\theta}_{np} + \Delta(z,t) - \ddot{x}_d(t) + g(z,t)\, u \right\} \tag{3.25}$$

Here $\boldsymbol{\varphi}_p \in \mathbb{R}^{p_1}$ is a vector of known basis functions that are linearly parametrized by unknown weights $\boldsymbol{\theta}_p \in \mathbb{R}^{p_1}$ and can be written in a special polytopic form, described later. $\boldsymbol{\varphi}_{np} \in \mathbb{R}^{p_2}$ is a vector of known basis functions that are linearly parametrized by unknown weights $\boldsymbol{\theta}_{np} \in \mathbb{R}^{p_2}$ but are non-polytopic, and $\Delta(z,t) \in \mathbb{R}$ is a nonlinear function which captures external disturbances, unmodeled dynamics, and terms that cannot be linearly parametrized. Although the uncertain parameters and external disturbances are unknown, we assume to have information about their bounds. This assumption is relatively mild and is stated as follows.

Assumption 3.2.1 *The uncertain parameters* $\boldsymbol{\theta} = [\boldsymbol{\theta}_p^T, \boldsymbol{\theta}_{np}^T]^T \in \mathbb{R}^l$ *lie in a known bounded region* Ω_θ *and the disturbances* Δ *are bounded by a known function* $\delta(z,t)$*, that is,*

$$\boldsymbol{\theta} \in \Omega_\theta \triangleq \{\boldsymbol{\theta} : \boldsymbol{\theta}_{min} < \boldsymbol{\theta} < \boldsymbol{\theta}_{max}\} \tag{3.26}$$

$$\Delta \in \Omega_\Delta \triangleq \{\Delta : |\Delta(z,t)| \leq \delta(z,t)\} \tag{3.27}$$

where the lower and upper bounds of the parameters $\boldsymbol{\theta}_{min} = [\theta_{1,min}, \ldots, \theta_{l,min}]^T$ *and* $\boldsymbol{\theta}_{max} = [\theta_{1,max}, \ldots, \theta_{l,max}]^T$ *and the function* $\delta(t,x)$ *are known.*

We propose the following control law, which is composed of three parts: u_s, a stabilizing state feedback and compensation for polytopic uncertainties, u_m, a dynamic model compensation term, and u_r, a robust control law, as shown in (3.28).

$$u = g(z,t)^{-1}(u_s + u_m + u_r) \tag{3.28}$$

The adaptive model compensation term u_m uses parameter estimates $\hat{\boldsymbol{\theta}}_p$ and $\hat{\boldsymbol{\theta}}_{np}(t)$, which is updated online via an adaptation law, to be specified later.

$$u_m = \ddot{x}_d(t) - \boldsymbol{\varphi}_p^T \hat{\boldsymbol{\theta}}_p - \boldsymbol{\varphi}_{np}^T \hat{\boldsymbol{\theta}}_{np}(t) \tag{3.29}$$

Thus, the closed-loop system is given by

$$\dot{z} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{ u_s + \boldsymbol{\varphi}_p^T \tilde{\boldsymbol{\theta}}_p \right\} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{ u_r + \boldsymbol{\varphi}_{np}^T \tilde{\boldsymbol{\theta}}_{np}(t) + \Delta(z,t) \right\} \tag{3.30}$$

where $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}$ is the error in the parameter estimates. The stabilizing feedback u_s, which is robust to polytopic uncertainty, is given by,

$$u_s = Kz = LS^{-1}z \tag{3.31}$$

where the symmetric positive definite (s.p.d.) matrix S and the matrix L are solutions to the optimization problem in (3.34), with a desired convergence rate α, and β as an upper bound on $||K||^2$ for the system

$$\dot{z} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \boldsymbol{\varphi}_p^T \tilde{\boldsymbol{\theta}}_p + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_s = A(z,t) + B(z,t)u_s \tag{3.32}$$

where $A(z,t)$ and $B(z,t)$ can be written in the following polytopic form.

$$\begin{aligned} A(x,t) &= A_0 + \psi_1(x,t)\Delta A_1 + \cdots + \psi_l(x,t)\Delta A_l \\ B(x,t) &= B_0 + \psi_1(x,t)\Delta B_1 + \cdots + \psi_l(x,t)\Delta B_l \\ a_i \leq \psi_i&(x,t) \leq b_i \end{aligned} \tag{3.33}$$

and $A_0,\ \Delta A_i \in \mathbb{R}^{(2\times 2)}$ and $B_0,\ \Delta B_i \in \mathbb{R}^{(2\times 1)}$ are constant matrices for $i = 1, \ldots, l$, and ψ_i are scalar-valued functions with known bounds, a_i and b_i. Note that ψ_i can be a bounded nonlinear function *or* an uncertain parameter that lies within known bounds. The LMI-based optimization problem is stated below.

$$\text{Minimize } \beta \text{ subject to:} \qquad \left\{ \begin{array}{rcl} AS + BL + SA^T + L^T B^T + 2\alpha S & \leq & 0 \\ S & \geq & I \\ \begin{pmatrix} \beta I & L^T \\ L & S \end{pmatrix} & \geq & 0 \end{array} \right. \tag{3.34}$$

for all (A, B) in $\mathcal{AB}$

where the 2^l pairs of (A, B) in the convex set $\mathcal{AB}$ are defined below:

$$\mathcal{AB} = \{(A_0 + \delta_1 A_1 + \cdots + \delta_l A_l, \quad B_0 + \delta_1 B_1 + \cdots + \delta_l B_l) : \delta_i = a_i \text{ or } b_i \text{ for } i = 1, \ldots, l\}$$

From this we also obtain the s.p.d Lyapunov matrix $P = S^{-1}$. The robust feedback term u_r overcomes the parametric uncertainties associated with non-polytopic terms and the disturbances, and is given by,

$$\begin{aligned} u_r &= -\rho(z,t) \tanh\left(\frac{\kappa\rho(z,t)}{\epsilon}\eta\right) \\ \rho(z,t) &\geq \left|\boldsymbol{\varphi}_{np}^T \tilde{\boldsymbol{\theta}}_{np} + \Delta(z,t)\right| \\ \eta &= p_{12} z_1 + p_{22} z_2 \\ \kappa &= 0.2785 \end{aligned} \tag{3.35}$$

with p_{ij} from the Lyapunov matrix P. Then, from (3.28), (3.29), (3.31), and (3.35), we have the following complete control law.

$$\begin{aligned} u &= g(z,t)^{-1}\,\bar{u} \\ \bar{u} &= \ddot{x}_d(t) - \boldsymbol{\varphi}_p^T\hat{\boldsymbol{\theta}}_p - \boldsymbol{\varphi}_{np}^T\hat{\boldsymbol{\theta}}_{np}(t) + LS^{-1}z - \rho(z,t)\tanh\left(\frac{\kappa\rho(z,t)}{\epsilon}\eta\right) \end{aligned} \tag{3.36}$$

The discontinuous projection-based adaptation law that completes the ARC-LMI algorithm is given by,

$$\begin{aligned} \dot{\hat{\boldsymbol{\theta}}}_{np} &= \mathrm{Proj}_{\hat{\theta}_{np}}(\boldsymbol{\Gamma}\boldsymbol{\varphi}_{np}\eta) \\ \mathrm{Proj}_{\hat{\theta}_i}(\bullet_i) &= \begin{cases} 0 & \text{if } \begin{cases} \hat{\theta}_i = \hat{\theta}_{i,max} \text{ and } \bullet_i > 0 \\ \hat{\theta}_i = \hat{\theta}_{i,min} \text{ and } \bullet_i < 0 \end{cases} \\ \bullet_i & \text{otherwise} \end{cases} \end{aligned} \tag{3.37}$$

where $\boldsymbol{\Gamma}$ is a diagonal positive definite matrix. The adaptation law (3.37) guarantees that $\hat{\boldsymbol{\theta}}(t) \in \Omega_\theta$ for all t, and therefore $\tilde{\boldsymbol{\theta}}(t)$ is bounded. Theorem 3.2.1 summarizes the results obtained from the ARC-LMI controller design.

Theorem 3.2.1 (ARC-LMI) *Given the uncertain second-order dynamic system* (3.24) *with the error dynamics shown in* (3.25) *and under Assumption 3.2.1, the nonlinear ARC-LMI control input* (3.36) *together with the adaptation law* (3.37) *yield the following results:*

I. All signals are bounded and the tracking error is guaranteed to exponentially converge to a ball of constant radius at a rate of convergence no less than α. The transient performance is prescribed and can be improved by increasing α and decreasing ϵ.

II. If the system is only subject to parametric uncertainties after some time t_0, i.e. $\Delta(z,t) = 0,\ \forall t \geq t_0$, then asymptotic tracking is guaranteed in addition to the results in Part I.

For a proof of Theorem 3.2.1, see pages 18 and 19.

3.2.2 Attitude-Altitude Control

The roll angle dynamics from (3.38) can be written explicitly as,

$$\dot{p} = c_1 pq + c_2 qr + c_3 \Omega_r q + c_4 \tau_\phi + \Delta \tag{3.38}$$

where,

$$\begin{aligned} c_1 &= \frac{I_{xz}}{\gamma}(I_{xx} - I_{yy} + I_{zz}) \ , & c_3 &= \frac{I_{zz} I_r}{\gamma} \\ c_2 &= \frac{1}{\gamma}(I_{yy} I_{zz} - I_{zz}^2 - I_{xz}^2) \ , & c_4 &= \frac{I_{zz}}{\gamma} \\ \Delta &= \frac{1}{\gamma}(I_{xz}\tau_\psi + I_{zz}\tau_{w,\phi} + I_{xz}\tau_{w,\psi}) \ , & \gamma &= I_{xx} I_{zz} - I_{xz}^2 \end{aligned}$$

From (3.3), we have the relation $\dot{\phi} = p + (q \sin\phi + r\cos\phi)\tan\theta$. Therefore, when ϕ and θ are small, $\dot{\phi} \approx p$. Furthermore, assuming small angular rates, $\ddot{\phi} \approx \dot{p}$. In order to track a constant reference roll angle, ϕ_d, we define the tracking error, $z_1 := \phi - \phi_d$, and the rate error $z_2 := \dot{\phi} - \dot{\phi}_d = \dot{\phi}$. Then we can write the second-order uncertain error dynamics of (3.38) as follows.

$$\dot{z} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \{c_1 pq + c_2 qr - c_3 \Omega_r q + \Delta\} + \begin{bmatrix} 0 \\ c_4 \end{bmatrix} \tau_\phi \tag{3.39}$$

Here $z = [z_1, \ z_2]^T$ is the error vector, Δ represents the lumped disturbances (including the coupling effect from the yaw controller), and τ_ϕ is the control input that will be designed to track ϕ_d. Separating the polytopic terms from the non-polytopic ones, we can express (3.39) in the form that is used for the ARC-LMI controller design method, that is,

$$\dot{z} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{\boldsymbol{\varphi}_p^T \boldsymbol{\theta}_p + \boldsymbol{\varphi}_{np}^T \boldsymbol{\theta}_{np} + \Delta\right\} + \begin{bmatrix} 0 \\ c_4 \end{bmatrix} \tau_\phi \tag{3.40}$$

where,

$$\begin{aligned} \boldsymbol{\varphi}_p &= pq \ , & \boldsymbol{\theta}_p &= c_1 \\ \boldsymbol{\varphi}_{np} &= [qr, \ \Omega_r q]^T \ , & \boldsymbol{\theta}_{np} &= [c_2, \ c_3]^T \end{aligned} \tag{3.41}$$

From Assumption 3.2.1, we know the bounds of the uncertain parameters $\boldsymbol{\theta} := [\boldsymbol{\theta}_p, \boldsymbol{\theta}_{np}]^T$, and the bound on the lumped disturbance term Δ. As there is an uncertain gain c_4 on the input τ_ϕ, we re-write (3.40) using the parameter estimation error $\tilde{c}_4 = c_4 - \hat{c}_4$, as follows.

$$\dot{z} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{ \boldsymbol{\varphi}_p^T \boldsymbol{\theta}_p + \boldsymbol{\varphi}_{np}^T \boldsymbol{\theta}_{np} + \tilde{c}_4 \tau_\phi + \Delta \right\} + \begin{bmatrix} 0 \\ \hat{c}_4 \end{bmatrix} \tau_\phi \tag{3.42}$$

Continuing with the ARC-LMI procedure, we propose the control law,

$$\tau_\phi = \frac{1}{\hat{c}_4} \left(u_m + u_s + u_r \right) \tag{3.43}$$

and begin with the adaptive model compensation term u_m,

$$u_m = -\boldsymbol{\varphi}_p^T \hat{\boldsymbol{\theta}}_p - \boldsymbol{\varphi}_{np}^T \hat{\boldsymbol{\theta}}_{np} \tag{3.44}$$

where $\hat{\boldsymbol{\theta}}$ are the parameter estimates. This yields the following closed-loop error dynamics.

$$\dot{z} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{ \boldsymbol{\varphi}_p^T \tilde{\boldsymbol{\theta}}_p + u_s \right\} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{ \boldsymbol{\varphi}_u^T \tilde{\boldsymbol{\theta}}_u + \Delta + u_r \right\} \tag{3.45}$$

where $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}$ is the error in parameter estimates, $\boldsymbol{\varphi}_u^T = [\boldsymbol{\varphi}_{np}^T,\, \tau_\phi]$ and $\boldsymbol{\theta}_u^T = [\boldsymbol{\theta}_u^T,\, c_4]^T$. Then, the first half of the right-hand-side of (3.45) can be written in the polytopic form,

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{ \boldsymbol{\varphi}_p^T \tilde{\boldsymbol{\theta}}_p + u_s \right\} = A(z,t)z + B(z,t)u_s \tag{3.46}$$

where,

$$\begin{aligned} A(z,t) &= A_0 + \psi_1(z,t)\Delta A_1 \\ B(z,t) &= B_0 + \psi_1(z,t)\Delta B_1 \end{aligned} \tag{3.47}$$

with,

$$A_0 = \Delta A_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \;,\; B_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \;,\; \Delta B_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \psi_1 = \tilde{c}_1 q \;,\; a_1 \leq \psi_1 \leq b_1 \tag{3.48}$$

The bounds a_1 and b_1 are known from the bounds on $\tilde{c}_1$ and the physical bound on the pitch rate q. The stabilizing feedback term u_s, then, can be computed from the LMI-based optimization problem (3.34), by choosing a desired convergence rate $\alpha > 0$, such that

$$u_s = LS^{-1}z = Kz \tag{3.49}$$

stabilizes the nominal system *with* polytopic uncertainties, i.e. the system in (3.46). A benefit of the ARC-LMI controller, compared to the traditional ARC method, is that the optimization problem minimizes β, which is an upper bound on the norm of the gain matrix $||K||^2$, thereby reducing the control effort and the possibility of actuator saturation. The second half of the right-hand-side of (3.45) is stabilized via a continuous approximation of the signum function, namely,

$$u_r = -\rho \tanh\left(\frac{\kappa\rho}{\epsilon}\eta\right) \tag{3.50}$$

where,

$$\begin{aligned} \rho(z,t) &= \left|\boldsymbol{\varphi}_{np}\right|^T \, \left|\boldsymbol{\theta}_{np,max} - \boldsymbol{\theta}_{np,min}\right| + \delta(z,t) \\ \delta(z,t) &\geq \left|\Delta\right|, \; \forall t, z \\ \kappa &= 0.2785 \\ \eta &= p_{12}z_1 + p_{22}z_2 \end{aligned} \tag{3.51}$$

the scalar $\epsilon > 0$ is a design parameter that relates to the bound on the tracking error, and p_{ij} is from the Lyapunov matrix $P = S^{-1}$ (obtained from the computation of u_s). Then, the complete control law from (3.43), (3.44), (3.49), and (3.50), which can be written as

$$\tau_\phi = \frac{1}{\hat{c}_4}\left(-\boldsymbol{\varphi}_p^T\hat{\boldsymbol{\theta}}_p - \boldsymbol{\varphi}_{np}^T\hat{\boldsymbol{\theta}}_{np} + Kz - \rho \tanh\left(\frac{\kappa\rho}{\epsilon}\eta\right)\right) \tag{3.52}$$

yields a tracking error the converges at a rate of α to a known bounded norm of the error, which is prescribed by the controller design parameters α and ϵ. Furthermore, with the discontinuous projection-based adaptation law,

$$\dot{\hat{\boldsymbol{\theta}}}_u = \text{Proj}_{\hat{\theta}_u}(\boldsymbol{\Gamma}\boldsymbol{\varphi}_u\eta) \tag{3.53}$$

the steady state performance of the controller is improved, and in the absence of disturbances, asymptotic tracking is guaranteed. Note that with the ARC-LMI controller design we have reduced the number of parameters in the adaptation from four parameters (as would have been required by the traditional ARC) to the three uncertain parameters in $\boldsymbol{\theta}_u$. The pitch and yaw controllers (τ_θ and τ_ψ, respectively) are designed in a similar fashion for the dynamics from (3.8), namely

$$\dot{q} = c_{1,q}pr + c_{2,q}(r^2 - p^2) - c_{3,q}\Omega_r p + c_{4,q}\tau_\theta + \Delta_q \tag{3.54}$$

$$\dot{r} = c_{1,r}pq + c_{2,r}qr + c_{3,r}\Omega_r q + c_{4,r}\tau_\psi + \Delta_r \tag{3.55}$$

respectively. The parameters are given by

$$\begin{aligned} c_{1,q} &= \frac{1}{I_{yy}}(I_{zz} - I_{xx}) \ , & c_{3,q} &= \frac{I_r}{I_{yy}} \\ c_{2,q} &= \frac{I_{xz}}{I_{yy}} \ , & c_{4,q} &= \frac{1}{I_{yy}} \\ \Delta_q &= \frac{\tau_{w,\theta}}{I_{yy}} \ , \end{aligned}$$

and

$$\begin{aligned} c_{1,r} &= \frac{1}{\gamma}(I_{xx}^2 - I_{xx}I_{yy} - I_{xz}^2) \ , & c_{3,r} &= \frac{I_{xz}I_r}{\gamma} \\ c_{2,r} &= \frac{I_{xz}}{\gamma}(-I_{xx} + I_{yy} - I_{zz}) \ , & c_{4,r} &= \frac{I_{xx}}{\gamma} \\ \Delta_r &= \frac{1}{\gamma}(I_{xz}\tau_\phi + I_{xz}\tau_{w,\phi} + I_{xx}\tau_{w,\psi}) \ , & \gamma &= I_{xx}I_{zz} - I_{xz}^2 \end{aligned}$$

The the three controllers τ_ϕ, τ_θ, and τ_ψ provide the inner-loop attitude control for tracking the desired angles ϕ_d, θ_d, and ψ_d respectively, which are generated by the outer-loop position controller (described in Section 3.2.3). The final part of the inner-loop control is the altitude controller. From (3.4), the altitude dynamics can be written in the following form.

$$\ddot{h} = -g - c_h(\dot{h} - w_h)|\dot{h} - w_h| + \frac{\cos\theta\cos\phi}{m}T \tag{3.56}$$

where $c_h = k_u/m$ and w_h is the upward wind in the inertial frame. Here we assume that drag coefficient of the quadrotor does not vary for small angles of pitch and roll, yet we consider the effect of this drag force, which acts as a damping term on the acceleration of the quadrotor in the vertical axis. In order to track a time-varying desired altitude trajectory $h_d(t)$, define the error states, $z = [z_1,\, z_2]^T$, with $z_1 = h - h_d$ and $z_2 = \dot{h} - \dot{h}_d$. Then the error dynamics of (3.56) can be written in the following form.

$$\dot{z} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left\{ -g - c_h \dot{h}|\dot{h}| - \ddot{h}_d + \Delta_h \right\} + \begin{bmatrix} 0 \\ c_m \end{bmatrix} \cos\theta \cos\phi\, T \tag{3.57}$$

where, $c_m = 1/m$ and $\Delta_h = c_h \dot{h}|\dot{h}| - c_h(\dot{h} - w_h)|\dot{h} - w_h|$. Note that we have separated the low frequency component of the drag disturbance for improved estimation of the parameter c_h. The control input T is the combined thrust of the four rotors, and can be designed as follows. Let,

$$T = \frac{1}{\hat{c}_m \cos\theta \cos\phi}(u_m + u_s + u_r) \tag{3.58}$$

The adaptive model compensation term u_m is given by,

$$u_m = \ddot{h}_d - \boldsymbol{\varphi}^T \hat{\boldsymbol{\theta}} \tag{3.59}$$

where,

$$\boldsymbol{\varphi} = \begin{bmatrix} -g \\ -\dot{h}|\dot{h}| \end{bmatrix}^T, \quad \hat{\boldsymbol{\theta}} = \begin{bmatrix} 1 \\ \hat{c}_h \end{bmatrix} \tag{3.60}$$

The stabilizing feedback term $u_s = Kz$ is computed as in (3.49), for the system,

$$\begin{aligned} A(z,t) = A_0 &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \\ B(z,t) = B_0 &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{aligned} \tag{3.61}$$

The robust feedback term $u_r = -\rho \tanh\left(\frac{\kappa\rho}{\epsilon}\eta\right)$ is obtained as in (3.50) and the projection-based adaptation from (3.53) is included, both using the vectors $\boldsymbol{\varphi}_u^T = [\boldsymbol{\varphi}^T,\, T]$ and $\hat{\boldsymbol{\theta}}_u^T = [\hat{\boldsymbol{\theta}}^T,\, \hat{c}_m]$. For improved transient performance, the desired altitude trajectory $h_d(t)$ is generated via a smooth function and using trajectory initialization to ensure $z_1(0) = z_2(0) = 0$ at each waypoint. This procedure is detailed in Section 3.2.4. This concludes the inner-loop controllers, namely τ_ϕ, τ_θ, τ_ψ, and T. In the following section we describe the ARC method used for position tracking control of the quadrotor in the (N, E)-plane of the flat-earth coordinate system.

3.2.3 Position Tracking Control

The outer-loop position controller in the schematic of Figure 3.4 receives the quadrotor states, e.g. angles and velocities, and the desired trajectory, i.e. $N_d(t)$ and $E_d(t)$, and computes the required roll, pitch, and yaw angles. In order to control the position of the quadrotor and to track a time varying trajectory in the (N, E)-plane, we return to the dynamic equations in (3.4).

$$\ddot{N} = -\frac{T}{m}\left(c\psi s\theta c\phi + s\psi s\phi\right) - \frac{k_s}{m}\left(\dot{N} - w_N\right)\left|\dot{N} - w_N\right| \tag{3.62}$$

$$\ddot{E} = -\frac{T}{m}\left(s\psi s\theta c\phi - c\psi s\phi\right) - \frac{k_s}{m}\left(\dot{E} - w_E\right)\left|\dot{E} - w_E\right| \tag{3.63}$$

Note that here the thrust input T is predetermined by the altitude controller (3.58), therefore the control of N and E is actually derived from the commanded angles ϕ and θ. Physically, this is explained by the fact that the quadrotor must roll or pitch in order to rotate its thrust vector in the direction of desired motion. We define the error dynamics between the North position N and the desired position N_d as follows,

$$z_N = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} N - N_d \\ \dot{N} - \alpha_N \end{bmatrix} \tag{3.64}$$

where $\alpha_N = \dot{N}_d - k_1 z_1$ is a virtual control input with $k_1 > 0$, as is required in the backstepping controller design approach. By stabilizing $\dot{z}_2$, i.e. when $z_2 \to 0$, then $\dot{N} = \alpha_N$, and therefore $\dot{z}_1 = \alpha_N - \dot{N}_d = -k_1 z_1$, which is stable, meaning that the tracking error converges to zero, i.e. $z_1 \to 0$. Similarly, for the error dynamics between the East position E and the desired position E_d, define,

$$z_E = \begin{bmatrix} z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} E - E_d \\ \dot{E} - \alpha_E \end{bmatrix} \tag{3.65}$$

where $\alpha_E = \dot{E}_d - k_3 z_3$ is a virtual control input with $k_3 > 0$. Then the second-order error dynamics for the North and East positions are written as follows.

$$\dot{z}_2 = -\frac{T}{m}\left(c\psi s\theta c\phi + s\psi s\phi\right) - \frac{k_s}{m}\dot{N}\left|\dot{N}\right| - \dot{\alpha}_N + \Delta_N \tag{3.66}$$

$$\dot{z}_4 = -\frac{T}{m}\left(s\psi s\theta c\phi - c\psi s\phi\right) - \frac{k_s}{m}\dot{E}\left|\dot{E}\right| - \dot{\alpha}_E + \Delta_E \tag{3.67}$$

where,

$$\begin{aligned} \Delta_N &= \frac{k_s}{m}\left\{\dot{N}\left|\dot{N}\right| - \left(\dot{N} - w_N\right)\left|\dot{N} - w_N\right|\right\} \\ \Delta_E &= \frac{k_s}{m}\left\{\dot{E}\left|\dot{E}\right| - \left(\dot{E} - w_E\right)\left|\dot{E} - w_E\right|\right\} \\ \dot{\alpha}_N &= \ddot{N}_d - k_1(\dot{N} - \dot{N}_d) \\ \dot{\alpha}_E &= \ddot{E}_d - k_3(\dot{E} - \dot{E}_d) \end{aligned} \tag{3.68}$$

As in the altitude controller design, we separated the structured uncertainty due to drag from the external disturbance due to wind. This provides better estimation capabilities for k_s/m, while still allowing for the robust controller to overcome bounded external disturbances. Given that ϕ and θ are coupled, we can stabilize (3.66) and (3.67) as follows. Let us define,

$$\begin{aligned} c\psi s\theta c\phi + s\psi s\phi &= u_2 \\ s\psi s\theta c\phi - c\psi s\phi &= u_4 \end{aligned} \tag{3.69}$$

where the intermediate control inputs,

$$u_2 = -\frac{1}{\hat{c}_1 T}\left(\hat{c}_2 \dot{N}\left|\dot{N}\right| + \dot{\alpha}_N + u_{r2} - k_2 z_2\right) \tag{3.70}$$

$$u_4 = -\frac{1}{\hat{c}_1 T}\left(\hat{c}_2 \dot{E}\left|\dot{E}\right| + \dot{\alpha}_E + u_{r4} - k_4 z_4\right) \tag{3.71}$$

and $\hat{c}_1 = 1/m$, $\hat{c}_2 = k_s/m$, and $k_2, k_4 > 0$. Then from (3.66) and (3.67), the closed-loop systems are given by,

$$\dot{z}_2 + k_2 z_2 = u_{r2} + \left\{ \boldsymbol{\varphi}_2^T \tilde{\boldsymbol{\theta}} + \Delta_N \right\} \tag{3.72}$$

$$\dot{z}_4 + k_4 z_4 = u_{r4} + \left\{ \boldsymbol{\varphi}_4^T \tilde{\boldsymbol{\theta}} + \Delta_E \right\} \tag{3.73}$$

respectively, where the vectors $\boldsymbol{\varphi}_2^T = [u_2,\ \dot{N}|\dot{N}|]$, $\boldsymbol{\varphi}_2^T = [u_4,\ \dot{E}|\dot{E}|]$, and $\tilde{\boldsymbol{\theta}}_2 = \tilde{\boldsymbol{\theta}}_4 = [\tilde{c}_1,\ \tilde{c}_2]^T$. The left hand sides of (3.72) and (3.73) represent the closed-loop stable dynamics of the nominal systems. The right hand sides contain all model uncertainties and disturbances, for which we know the bounds from Assumption 3.2.1. Therefore, we can design the robust feedback terms u_{r2} and u_{r4} to stabilize the error dynamics as we have done in (3.50). To improve the steady state tracking performance, we use the projection-based parameter adaptation law,

$$\dot{\hat{\boldsymbol{\theta}}}_i = \text{Proj}_{\hat{\theta}_i}(\boldsymbol{\Gamma} \boldsymbol{\varphi}_i z_i) \tag{3.74}$$

for $i = 2, 4$. Having designed the intermediate control inputs u_2 and u_4 and their respective adaptation laws, we can return to the coupled equations in (3.69). To solve for the roll and pitch angles commands, we write (3.69) in the following form.

$$\begin{bmatrix} s\psi & c\psi c\phi \\ -c\psi & s\psi c\phi \end{bmatrix} \begin{bmatrix} s\phi \\ s\theta \end{bmatrix} = \begin{bmatrix} u_2 \\ u_4 \end{bmatrix} \tag{3.75}$$

If we restrict the angles $|\phi(t)| \leq \beta$ and $|\theta(t)| \leq \beta$ for all t and for some angle $\beta < \pi/2$, the previous system can be inverted to yield,

$$\begin{bmatrix} s\phi \\ s\theta \end{bmatrix} = \begin{bmatrix} s\psi & -c\psi \\ c\psi/c\phi & s\psi/c\phi \end{bmatrix} \begin{bmatrix} u_2 \\ u_4 \end{bmatrix} \tag{3.76}$$

Then the desired roll and pitch angles that are required for tracking $N_d(t)$ and $E_d(t)$ are given by,

$$\phi_d = \text{sat}\left\{ \arcsin\left(s\psi u_2 - c\psi u_4 \right) \right\} \tag{3.77}$$

$$\theta_d = \text{sat}\left\{ \arcsin\left(\frac{c\psi}{c\phi} u_2 - \frac{s\psi}{c\phi} u_4 \right) \right\} \tag{3.78}$$

where the saturation function, sat(•), is defined as follows,

$$\text{sat}(u) = \begin{cases} \beta\ , & \text{if } u > \beta \\ u\ , & \text{if } |u| \leq \beta \\ -\beta\ , & \text{if } u < -\beta \end{cases} \tag{3.79}$$

with $\beta = \pi/4$ radians. Note that the position control is achieved by pitch and roll angles, irrespective of the heading angle ψ. For the quadrotor system, the heading angle is a degree of freedom that can be set to an arbitrary value. Nevertheless, in situations where the quadrotor is taking measurements as it flies, e.g. with a multi-spectral or infrared sensor, the heading is usually desired to be along the flight path.

We propose the guidance logic in Algorithm 1 for computing the desired heading angle, ψ_d. In this algorithm, atan2(•) is the arctangent function that returns the angle in its appropriate quadrant, ψ_i is the heading at the previous waypoint, and ψ_{wp}, ΔN, and ΔE are defined in Figure 3.5. The time variable t is re-initialized upon reaching each new waypoint, and a is a design parameter.

Algorithm 1 Computing the desired heading angle ψ_d

```
$\psi_{wp} \leftarrow \pi/2 - \text{atan2}(\Delta N, \Delta E)$
$\Delta\psi \leftarrow \psi_{wp} - \psi_i$
if $|\Delta\psi| \leq \pi$ then
    $\psi_d \leftarrow \psi_{wp} - \Delta\psi e^{-at}$
else
    $\psi_d = \psi_{wp} - \text{sgn}(\Delta\psi)2\pi - (\Delta\psi - \text{sgn}(\Delta\psi)2\pi)e^{-at}$
end if
```

The guidance logic in Algorithm 1 ensures that the quadrotor turns in the shortest direction of rotation, either clockwise or counterclockwise, at a rate that does not exceed the physical abilities of the quadrotor, i.e. the turn rate is constrained by $\dot{\psi}_{max} = (\pi a)$ rad/s. This completes the position controller in the outer-loop of the

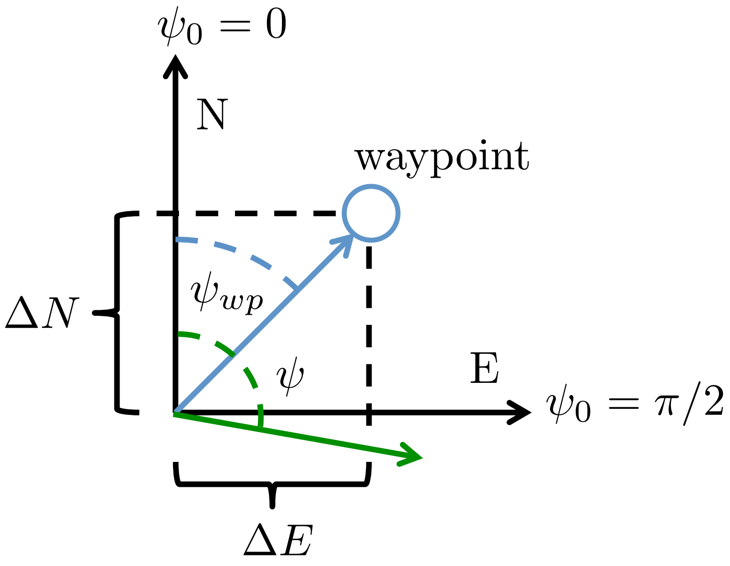


Figure 3.5.. Definition of heading and waypoint angles in the inertial frame

overall controller architecture. In the following section, we discuss the trajectory generation method that improves the transient performance of the quadrotor and allows for autonomous waypoint navigation.

3.2.4 Trajectory Generation

Generating a trajectory from a set of waypoints enables the quadrotor to navigate autonomously and provides a framework for additional capabilities, such as mid-flight trajectory re-planning and collision avoidance. In [4], it was shown that a better performance is attainable when using path-following for nonlinear systems rather than reference tracking. The authors refer to path-following as tracking a geometric path while satisfying dynamic specifications, such as a desired velocity. In accordance with this, the authors in [59] examine the benefits of generating a desired trajectory $x_d(t)$ from a reference trajectory $x_r(t)$ by using an n-th order filter,

$$x_d^{(n)} + \beta_1 x_d^{(n-1)} + \ldots + \beta_n x_d \;=\; x_r^{(n)} + \beta_1 x_r^{(n-1)} + \ldots + \beta_n x_r \tag{3.80}$$

and assigning the initial conditions of (3.80) such that the initial errors $z_i(0) = 0$ for $i = 1, \ldots, n$. The parameters, β_i, in this problem need to be chosen properly in order to have a stable filter with good performance. In [9], the trajectory generation algorithm uses the concept of differential flatness to compute the desired trajectory and then optimizes the results based on time or flight envelope constraints. Inspired by these previous works, we propose the following algorithm for trajectory generation, which incorporate the physical capabilities of the specific quadrotor. Let the desired state $x_d(t)$ denote the generated trajectory for the N, E, or h states, and consider the function

$$x_d(t) = x_f - (x_f - x_i)(1 + at)e^{-at} + \dot{x}_i t e^{-at} \tag{3.81}$$

where x_f is the final desired position, $x_i = x(0)$ and $\dot{x}_i = \dot{x}(0)$ are the initial position and velocity, and $a > 0$ is a design parameter. The time variable t is re-initialized upon reaching each successive waypoint. The first and second derivatives of (3.81) are given by

$$\dot{x}_d(t) = e^{-at}\left[\dot{x}_i + at((x_f - x_i)a - \dot{x}_i)\right] \tag{3.82}$$

$$\ddot{x}_d(t) = -ae^{-at}\left[(x_f - x_i)a(at - 1) + \dot{x}_i(2 - at)\right] \tag{3.83}$$

It is simple to verify that $x_d(0) = x_i$ and $\dot{x}(0) = \dot{x}_i$, and therefore, the initial tracking errors $z_1(0) = z_2(0) = 0$. This improves the transient performance for trajectory tracking, as it guarantees that the Lyapunov function $V(0) = 0$ and therefore the error bound is smaller. Two example trajectories and their derivatives are shown for $x_i = \dot{x}_i = \ddot{x}_i = 0$, and $x_f = 5$ meters in Figure 3.6. In both cases, the generated trajectory satisfies $z_1(0) = z_2(0) = 0$. For the design parameter $a = 0.5$, the initial error is $\dot{z}_2(0) = 1.25$ m/s^2, and for $a = 1.0$, $\dot{z}_2(0) = 5$ m/s^2. Therefore, a smaller a

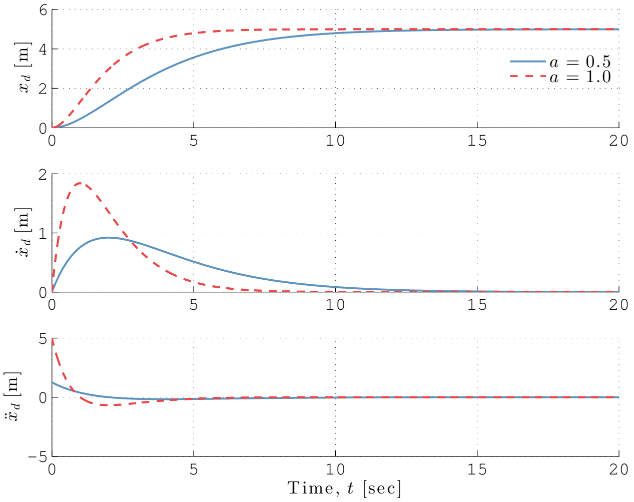


Figure 3.6.. Trajectory generated for $x_i = \dot{x}_i = \ddot{x}_i = 0$, and $x_f = 5$

can guarantee better transient trajectory tracking as the initial acceleration error is smaller, but it comes at the cost of a slower rise time. Although a can be chosen arbitrarily, we propose two techniques to further improve the transient performance. Observe that the initial value of the desired acceleration is given by,

$$\ddot{x}_d(0) = (x_f - x_i)a^2 - 2\dot{x}_i a \tag{3.84}$$

Ideally, we would like $\ddot{x}_d(0) = \ddot{x}_i$. However, this cannot always be guaranteed. One method to overcome this difficulty is to minimize $\delta = (\ddot{x}_d(0) - \ddot{x}_i)$ by solving the following optimization problem,

$$\min_{a > \epsilon} \left| (x_f - x_i)a^2 - 2\dot{x}_i a - \ddot{x}_i \right| \tag{3.85}$$

where $\epsilon > 0$ is an acceptable minimum value for a. Note that this optimization problem only needs to be solved once at each new waypoint. Another approach is to predetermine a constant a, and ensure that the chosen waypoints are relatively close to each other, such that $(x_f - x_i)$ is small. This can be done, for example, by dividing a long journey into smaller intermediate trajectories with adjoining waypoints. In the following section, we demonstrate the controller performance with two illustrative simulations.

3.3 Simulation Setup and Results

In this section, we present two illustrative examples to demonstrate the performance of the ARC-LMI controller. The first example is a mid-flight package delivery mission in the presence of wind disturbances (Figure 3.7) and the second example is a building inspection mission in the presence of strong wind gusts and turbulence (Figure 3.8).

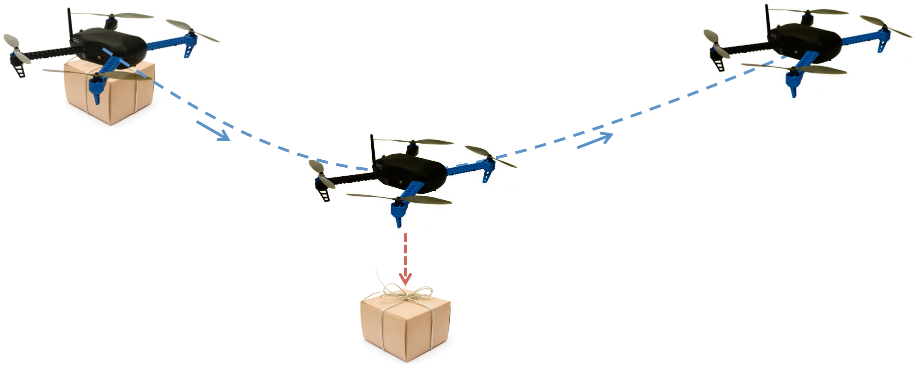


Figure 3.7.. Package delivery mission schematic

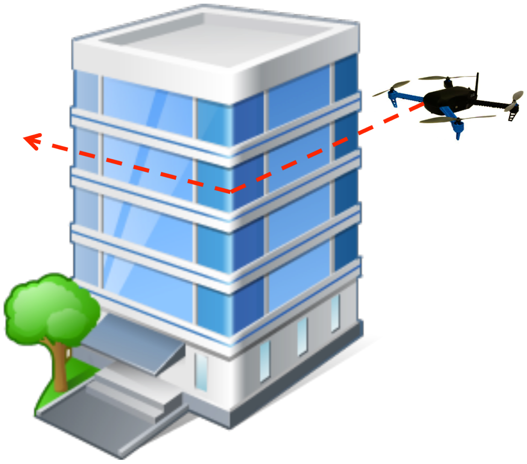


Figure 3.8.. Building inspection mission schematic

Matlab's Simulink software is used to model the full nonlinear dynamics of the quadrotor (with S-functions) and to model the wind conditions (with the Dryden continuous turbulence model and the Dryden discrete wind gust model [30], [35]). The ARC-LMI controllers and trajectory generator are also constructed with S-functions. Further details regarding the software architecture are given in Appendix C. The quadrotor simulation parameters are based on [1] and [47], and are shown in Table 3.1.

Table 3.1.. Quadrotor simulation parameters

Parameter	**Description**	**Value**	**Unit**
m	quadrotor mass	1.282	kg
m_p	package mass	0.500	kg
g	gravitational acceleration	9.810	m/s^2
I_{xx}, I_{yy}	moments of inertia	4.856×10^{-3}	kg·m^2
I_{zz}	moment of inertia	8.801×10^{-3}	kg·m^2
I_{xz}	moment of inertia	2.428×10^{-4}	kg·m^2
I_r	rotor moment of inertia	8.801×10^{-5}	kg·m^2
k_s, k_u	drag coefficients	0.250	kg/m
k_t	motor thrust factor	1.733×10^{-5}	kg·m
k_m	motor gain	0.200×10^{2}	1/s
b	rotor drag factor	1.140×10^{-7}	kg·m^2
$l_{\phi,f}$	length	0.222	m
$l_{\phi,b}$	length	0.206	m
$l_{\theta,f}$	length	0.130	m
$l_{\theta,b}$	length	0.136	m

3.3.1 Example 1: Package Delivery

In the first example, we simulate a package delivery scenario in the presence of constant wind. The quadrotor is required to descend from a height of 10 meters to 5 meters, release a package with an unknown mass of $m_p = 0.5$ kilograms, and then climb back to a height of 10 meters. The flight is autonomous and based on a set of waypoints (shown as green spheres in Figure 3.9). The external disturbances include a constant north-east wind velocity of 1 m/s. Once the package is dropped, the total mass of the system ($m_{tot} = m + m_p$) abruptly changes. In reality, the package influences the overall system's drag coefficients and moments of inertia, too. To take these effects into account, the parameters that vary after dropping the package are shown in Table 3.2.

Table 3.2.. Change in simulation parameters after package drop

Parameter	**Description**	**Value**	**Unit**
m_p	package mass	0.00	kg
I_{xx}, I_{yy}	moments of inertia	$0.95 \times I_{xx}$	kg·m^2
I_{zz}	moment of inertia	$0.90 \times I_{zz}$	kg·m^2
I_{xz}	moment of inertia	$0.90 \times I_{xz}$	kg·m^2
k_s, k_u	drag coefficients	$0.90 \times k_s$	kg/m

A time history of the quadrotor height is shown in Figure 3.10 for clarity, and the tracking errors are shown in Figure 3.11. The initial parameter estimates $\hat{\boldsymbol{\theta}}(t_0)$ are equal to the parameters with no mass attached, i.e. the parameters in Table 3.1. It is evident that the errors remain bounded even in the presence of wind disturbances (shown in Figure 3.13) and after the abrupt change of several parameters. The norm of the three-dimensional tracking error vector $||\xi_e||$ does not exceed 0.1 meters.

In this scenario, the altitude dynamics have a relatively 'exciting' trajectory. Therefore, we can expect the parameter estimates to be closer to their true val-

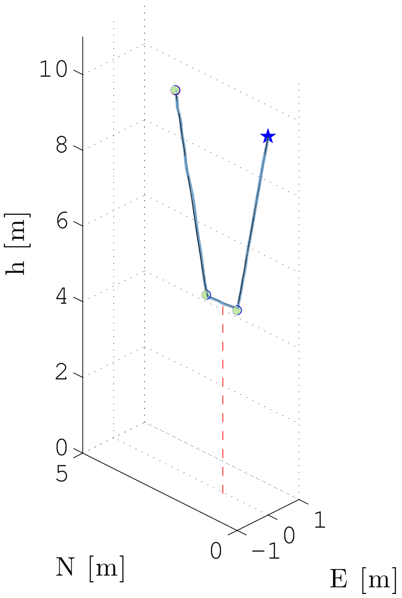


Figure 3.9.. Package delivery: three-dimensional trajectory tracking. The green shperes are waypoints, the red dashed line shows the package drop location, and the star indicates the initial position.

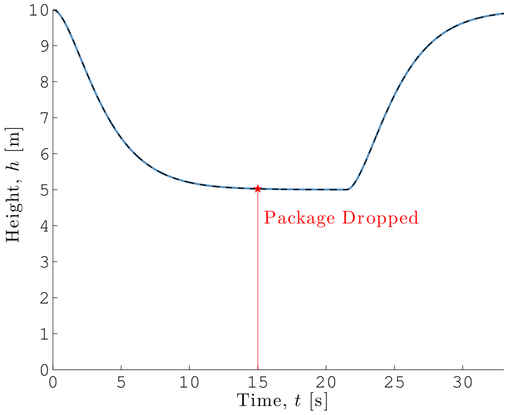


Figure 3.10.. Package delivery: altitude history (solid blue line) reference height (dashed line).

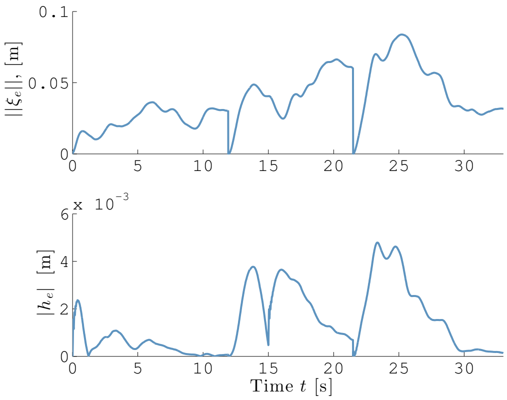


Figure 3.11.. Package delivery: norm of tracking error, $||\xi_e|| = ||\xi_d - \xi||$ and absolute altitude error, $|h_e| = |h - h_d|$.

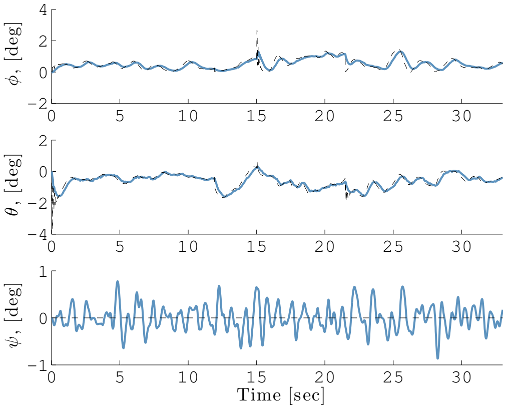


Figure 3.12.. Package delivery: roll, pitch, and yaw histories (solid lines), and reference angles (dashed lines).

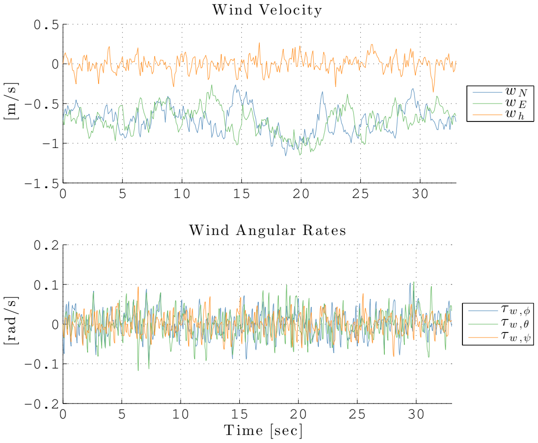


Figure 3.13.. Package delivery: wind disturbances: 1 m/s NW wind.

ues. Figure 3.15 presents the time history of the total mass estimate $\hat{m}_{tot}$ from the altitude controller's adaptation law, together with the true value m_{tot}. Evidently, the mass estimate quickly approaches the correct value, which helps to reduce the steady state tracking error. Furthermore, the motor speeds in Figure 3.14 indicate that there was no actuator saturation and the control inputs responded quickly to the sudden package drop.

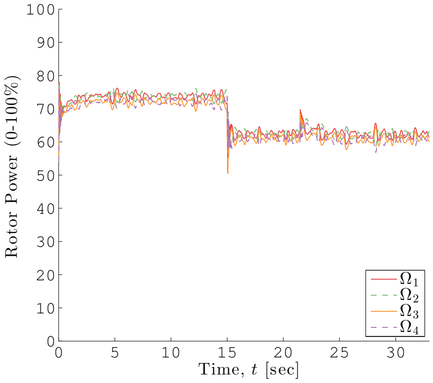


Figure 3.14.. Package delivery: motor angular velocities, normalized.

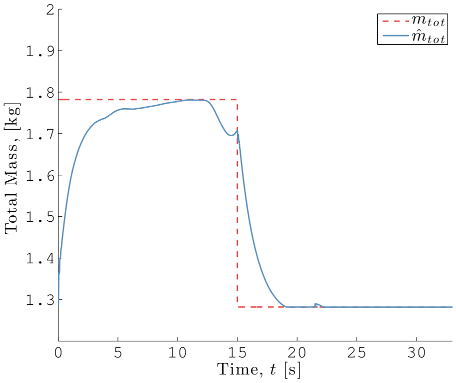


Figure 3.15.. Package delivery: total mass estimate (solid blue line) and true total mass (dashed red line).

To further demonstrate the advantages of the ARC-LMI controller, we simulate a sinusoidal altitude reference tracking, in the presence of a 7 m/s wind gust (equivalent to a downward force of 0.75 times the weight of the quadrotor), *and* a 0.5 kg package dropped at $t = 10$ sec (see Figure 3.16). The absolute altitude tracking error is compared in Figure 3.17 with two other control methods: a project-based direct adaptive controller (i.e. the ARC-LMI controller without the robust feedback), and a deterministic robust controller (i.e. the ARC-LMI controller with the adaptation switched off). The initial parameter estimates are those of the nominal system without a package.

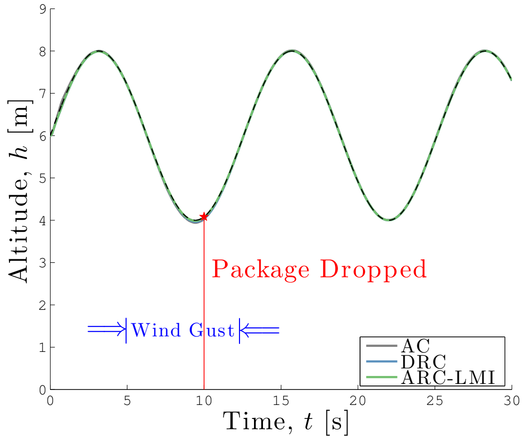


Figure 3.16.. Controller comparison: altitude tracking and package drop (at $t = 10$ sec)in the presence of a 1 m/s NW wind and a 7 m/s wind gust from 5-12 seconds.

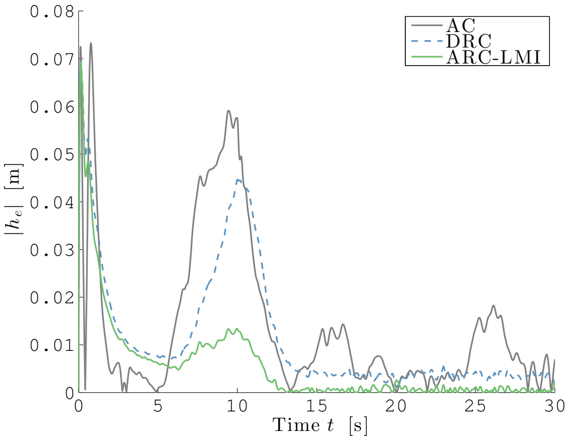


Figure 3.17.. Controller comparison: absolute altitude tracking error, $|h_e| = |h - h_d|$.

From Figure 3.17, it is evident that the ARC-LMI controller has at most the same absolute altitude tracking error as the other controllers. Aided by the online learning scheme, the ARC-LMI controller has significantly smaller errors during the strong wind gust disturbance (between 5-12 seconds). Furthermore, the steady state tracking error is significantly improved from 15 seconds onward, i.e. after the package is dropped and the wind gust ends. Therefore, it should be evident that the ARC-LMI control algorithm successfully merges the adaptive and robust approaches without sacrificing the performance of either one.

3.3.2 Example 2: Building Inspection in Wind Gusts

In this second example, we simulate a building inspection mission in an urban environment. The quadrotor is carrying an uncertain mass ($m_p = 0.25$ kg) that represents an added camera or other visual sensor. The mission requires a close flight around a right-angle corner of a building. However, the wind in the simulation is turbulent and a sudden downward gust begins at $t = 5$ seconds. This type of airflow is common in urban environments and in close proximity to tall buildings [50].

This mission motivates the importance of precise trajectory tracking, because a large deviation could signify a collision between the quadrotor and the building. It should be made clear that this scenario is a particular case that belongs to a larger class of problems, for which the ARC-LMI controller is well suited, as shown in the following figures. Figure 3.18 displays the three-dimensional trajectory and waypoints of the quadrotor, and Figure 3.19 presents the tracking errors. Considering the strong downward wind gust and turbulent air flow (shown in Figure 3.21), the trajectory tracking is highly precise, and the three-dimensional error vector is norm-bounded by at most 0.2 meters. The attitude angles, which receive the reference commands from the outer-loop position controller, are presented in Figure 3.20.

The drag force from the wind gust was equivalent to 0.75 times the weight of the quadrotor, which is a considerable disturbance. It should be noted from Figure 3.22 that the motors were not saturated, but were close to their maximum thrust capability. Nevertheless, the trajectory tracking performance was not compromised. Furthermore, as the the trajectory in this example was constant, we cannot expect the parameter estimates to converge to their true values, but we still guarantee that the robust feedback will overcome these uncertainties in addition to external disturbances.

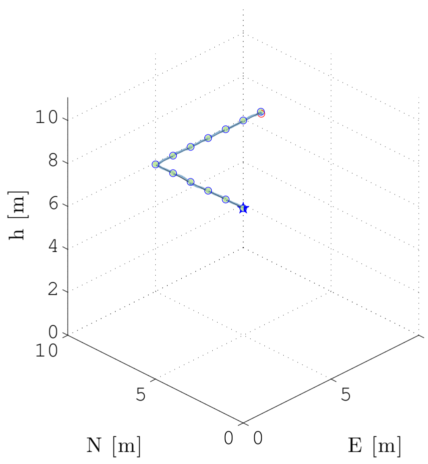


Figure 3.18.. Building inspection: three-dimensional trajectory tracking. The green shperes are waypoints, the red dashed line shows the package drop location, and the star indicates the initial position.

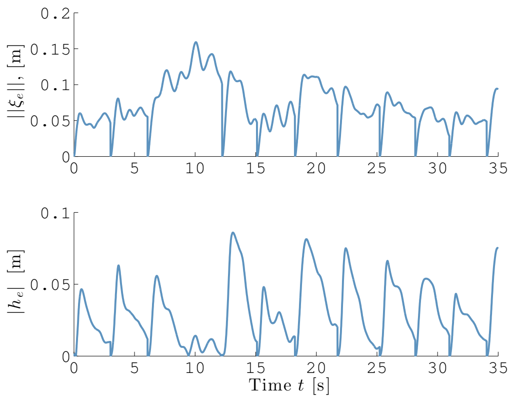


Figure 3.19.. Building inspection: norm of position error, $||\xi_e|| = ||\xi_d - \xi||$ and absolute altitude error, $|h_e| = |h(t) - h_d(t)|$.

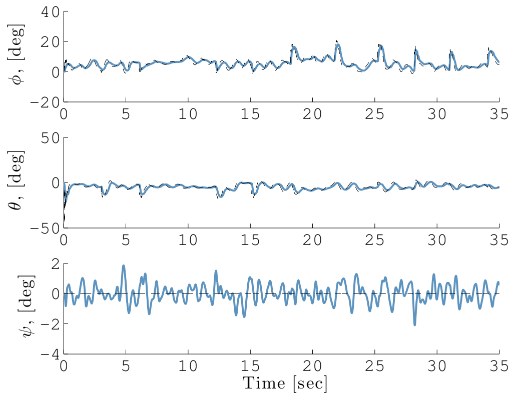


Figure 3.20.. Building inspection: roll, pitch, and yaw histories (solid blue lines), and reference angles (dashed lines).

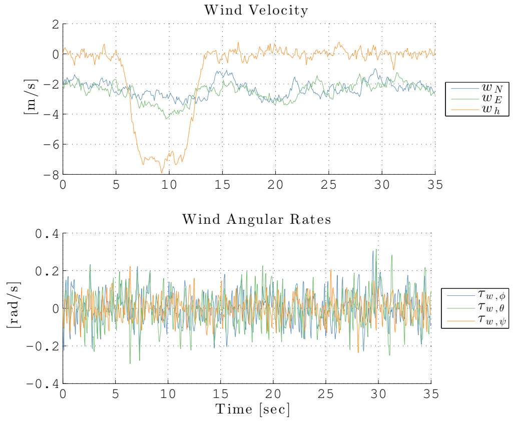


Figure 3.21.. Building inspection: wind disturbances: 7 m/s down gust from 6-12 sec (drag force ≈ 0.75 mg).

This is justified by the incorrect total mass estimate shown in Figure 3.23. The mass estimator is, in fact, excited by the downward wind, which cannot be differentiated from additional weight.

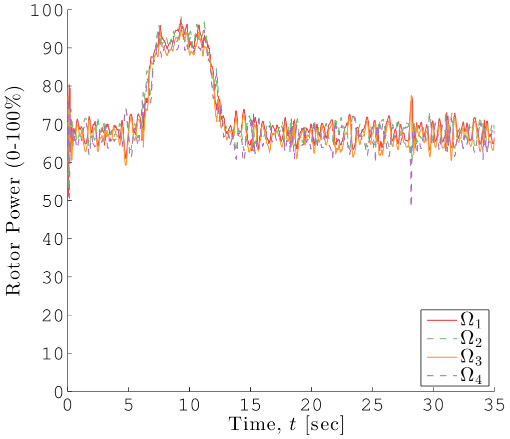


Figure 3.22.. Building inspection: motor velocities, normalized.

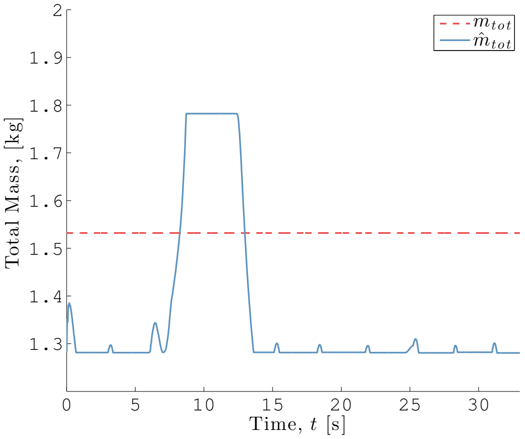


Figure 3.23.. Building inspection: total mass estimate (solid blue line) and true total mass (dashed red line).

In summary, the ARC-LMI controller was designed in Section 3.2 with the objective of precise trajectory tracking in the presence of parameteric uncertainties and disturbances. In addition to external disturbances, the ARC-LMI controller was shown to be robust to unmodeled dynamics (e.g., we have neglected the motor dynamics in the controller design). We have presented two illustrative examples, which are based on recent quadrotor applications, to demonstrate the performance of the controller. In both cases, the ARC-LMI controller overcame the parameteric uncertainties, the external disturbances, and the unmodeled dynamics, and attained a high level of precision in transient and steady state tracking.

4. CONCLUSION

This thesis has dealt primarily with the nonlinear controller design for dynamic systems in the presence of parametric uncertainties and matched disturbances. The theoretical framework was presented and applied to an autonomous quadrotor simulation.

Chapter 2 has presented a novel control approach, which fuses the synergistic qualities of adaptive robust control (ARC) and the powerful design tools of linear matrix inequalities (LMI) to control uncertain second-order nonlinear systems. The ARC-LMI control input is continuous and bounded, and guarantees fast and robust trajectory tracking in the presence of bounded disturbances. The LMI-based robust stabilizing feedback term is optimized to reduce the control effort. In general, transient performance can be prescribed by tuning two design parameters in a known form. In the absence of external disturbances, asymptotic tracking is attained, and, if the trajectory is sufficiently exciting, the parameters converge to their true values. We have illustrated the ARC-LMI controller on a single-link manipulator system and demonstrated that the tracking error remains within the prescribed bounds even in the presence of external disturbances. As expected, the tracking error asymptotically converged to zero when the disturbance was removed.

Chapter 3 has presented a new nonlinear control approach applied to the attitude and trajectory tracking of a quadrotor helicopter with an X-configuration. A detailed dynamic model of the quadrotor helicopter was derived based on Newtonian mechanics prior to the controller design process. The ARC-LMI controller guarantees a prescribed transient performance in the presence of uncertain parameter uncertainties (e.g. moments of inertia and quadrotor mass) and bounded external disturbances (e.g. constant wind disturbance and wind gusts). These properties made the ARC-LMI controller well suited for problems such as precise trajectory tracking in windy conditions and package delivery scenarios. We have demonstrated the proposed ARC-LMI controller with two illustrative examples. As expected, the ARC-LMI controller was able to maintain the bounded tracking errors and, after dropping the package of uncertain mass (which caused an abrupt change in the system's mass and inertia parameters) the online adaptation law reduced the steady state tracking errors. Moreover, by using the proposed trajectory generation scheme, a precise trajectory tracking performance was achieved, while minimizing the control effort and avoiding actuator saturation.

4.1 Future Work

In future works, we will extend the ARC-LMI theory to multiple-input multiple-output systems for more general systems in a semi-strict feedback form. The ARC-LMI algorithm can also be extended to account for norm-bound linear differential inclusions, such that the LMI-based feedback term can stabilize a larger class of uncertain nonlinearities. Additionally, by considering an integrated direct/indirect adaptive robust controller with linear matrix inequalities, the parameter estimation can be improved without deteriorating the robust transient performance. This type of controller can be used with fault-detection schemes to alert the control engineer of actuator malfunction.

Furthermore, we will test the performance of the ARC-LMI controller on the IRIS quadrotor, both in autonomous flight in windy conditions and in package dropping experiments. By extending the ARC-LMI controller approach to multi-input multi-output systems, the attitude controller can be designed simultaneously for all three torque-input channels, which could produce an even better performance than the modular approach in this thesis (at the expense of a more sophisticated controller design).

REFERENCES

REFERENCES

[1] 3D Robotics. *IRIS Operation Manual V5*, March 2014.

[2] Adafruit. 3D Robotics Iris drone featured on blog.adafruit.com. `http://bit.ly/1vP7vYh`. Accessed: 2015-02-20.

[3] P. Adigbli, C. Grand, J.-B. Mouret, and S. Doncieux. Nonlinear attitude and position control of a micro quadrotor using sliding mode and backstepping techniques, 2007.

[4] A. P. Aguiar, J. P. Hespanha, and P. V. Kokotović. Performance limitations in reference tracking and path following for nonlinear systems. *Automatica*, 44(3):598–610, 2008.

[5] K. Alexis, G. Nikolakopoulos, and A. Tzes. Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances. *Control Engineering Practice*, 19(10):1195–1207, 2011.

[6] H. Bolandi, M. Rezaei, R. Mohsenipour, H. Nemati, and S. M. Smailzadeh. Attitude control of a quadrotor with optimized pid controller. *Intelligent Control and Automation*, 4:335, 2013.

[7] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory*, volume 15. SIAM, 1994.

[8] L. R. G. Carrillo, A. E. D. López, R. Lozano, and C. Pégard. Modeling the quad-rotor mini-rotorcraft. In *Quad Rotorcraft Control*, pages 23–34. Springer, 2013.

[9] A. Chamseddine, Y. Zhang, C. A. Rabbath, and D. Theilliol. Trajectory planning and replanning strategies applied to a quadrotor unmanned aerial vehicle. *Journal of Guidance, Control, and Dynamics*, 35(5):1667–1671, 2012.

[10] H. Choi and K.-S. Ro. LMI-based sliding-mode observer design method. *IEE Proceedings-Control Theory and Applications*, 152(1):113–115, 2005.

[11] M. Corless. Course Notes: *AAE 666* (2013). Purdue University, West Lafayette, IN, 47907.

[12] M. Corless and G. Leitmann. Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems. *IEEE Transactions on Automatic Control*, 26(5):1139, 1981.

[13] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky. Adaptive control of quadrotor uavs: A design trade study with flight evaluations. *Control Systems Technology, IEEE Transactions on*, 21(4):1400–1406, 2013.

[14] R. Freeman, M. Krstić, and P. Kokotović. Robustness of adaptive nonlinear control to bounded uncertainties. *Automatica*, 34(10):1227–1230, 1998.

[15] P. C. Garcia, R. Lozano, and A. E. Dzul. *Modelling and control of mini-flying machines*. Springer Science & Business Media, 2006.

[16] N. K. Gupta, R. Goel, and N. Ananthkrishnan. Design/development of mini/micro air vehicles through modelling and simulation: Case of an autonomous quadrotor. *Defence Science Journal*, 61(4):337–345, 2011.

[17] S. Gupte, P. I. T. Mohandas, and J. M. Conrad. A survey of quadrotor unmanned aerial vehicles. In *Southeastcon, 2012 Proceedings of IEEE*, pages 1–6. IEEE, 2012.

[18] A. Hern. DHL launches first commercial drone 'parcelcopter' delivery service. *The Guardian*, September 2014. `http://www.theguardian.com/technology/2014/sep/25/german-dhl-launches-first-commercial-drone-delivery-service`.

[19] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. In *Proc. of the AIAA Guidance, Navigation, and Control Conference*, volume 2, 2007.

[20] F. Hong, S. Ge, B. Ren, and T. Lee. Robust adaptive control for a class of uncertain strict-feedback nonlinear systems. *International Journal of Robust and Nonlinear Control*, 19(7):746–767, 2009.

[21] P. Ioannou and J. Sun. *Robust adaptive control*. Dover Publications, 2012.

[22] B. Jansen. FAA approves first commercial drone over land. *USA Today*, June 2014. `http://www.usatoday.com/story/money/business/2014/06/10/faa-drones-bp-oil-pipeline-aerovironment-north-shore/10264197/`.

[23] F. Kendoul, D. Lara, I. Fantoni, and R. Lozano. Real-time nonlinear embedded control for an autonomous quadrotor helicopter. *Journal of guidance, control, and dynamics*, 30(4):1049–1061, 2007.

[24] H. K. Khalil. *Nonlinear systems, 3rd ed.*, volume 9. New Jewsey, Prentice Hall, 2002.

[25] B.-Y. Lee, D.-W. Yoo, and M.-J. Tahk. Performance comparison of three different types of attitude control systems of the quad-rotor uav to perform flip maneuver. *International Journal of Aeronautical and Space Sciences*, 14(1):58–66, 2013.

[26] H. Lim, J. Park, D. Lee, and H. J. Kim. Build your own quadrotor: Open-source projects on unmanned aerial vehicles. *Robotics & Automation Magazine, IEEE*, 19(3):33–45, 2012.

[27] K. Y. Lim and M. Eslami. Robust adaptive controller designs for robot manipulator systems. *Robotics and Automation, IEEE Journal of*, 3(1):54–66, 1987.

[28] V. Martinez. Modelling of the flight dynamics of a quadrotor helicopter. Master's thesis, Cranfield University, 2007.

[29] I. Masubuchi, A. Ohara, and N. Suda. LMI-based controller synthesis: A unified formulation and solution. *International Journal of Robust and Nonlinear Control*, 8(8):669–686, 1998.

[30] R. E. McFarland. A standard kinematic model for flight simulation at nasa ames. 1975.

[31] D. Mellinger, N. Michael, and V. Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *The International Journal of Robotics Research*, page 0278364911434236, 2012.

[32] B. Mirkin, P.-O. Gutman, and Y. Shtessel. Tracking for nonlinear plants with multiple unknown time-varying state delays using sliding mode with adaptation. *International Journal of Robust and Nonlinear Control*, 20(13):1455–1464, 2010.

[33] M. Mohammadi and A. M. Shahri. Adaptive nonlinear stabilization control for a quadrotor uav: theory, simulation and experimentation. *Journal of Intelligent & Robotic Systems*, 72(1):105–122, 2013.

[34] A. Nagaty, S. Saeedi, C. Thibault, M. Seto, and H. Li. Control and navigation framework for quadrotor helicopters. *Journal of Intelligent & Robotic Systems*, 70(1-4):1–12, 2013.

[35] M. Natick. *Aerospace Blockset UserâĂŹs Guide*. The MathWorks, Inc., 2007.

[36] C. Nicol, C. Macnab, and A. Ramirez-Serrano. Robust adaptive control of a quadrotor helicopter. *Mechatronics*, 21(6):927–938, 2011.

[37] T. Pancake, M. Corless, and M. Brockman. Analysis and control of polytopic uncertain/nonlinear systems in the presence of bounded disturbance inputs. In *American Control Conference, 2000. Proceedings of the 2000*, volume 1, pages 159–163. IEEE, 2000.

[38] C.-W. Park. LMI-based robust stability analysis for fuzzy feedback linearization regulators with its applications. *Information sciences*, 152:287–301, 2003.

[39] L. Pollini and A. Metrangolo. Simulation and robust backstepping control of a quadrotor aircraft. In *AIAA Modeling and Simulation Technologies Conference*, 2008.

[40] X. Qi, D. Theilliol, J. Qi, Y. Zhang, J. Han, D. Song, L. Wang, and Y. Xia. Fault diagnosis and fault tolerant control methods for manned and unmanned helicopters: a literature review. In *Control and Fault-Tolerant Systems (SysTol), 2013 Conference on*, pages 132–139. IEEE, 2013.

[41] J. Reed and P. Ioannou. Instability analysis and robust adaptive control of robotic manipulators. *IEEE transactions on robotics and automation*, 5(3), 1989.

[42] E. Reyes-Valeria, R. Enriquez-Caldera, S. Camacho-Lara, and J. Guichard. Lqr control for a quadrotor using unit quaternions: Modeling and simulation. In *Electronics, Communications and Computing (CONIELECOMP), 2013 International Conference on*, pages 172–178. IEEE, 2013.

[43] M. Rich. Model development, system identification, and control of a quadrotor helicopter. Master's thesis, Iowa State University, 2012.

[44] K. Runcharoon and V. Srichatrapimuk. Sliding mode control of quadrotor. In *Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), 2013 International Conference on*, pages 552–557. IEEE, 2013.

[45] D. Shamah. UAV 'eye in the sky' protects Jerusalem light rail. *The Times of Israel*, July 2014. `http://www.timesofisrael.com/uav-eye-in-the-sky-protects-jerusalem-light-rail/`.

[46] J. Slotine and W. Li. *Applied nonlinear control*, volume 60. Prentice-Hall Englewood Cliffs, NJ, 1991.

[47] A. Tayebi and S. McGilvray. Attitude stabilization of a four-rotor aerial robot. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 2, pages 1216–1221. IEEE, 2004.

[48] T. Van Den Boom. Robust nonlinear predictive control using feedback linearization and linear matrix inequalities. In *American Control Conference, 1997. Proceedings of the 1997*, volume 5, pages 3068–3072. IEEE, 1997.

[49] Q. Wang, Y. Hou, and C. Dong. Model reference robust adaptive control for a class of uncertain switched linear systems. *International Journal of Robust and Nonlinear Control*, 22(9):1019–1035, 2012.

[50] S. L. Waslander and C. Wang. Wind disturbance estimation and rejection for quadrotor position control. In *AIAA Infotech Aerospace Conference and AIAA Unmanned... Unlimited Conference, Seattle, WA*, 2009.

[51] Y. Xia, Z. Zhu, C. Li, H. Yang, and Q. Zhu. Robust adaptive sliding mode control for uncertain discrete-time systems with time delay. *Journal of the Franklin Institute*, 347(1):339–357, 2010.

[52] L. Xu and B. Yao. Adaptive robust precision motion control of linear motors with negligible electrical dynamics: theory and experiments. *Mechatronics, IEEE/ASME Transactions on*, 6(4):444–452, 2001.

[53] B. Yao. High performance adaptive robust control of nonlinear systems: a general framework and new schemes. In *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, volume 3, pages 2489–2494. IEEE, 1997.

[54] B. Yao. Integrated direct/indirect adaptive robust control of SISO nonlinear systems in semi-strict feedback form. In *American Control Conference, 2003. Proceedings of the 2003*, volume 4, pages 3020–3025. IEEE, 2003.

[55] B. Yao and M. Tomizuka. Smooth robust adaptive sliding mode control of manipulators with guaranteed transient performance. *Journal of dynamic systems, measurement, and control*, 118(4):764–775, 1996.

[56] B. Yao and M. Tomizuka. Adaptive robust control of SISO nonlinear systems in a semi-strict feedback form. *Automatica*, 33(5):893–900, 1997.

[57] B. Yao and M. Tomizuka. Adaptive robust control of MIMO nonlinear systems in semi-strict feedback forms. *Automatica*, 37(9):1305–1321, 2001.

[58] B. Yao and L. Xu. Observer-based adaptive robust control of a class of nonlinear systems with dynamic uncertainties. *International Journal of Robust and Nonlinear Control*, 11(4):335–356, 2001.

[59] B. Yao and L. Xu. Adaptive robust motion control of linear motors for precision manufacturing. *Mechatronics*, 12(4):595–616, 2002.

[60] F. Zhang. *The Schur complement and its applications*, volume 4. Springer Science & Business Media, 2006.

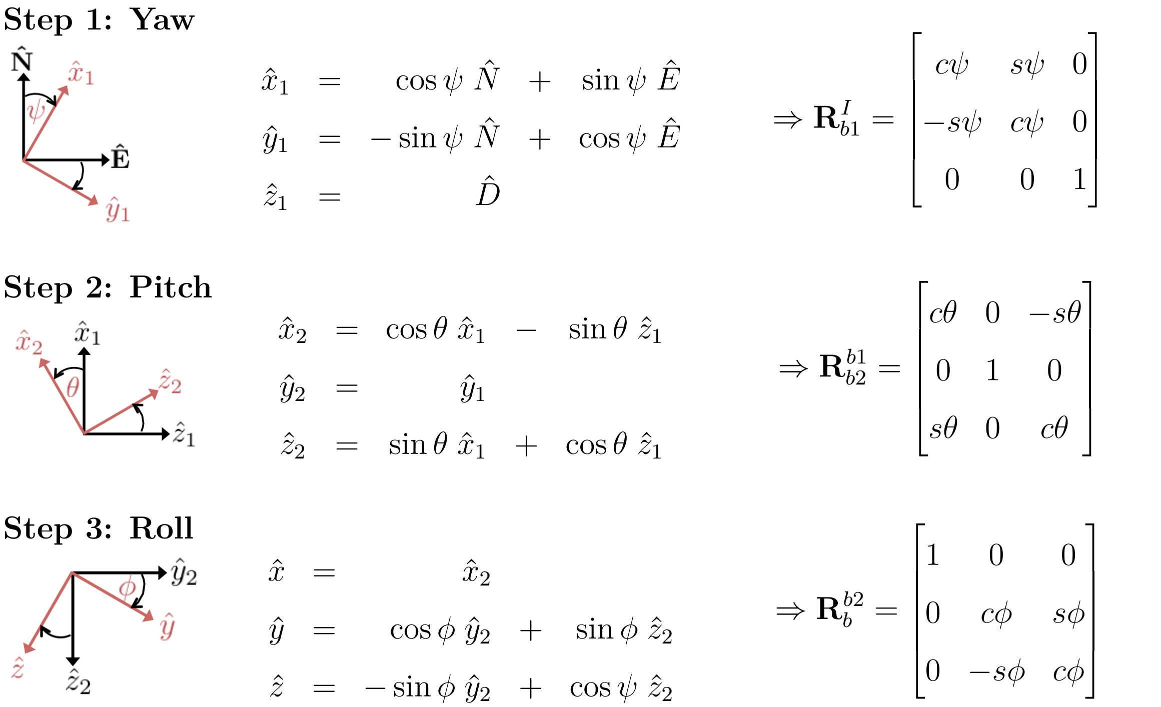
[61] F. Zheng, Q.-G. Wang, and T. H. Lee. Adaptive robust control of uncertain time delay systems. *Automatica*, 41(8):1375–1383, 2005.

[62] J. Zhong and B. Yao. Adaptive robust repetitive control of piezoelectric actuators. In *ASME 2005 International Mechanical Engineering Congress and Exposition*, pages 85–92. American Society of Mechanical Engineers, 2005.

APPENDICES

A. Derivation of the Rotation Matrix

We can derive the rotation matrix from the inertial frame (N, E, D) to the body frame (x, y, z) by successive rotations following the conventional order of yaw $(\psi) \rightarrow$ pitch $(\theta) \rightarrow$ roll (ϕ). The rotation matrix that transforms an a-frame to a b-frame is denoted $\mathbf{R}_b^a : \mathscr{F}^a \rightarrow \mathscr{F}^b$. For a more concise notation, the trigonometric functions are abbreviated as: $c\alpha \triangleq \cos(\alpha)$, $s\alpha \triangleq \sin(\alpha)$, $t\alpha \triangleq \tan(\alpha)$.



Step 1: Yaw

$$\begin{aligned} \hat{x}_1 &= \cos\psi\ \hat{N} + \sin\psi\ \hat{E} \\ \hat{y}_1 &= -\sin\psi\ \hat{N} + \cos\psi\ \hat{E} \\ \hat{z}_1 &= \hat{D} \end{aligned} \quad \Rightarrow \mathbf{R}_{b1}^{I} = \begin{bmatrix} c\psi & s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Step 2: Pitch

$$\begin{aligned} \hat{x}_2 &= \cos\theta\ \hat{x}_1 - \sin\theta\ \hat{z}_1 \\ \hat{y}_2 &= \hat{y}_1 \\ \hat{z}_2 &= \sin\theta\ \hat{x}_1 + \cos\theta\ \hat{z}_1 \end{aligned} \quad \Rightarrow \mathbf{R}_{b2}^{b1} = \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix}$$

Step 3: Roll

$$\begin{aligned} \hat{x} &= \hat{x}_2 \\ \hat{y} &= \cos\phi\ \hat{y}_2 + \sin\phi\ \hat{z}_2 \\ \hat{z} &= -\sin\phi\ \hat{y}_2 + \cos\psi\ \hat{z}_2 \end{aligned} \quad \Rightarrow \mathbf{R}_{b}^{b2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix}$$

Figure A.1.. Yaw → pitch → roll rotations and rotation matrices

Therefore, the rotation matrix from the inertial frame (N, E, D) to the body frame (x, y, z) is given by,

$$\mathbf{R}_b^I = \mathbf{R}_b^{b2}\,\mathbf{R}_{b2}^{b1}\,\mathbf{R}_{b1}^{I} = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{bmatrix}$$

Furthermore, it is clear that $\mathbf{R}_b^{b2}$, $\mathbf{R}_{b2}^{b1}$, and $\mathbf{R}_{b1}^{I}$ are orthogonal, and therefore, their product $\mathbf{R}_b^I = \mathbf{R}_b^{b2}\,\mathbf{R}_{b2}^{b1}\,\mathbf{R}_{b1}^{I}$ is also orthogonal.

B. Summary of Quadrotor Equations of Motion

The following equations summarize the dynamic model of the quadrotor system. The bold symbols denote the control inputs to the system ($\boldsymbol{T}, \boldsymbol{\tau_\phi}, \boldsymbol{\tau_\theta}, \boldsymbol{\tau_\psi}$).

FORCE EQUATIONS:

$$\begin{array}{rclclcll}
\dot{u} & = & rv - qw & - & g\sin\theta & - & \frac{k_s}{m}(u - w_x)|u - w_x| & \\
\dot{v} & = & pw - ru & + & g\cos\theta\sin\phi & - & \frac{k_s}{m}(v - w_y)|v - w_y| & \\
\dot{w} & = & qu - pv & + & g\cos\theta\cos\phi & - & \frac{k_u}{m}(w - w_z)|w - w_z| & - \ \frac{1}{m}\boldsymbol{T}
\end{array} \tag{B.1}$$

KINEMATIC EQUATIONS:

$$\begin{array}{rcl}
\dot{\phi} & = & p + (q\sin\phi + r\cos\phi)\tan\theta \\
\dot{\theta} & = & q\cos\phi - r\sin\phi \\
\dot{\psi} & = & (q\sin\phi + r\cos\phi)/\cos\theta
\end{array} \tag{B.2}$$

MOMENT EQUATIONS:

$$\begin{array}{rcrl}
\dot{p} & = & \frac{1}{I_{xx}I_{zz} - I_{xz}^2} & \{I_{xz}(I_{xx} - I_{yy} + I_{zz})pq + (I_{yy}I_{zz} - I_{zz}^2 - I_{xz}^2)qr \\
 & & & \quad +I_{zz}I_r\Omega_r q + I_{zz}\boldsymbol{\tau_\phi} + I_{xz}\boldsymbol{\tau_\psi} + I_{zz}\tau_{w,\phi} + I_{xz}\tau_{w,\psi}\} \\
\dot{q} & = & \frac{1}{I_{yy}} & \{(I_{zz} - I_{xx})pr + I_{xz}(r^2 - p^2) - I_r\Omega_r p + \boldsymbol{\tau_\theta} + \tau_{w,\theta}\} \\
\dot{r} & = & \frac{1}{I_{xx}I_{zz} - I_{xz}^2} & \{(I_{xx}^2 - I_{xx}I_{yy} + I_{xz}^2)pq + I_{xz}(-I_{xx} + I_{yy} - I_{zz})qr \\
 & & & \quad +I_{xz}I_r\Omega_r q + I_{xx}\boldsymbol{\tau_\psi} + I_{xz}\boldsymbol{\tau_\phi} + I_{xx}\tau_{w,\psi} + I_{xz}\tau_{w,\phi}\}
\end{array} \tag{B.3}$$

NAVIGATION EQUATIONS:

$$\begin{array}{rcccccc}
\dot{N} & = & c\psi c\theta\, u & + & (c\psi s\theta s\phi - s\psi c\phi)\, v & + & (c\psi s\theta c\phi + s\psi s\phi)\, w \\
\dot{E} & = & s\psi c\theta\, u & + & (s\psi s\theta s\phi + c\psi c\phi)\, v & + & (s\psi s\theta c\phi - c\psi s\phi)\, w \\
\dot{h} & = & s\theta\, u & - & c\theta s\phi\, v & - & c\theta c\phi\, w
\end{array} \tag{B.4}$$

C. Quadrotor Simulation: Simulink Structure

The Matlab Simulink model in Figure C.1 was used to simulate and control the full nonlinear quadrotor. The controllers within the subsystems of Figure C.1 were coded with Simulink S-functions. The trajectory generation algorithm was also coded with Simulink S-functions.

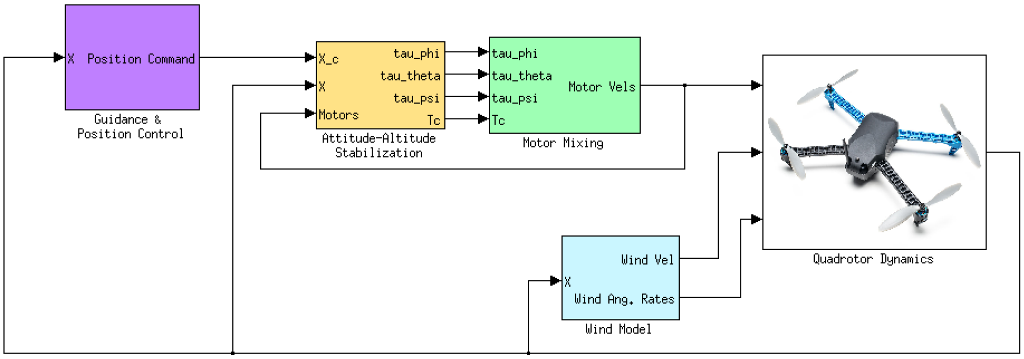


Figure C.1.. Simulink model of quadrotor

The wind disturbances were simulated with the aid of the following built-in Simulink models: horizontal wind model, Dryden discrete wind gust model, and Dryden continuous turbulence model, as show in Figure C.2. The second discrete wind gust block was used to cancel the wind gust generated by the first block (after a predetermined number of seconds), as there is no built-in method for ending a wind gust.

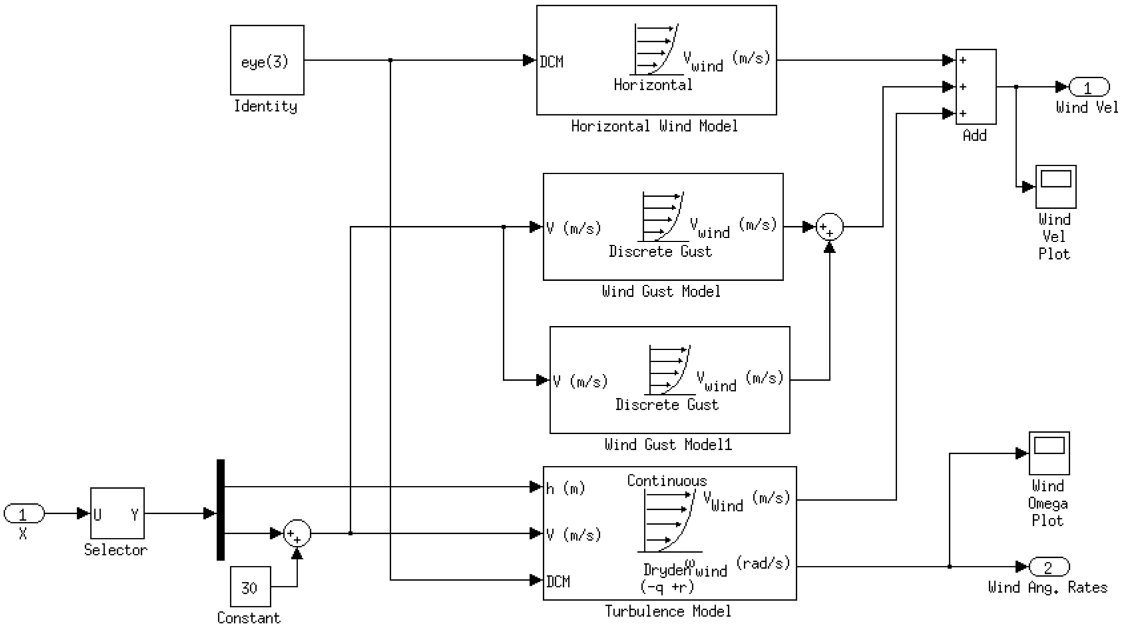


Figure C.2.. Simulink model of wind disturbances

VITA

VITA

David Kun grew up in Jerusalem, Israel and served for three years in the Israeli Air Force as a technician, working on the Sikorski CH-53 'Sea-Stallion' helicopter. Following his mandatory military service, David enrolled at Purdue University in the fall of 2009.

As an undergraduate student, David majored in Dynamics & Control and minored in Aerodynamics in the School of Aeronautics and Astronautics. He graduated with distinction in 2013 with his B.S. in Aeronautical & Astronautical Engineering. He proceeded to pursue a Master's degree at Purdue University, with a focus on guidance, navigation, and nonlinear control of unmanned aerial vehicles. Besides his thesis work, David worked on several projects, including: developing a software testbed for Hyperspectral Unmixing algorithms, integrating GPS/INS measurements with cascaded unscented Kalman filters, and a collaborative research effort with C. Kwon on cooperative multi-agent systems. During the summer of 2014, David researched ADS-B 'Sense & Avoid' technology for unmanned aerial vehicles at the NASA Armstrong Flight Research Center.

After obtaining his M.S. in Aeronautics & Astronautics in May of 2015, David will begin to work for Rockwell Collins in Cedar Rapids, Iowa. He will be joining the Flight Control Systems / Control Laws division as a Systems Engineer.