Guidance and Model Predictive Control of a Tandem-rotor Helicopter

by

Faraaz Ahmed

Department of Mechanical Engineering
McGill University, Montreal

August 2022

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Master of Science in Mechanical Engineering

Supervisor:

Professor James Richard Forbes

Faraaz Ahmed
faraaz.ahmed@mail.mcgill.ca

Abstract

The tandem-rotor helicopter is a platform of increased interest for autonomous development. This thesis considers the guidance and model predictive control (MPC) of a tandem-rotor helicopter. First, the dynamics of the tandem-rotor system are developed and a simplified dynamic model is presented. An error definition is formulated using the matrix Lie group $SE_2(3)$, which is used to linearize the system. Next, a computationally simple, two-stage guidance strategy is presented. A coarse reference trajectory to a target is calculated using a quartic guidance law, leveraging the differentially flat properties of the system. The trajectory is then refined using a finite-horizon linear quadratic regulator (LQR). To control the system, an MPC strategy is selected for its ability to directly incorporate constraints. The nonlinear system is linearized about the reference trajectory enabling the formulation of a quadratic program (QP) with control input and state constraints. A novel method of constraining the attitude is developed through the use of a linearized attitude keep-in zone and an ℓ_1-norm constraint on the attitude error. A non-uniformly spaced prediction horizon is leveraged in the MPC formulation to capture the multi-timescale vehicle dynamics while keeping the problem size tractable. Monte-Carlo simulations are performed to demonstrate the robustness of the proposed guidance and control structure to random initial conditions, model uncertainty, and environmental disturbances. Lastly, a cascaded MPC approach is considered as an alternative control strategy with a smaller computational footprint. Further Monte-Carlo simulations are carried out to compare the performance of the cascaded MPC approach with the single MPC structure.

Résumé

L'hélicoptère tandem-rotor est une plateforme d'intérêt croissant pour le développement autonome. Cette thèse considère le guidage et le contrôle prédictif de modèle (MPC) d'un hélicoptère tandem-rotor. Tout d'abord, la dynamique du système tandem-rotor est développée et un modèle dynamique simplifié est présenté. Une définition d'erreur est formulée en utilisant le groupe de Lie matriciel $SE_2(3)$, qui est utilisé pour linéariser le système. Ensuite, une stratégie de guidage en deux étapes, simple à calculer, est présentée. Une trajectoire de référence grossière vers une cible est calculée à l'aide d'une loi de guidage quartique, en exploitant les propriétés différentiellement plates du système. La trajectoire est ensuite raffinée à l'aide d'une régulateur linéaire quadratique (LQR) à horizon fini. Pour contrôler le système, une stratégie MPC est sélectionnée pour sa capacité à incorporer directement les contraintes. Le système non linéaire est linéarisé autour de la trajectoire de référence, ce qui permet la formulation d'un programme quadratique (QP) avec des contraintes d'entrée de commande et d'état. Une nouvelle méthode de contrainte de l'attitude est développée par l'utilisation d'une zone de maintien d'attitude linéarisée et d'une contrainte de norme ℓ_1 sur l'erreur d'attitude. Un horizon de prédiction à espacement non uniforme est utilisé dans la formulation du MPC pour capturer la dynamique du véhicule à plusieurs échelles de temps tout en maintenant la taille du problème raisonnable. Des simulations de Monte-Carlo sont effectuées pour démontrer la robustesse de la structure de guidage et de contrôle proposée aux conditions initiales aléatoires, à l'incertitude du modèle et aux perturbations environnementales. Enfin, une approche MPC en cascade est considérée comme une stratégie de contrôle alternative avec une empreinte de calcul plus petite. D'autres simulations de Monte-Carlo sont effectuées pour comparer les performances de l'approche MPC en cascade avec la structure MPC simple.

Acknowledgements

I would firstly like to thank Professor James Richard Forbes for his mentorship and support over the past two years. His expertise and dedication to his students is exceptional. I am grateful to Alex Walsh for introducing me to Professor Forbes and to both of them for helping convince me to pursue graduate studies.

I would also like to acknowledge NGC Aerospace, Mitacs and NSERC who supported this research through the Mitacs Accelerate and CGS-M programs. Specifically, I would like to thank Ludwik Sobiesiak, a true GNC master, for assisting me at NGC.

Thank you to the members of the DECAR systems group for creating a fantastic, supportive research team who set the bar of achievement incredibly high.

I greatly appreciate the support of my family, who have made my many long-distance moves much easier and have always encouraged me to do my best. Lastly, I would like to thank Kelsey and Mogo for their unending love and support. Starting and completing a degree through a global pandemic was not easy, but I am forever grateful to have you by my side.

Preface

The contributions of this thesis that are original to the author's knowledge are as follows.

- Chapter 4
    - The use of a two-step trajectory generation scheme where a coarse trajectory is first generated using quartic polynomials and refined using a discrete-time, finite-horizon LQR algorithm.
- Chapter 5
    - The formulation of an MPC problem using an augmented $SE_2(3)$ error definition.
    - The use of a linearized keep-in zone constraint on the attitude along with an ℓ_1-norm constraint on the attitude error to effectively limit the roll and pitch of the vehicle in an MPC framework.
- Chapter 6
    - The use of a cascaded MPC structure in the context of UAV control where the dynamic model used for inner-loop control is linearized about the outputs of the outer-loop controller.

All text, plots, figures, and results in this thesis are produced by Faraaz Ahmed.

Table of Contents

Chapter

List of Figures

Figure

List of Tables

Table

List of Abbreviations

MPC	Model predictive control
NMPC	Nonlinear model predictive control
LQR	Linear quadratic regulator
UAV	Unmanned aerial vehicle
PI	Proportional-integral
RMSE	Root-mean-square error
DCM	Direction cosine matrix
LTI	Linear time-invariant
LTV	Linear time-varying
NLP	Nonlinear program
QP	Quadratic program

List of Symbols

$\left\|\cdot\right\|_p$	the p-norm
$\mathbb{R}^n$	the vector space of real n-dimensional vectors
$\mathbb{R}^{n\times m}$	the space of real $m \times n$-dimensional matrices
$(\cdot)^\mathsf{T}$	transpose
$(\cdot)^\times$	cross operator for $\mathfrak{so}(3)$
$(\cdot)^\wedge$	operator mapping an element of $\mathbb{R}^d$ to $\mathfrak{g}$
$(\cdot)^\vee$	operator mapping an element of $\mathfrak{g}$ to $\mathbb{R}^d$
$\mathbf{0}$	zero matrix
$\mathbf{1}$	identity matrix
$\mathcal{F}_i$	reference frame i
$\underrightarrow{u}$	a physical vector
$\underrightarrow{r}^{zw}$	the position of point z relative to point w
$\underrightarrow{r}^{\bullet a}$	the time derivative of $\underrightarrow{r}$ with respect for $\mathcal{F}_a$
$\underrightarrow{r}^{zw^{\bullet a}} = \underrightarrow{v}^{zw/a}$	the velocity of point z relative to point w with respect to $\mathcal{F}_a$
$\underrightarrow{\boldsymbol{\mathcal{F}}}_a$	a vectrix, that is a matrix of unit length physical vectors that form a basis for $\mathcal{F}_a$, where $\underrightarrow{\boldsymbol{\mathcal{F}}}_a^\mathsf{T} = [\underrightarrow{a}^1 \;\; \underrightarrow{a}^2 \;\; \underrightarrow{a}^3]$
$\mathbf{u}_a$	the physical vector $\underrightarrow{u}$ resolved in the reference frame $\mathcal{F}_a$
$\mathbf{C}_{ab}$	a direction cosine matrix describing the attitude of $\mathcal{F}_a$ relative to $\mathcal{F}_b$
$\underrightarrow{\omega}^{ba}$	the angular velocity of $\mathcal{F}_b$ relative to $\mathcal{F}_a$
$\mathrm{diag}(\mathbf{A}_1, \cdots, \mathbf{A}_n)$	block diagonal matrix with $\mathbf{A}_1, \cdots, \mathbf{A}_n$ on the diagonals, and zeros elsewhere

tr($\mathbf{A}$)	the trace of square matrix $\mathbf{A}$
det($\mathbf{A}$)	the determinant of square matrix $\mathbf{A}$

Chapter 1

Introduction

The number of applications for unmanned aerial vehicles (UAVs) are growing, now including delivery, search and rescue, surveillance, and inspection [1, 2]. While quadrotor and traditional helicopter platforms are popular for these types of tasks, a tandem-rotor helicopter offers several advantages, including a large center of mass range, and larger lift capacities with smaller rotors [3]. The ability to successfully complete this growing list of missions is dependent on the ability to operate autonomously in a variety of different environments. Autonomous operation of a UAV consists of solving the navigation, guidance, and control problems simultaneously and continuously. The navigation estimates the current states of the UAV, while the guidance is responsible for trajectory planning, and the control must generate inputs for the vehicle's actuators. Improved navigation, guidance and control methods enable the tandem-rotor helicopter to operate in challenging environmental conditions, such as those with high winds and a large number of obstacles. This thesis focuses on the guidance and control problems and assumes that an external navigation algorithm is capable of providing accurate state estimates.

The ability to effectively enforce state and control input constraints are vital for the safe operation of UAVs. Vehicle actuators such as servos, motors, and swashplates have finite control authority. Commands from a controller that are unachievable by the actuators can result in loss of tracking performance, instabilities, and in the worst case, loss of control [4]. Similarly, certain vehicle states such as attitude, position, and velocity should be constrained to avoid dangerous operating conditions or physical obstacles in the surrounding environment. Model predictive control (MPC) is a method of optimal control extensively used in industry for its ability to explicitly enforce constraints in an optimization problem [5]. Because an optimization problem must be solved at each timestep, which can be computationally expensive, MPC has historically been used in process control applications with slow dynamics [6]. However, rapid developments in small-scale computing has resulted in the ability for MPC to be implemented in real-time on faster systems [7]. Still, for large systems with fast dynamics, real-time MPC can be difficult to implement [8]. In addition to having fast, nonlinear dynamics, the tandem-rotor helicopter has a large number of system states and control inputs. Therefore, a number of challenges must be overcome to successfully implement MPC.

1.1 Objectives

The objective of this thesis is to develop a robust guidance and control strategy for the tandem-rotor helicopter platform that will enable it to plan and accurately track a desired trajectory. First, a computationally lightweight, optimal guidance strategy is developed, which can compute a desired reference trajectory from the current location to a target. The ability to recompute the reference trajectory in real-time is critical to the robustness of the guidance and control structure. The second objective is to develop an MPC strategy that can be implemented in a real-time environment using limited computing platforms. To achieve this objective, several methodologies are investigated including a non-uniformly spaced MPC prediction horizon, system linearization using an augmented matrix Lie group error definition along a reference trajectory, and a cascaded MPC formulation. The use of input and state constraints is heavily leveraged in the MPC formulation. Specifically, a major contribution of this thesis is the synthesis of a linearized attitude keep-in zone and an ℓ_1-norm constraint on the attitude error, which together enforce attitude constraint satisfaction.

1.2 Organization

The remainder of this thesis is organized as follows.

- Chapter 2 introduces several relevant mathematical concepts that are used throughout this thesis including geometry, kinematics, matrix Lie groups, differential flatness, and optimization.
- Chapter 3 presents the dynamic model used to represent the tandem-rotor system. The linearization and discretization procedures used in the guidance and control algorithms are also presented in detail. Additionally, the model used to introduce environmental disturbances in simulations is presented.
- Chapter 4 details the guidance algorithm developed for online trajectory generation. The two-stage guidance strategy is described along with simulations demonstrating the

trajectory generation process for a landing maneuver.

- Chapter 5 presents the control strategy developed for the tandem-rotor helicopter. A brief introduction to MPC is given along with the QP formulation for the linearized tandem-rotor system. The controller performance is demonstrated through simulations of a landing maneuver.
- Chapter 6 considers an alternative control strategy by splitting the proposed MPC controller from Chapter 5 into a cascaded control structure. Additional simulation results are presented to demonstrate the improvements in computation time attained using this method.
- Chapter 7 provides a summary of the findings, concluding remarks and recommendations for possible future work.

Chapter 2

Preliminaries

In this chapter, various mathematical concepts and tools are briefly introduced. These tools are used extensively throughout this thesis to develop the guidance and control algorithms for the tandem-rotor helicopter.

2.1 Geometry

2.1.1 Physical Vectors and Reference Frames

A physical vector $\underrightarrow{u}$, is an element with both magnitude and direction that is used to describe how a body moves in physical space. A reference frame $\mathcal{F}_a$, is defined by three orthonormal physical basis vectors, $\underrightarrow{a}^1$, $\underrightarrow{a}^2$, and $\underrightarrow{a}^3$ [9]. The vectrix $\underrightarrow{\boldsymbol{\mathcal{F}}}_a$, is defined as

$$\underrightarrow{\boldsymbol{\mathcal{F}}}_a = \begin{bmatrix} \underrightarrow{a}^1 \\ \underrightarrow{a}^2 \\ \underrightarrow{a}^3 \end{bmatrix}. \tag{2.1}$$

A physical vector $\underrightarrow{u}$ can be written as

$$\begin{aligned} \underrightarrow{u} &= \underrightarrow{\boldsymbol{\mathcal{F}}}_a^{\mathsf{T}} \mathbf{u}_a, && (2.2) \\ &= \begin{bmatrix} \underrightarrow{a}^1 & \underrightarrow{a}^2 & \underrightarrow{a}^3 \end{bmatrix} \begin{bmatrix} u_a^1 \\ u_a^2 \\ u_a^3 \end{bmatrix}, && (2.3) \end{aligned}$$

where $\mathbf{u}_a$ are the components of $\underrightarrow{u}$ resolved in $\mathcal{F}_a$.

2.1.2 The Direction Cosine Matrix

The orientation of $\mathcal{F}_a$, relative to a second reference frame $\mathcal{F}_b$, is given by a direction cosine matrix (DCM) $\mathbf{C}_{ab} \in SO(3)$. The DCM $\mathbf{C}_{ab}$ is defined as

$$\begin{aligned}
\mathbf{C}_{ab} &= \underrightarrow{\boldsymbol{\mathcal{F}}}_a \cdot \underrightarrow{\boldsymbol{\mathcal{F}}}_b^{\mathsf{T}}, && (2.4)\\
&= \begin{bmatrix} \underrightarrow{a}^1 \\ \underrightarrow{a}^2 \\ \underrightarrow{a}^3 \end{bmatrix} \cdot \begin{bmatrix} \underrightarrow{b}^1 & \underrightarrow{b}^2 & \underrightarrow{b}^3 \end{bmatrix}. && (2.5)
\end{aligned}$$

The DCM $\mathbf{C}_{ab}$ and the DCM $\mathbf{C}_{ba}$ are related in the following way,

$$\mathbf{C}_{ba} = \mathbf{C}_{ab}^{-1} = \mathbf{C}_{ab}^{\mathsf{T}}. \qquad (2.6)$$

Additionally, since $\mathbf{C}_{ab}$ and $\mathbf{C}_{ba}$ are orthonormal matrices,

$$\mathbf{C}_{ba}\mathbf{C}_{ab} = \mathbf{C}_{ab}^{\mathsf{T}}\mathbf{C}_{ab} = \mathbf{C}_{ab}^{-1}\mathbf{C}_{ab} = \mathbf{1}. \qquad (2.7)$$

A physical vector $\underrightarrow{u}$ can be resolved in either $\mathcal{F}_a$ as $\mathbf{u}_a$, or in $\mathcal{F}_b$ as $\mathbf{u}_b$. The two are related by

$$\mathbf{u}_a = \mathbf{C}_{ab}\mathbf{u}_b = \mathbf{C}_{ba}^{\mathsf{T}}\mathbf{u}_b. \qquad (2.8)$$

The column-matrix $\mathbf{q}^{ba}$ is a parameterization of the DCM $\mathbf{C}_{ba}$ defined as

$$\mathbf{q}^{ba} = \bar{\mathbf{c}}_{ba} = \begin{bmatrix} \bar{\mathbf{c}}_{ba}^1 \\ \bar{\mathbf{c}}_{ba}^2 \\ \bar{\mathbf{c}}_{ba}^3 \end{bmatrix}, \qquad (2.9)$$

where

$$\mathbf{C}_{ab} = \begin{bmatrix} \bar{\mathbf{c}}_{ba}^1 & \bar{\mathbf{c}}_{ba}^2 & \bar{\mathbf{c}}_{ba}^3 \end{bmatrix}. \qquad (2.10)$$

2.2 Kinematics

Kinematics describe the motion of geometry without concern for the cause of the motion. The position of a point z relative to another point w is described by the physical vector $\underrightarrow{r}^{zw}$.

The time rate of change of $\underset{\rightarrow}{r}^{zw}$ with respect to frame $\mathcal{F}_a$ is given by

$$\underset{\rightarrow}{r}^{zw^{\bullet a}} = \underset{\rightarrow}{v}^{zw/a}, \tag{2.11}$$

where $\underset{\rightarrow}{v}^{zw/a}$ is the velocity of point z relative to point w, with respect to $\mathcal{F}_a$. Similarly, the time rate of change of $\underset{\rightarrow}{v}^{zw/a}$ with respect to $\mathcal{F}_a$ is given by

$$\underset{\rightarrow}{r}^{zw^{\bullet a \bullet a}} = \underset{\rightarrow}{v}^{zw/a^{\bullet a}} = \underset{\rightarrow}{a}^{zw/a/a}, \tag{2.12}$$

where $\underset{\rightarrow}{a}^{zw/a/a}$ is the acceleration of point z relative to point w, with respect to $\mathcal{F}_a$. The angular velocity of $\mathcal{F}_b$ relative to $\mathcal{F}_a$ is described by the physical vector $\underset{\rightarrow}{\omega}^{ba}$. The physical vectors for position, velocity, acceleration, and angular velocity can all be resolved in a particular reference frame using (2.2).

The time rate of change of the DCM $\mathbf{C}_{ab}$ is given by Poisson's equation

$$\dot{\mathbf{C}}_{ab} = \mathbf{C}_{ab}\boldsymbol{\omega}_b^{ba^\times}. \tag{2.13}$$

In discrete-time, Poisson's equation becomes

$$\mathbf{C}_{ab_{k+1}} = \mathbf{C}_{ab_k} \exp\left(\left(T\boldsymbol{\omega}_{b_k}^{ba}\right)^\times\right), \tag{2.14}$$

where $T = t_{k+1} - t_k$ is the discretization timestep.

2.3 Matrix Lie Groups

Matrix Lie groups provide a framework that allow various vehicle states, such as attitude, velocity, and position, to be described by a single matrix element. This matrix structure is exploited to form an *invariant* error definition when linearizing the system dynamics, the benefit of which is discussed in Chapter 3. In this section, the two matrix Lie groups used in this thesis, $SO(3)$, and $SE_2(3)$, are reviewed.

2.3.1 Overview

Let G denote a matrix Lie group and let $\mathfrak{g}$ denote the matrix Lie algebra associated with G [10]. An element of $\mathfrak{g}$ can be mapped to G using the exponential map, $\exp(\cdot) : \mathfrak{g} \rightarrow G$ and the inverse operation is achieved using the logarithmic map, $\log(\cdot) : G \rightarrow \mathfrak{g}$. For matrix Lie groups, the exponential map is the matrix exponential and the logarithmic map is the matrix natural logarithm. The matrix Lie algebra is mapped to a k dimensional column matrix using the linear "vee" operator $(\cdot)^\vee : \mathfrak{g} \rightarrow \mathbb{R}^k$, and the inverse operation is performed using the "wedge" operator $(\cdot)^\wedge : \mathbb{R}^k \rightarrow \mathfrak{g}$.

2.3.2 Linearization

An element of a matrix Lie group can be expressed in terms of an element of the Lie algebra using the exponential map,

$$\mathbf{X} = \exp\left(\boldsymbol{\xi}^\wedge\right). \tag{2.15}$$

The exponential map is defined by the power series

$$\begin{aligned} \exp\left(\boldsymbol{\xi}^\wedge\right) &= \sum_{k=0}^{\infty} \frac{1}{k!} \left(\boldsymbol{\xi}^\wedge\right)^k, &(2.16)\\ &= \mathbf{1} + \boldsymbol{\xi}^\wedge + \frac{\left(\boldsymbol{\xi}^\wedge\right)^2}{2} + \frac{\left(\boldsymbol{\xi}^\wedge\right)^3}{6} + \cdots. &(2.17)\end{aligned}$$

Consider a small perturbation $\delta\boldsymbol{\xi}$ such that

$$\delta\mathbf{X} = \exp(\delta\boldsymbol{\xi}^\wedge). \tag{2.18}$$

Then a first order approximation of (2.18) can be obtained from (2.17) by neglecting terms of order $\mathcal{O}\left(\|\delta\boldsymbol{\xi}^\wedge\|^2\right)$, resulting in

$$\delta\mathbf{X} \approx \mathbf{1} + \delta\boldsymbol{\xi}^\wedge. \tag{2.19}$$

2.3.3 The Special Orthogonal Group $SO(3)$

The matrix Lie group $SO(3)$ is used to represent three-dimensional rotations and is given by [11]

$$SO(3) = \left\{\mathbf{C} \in \mathbb{R}^{3\times3} \mid \mathbf{C}^\mathsf{T}\mathbf{C} = \mathbf{1},\ \det(\mathbf{C}) = 1\right\}, \tag{2.20}$$

where the matrix $\mathbf{C}$ is a DCM. The Lie algebra associated with $SO(3)$ is $\mathfrak{so}(3)$ and is given by

$$\mathfrak{so}(3) = \left\{\boldsymbol{\phi}^\times \in \mathbb{R}^{3\times3} \mid \boldsymbol{\phi} \in \mathbb{R}^{3\times3}\right\}, \tag{2.21}$$

where the cross operator $(\cdot)^\times$ maps elements of $\mathbb{R}^3$ to skew-symmetric matrices in $\mathbb{R}^{3\times 3}$ and is defined as

$$\boldsymbol{\phi}^\times = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}^\times = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix}. \tag{2.22}$$

The identities [11],

$$\mathbf{u}^\times \mathbf{v} = -\mathbf{v}^\times \mathbf{u}, \tag{2.23}$$

$$\left(\mathbf{u}^\times \mathbf{v}\right)^\times = \mathbf{u}^\times \mathbf{v}^\times - \mathbf{v}^\times \mathbf{u}^\times, \tag{2.24}$$

are used in various derivations in this thesis. When working with elements of $\mathfrak{so}(3)$, the cross operator is equivalent to the wedge operator $(\cdot)^\wedge$ seen elsewhere.

The exponential map from $\mathfrak{so}(3)$ to $SO(3)$ is given by Rodrigues formula

$$\exp_{SO(3)}\left(\boldsymbol{\phi}^\times\right) = \cos\phi \mathbf{1} + (\mathbf{1} - \cos\phi)\, \mathbf{a}\mathbf{a}^\mathsf{T} + \sin\phi \mathbf{a}^\times, \tag{2.25}$$

where $\phi = \|\boldsymbol{\phi}\|$ and $\mathbf{a} = \boldsymbol{\phi}/\phi$. The logarithmic map from $SO(3)$ to $\mathfrak{so}(3)$ is given by

$$\log_{SO(3)}(\mathbf{C}) = (\mathbf{a}\phi)^\times, \tag{2.26}$$

where the angle ϕ is given by

$$\phi = \cos^{-1}\left(\frac{\operatorname{tr}(\mathbf{C}) - 1}{2}\right), \tag{2.27}$$

and the axis $\mathbf{a}$ is given by

$$\mathbf{a} = \frac{1}{2\sin(\phi)} \begin{bmatrix} \mathbf{C}_{2,3} - \mathbf{C}_{3,2} \\ \mathbf{C}_{3,1} - \mathbf{C}_{1,3} \\ \mathbf{C}_{1,2} - \mathbf{C}_{2,1} \end{bmatrix}. \tag{2.28}$$

The left Jacobian $\mathbf{J}(\boldsymbol{\phi})$ and its inverse are given by

$$\mathbf{J}(\boldsymbol{\phi}) = \frac{\sin\phi}{\phi}\mathbf{1} + \left(1 - \frac{\sin\phi}{\phi}\right)\mathbf{a}\mathbf{a}^\mathsf{T} + \frac{1-\cos\phi}{\phi}\mathbf{a}^\times, \tag{2.29}$$

$$\mathbf{J}(\boldsymbol{\phi})^{-1} = \frac{\phi}{2}\cot\frac{\phi}{2}\mathbf{1} + \left(1 - \frac{\phi}{2}\cot\frac{\phi}{2}\right)\mathbf{a}\mathbf{a}^\mathsf{T} - \frac{\phi}{2}\mathbf{a}^\times, \tag{2.30}$$

where $\phi = \left\| \boldsymbol{\xi}^{\phi} \right\|$ and $\mathbf{a} = \boldsymbol{\xi}^{\phi}/\phi$.

2.3.4 The Group of Double Direct Isometries $SE_2(3)$

The group of double direct isometries, $SE_2(3)$, introduced by [10], is given by

$$SE_2(3) = \left\{ \mathbf{X} = \begin{bmatrix} \mathbf{C} & \mathbf{v} & \mathbf{r} \\ \mathbf{0} & 1 & 0 \\ \mathbf{0} & 0 & 1 \end{bmatrix} \in \mathbb{R}^{5\times 5} \middle| \mathbf{C} \in SO(3),\ \mathbf{v},\ \mathbf{r} \in \mathbb{R}^3 \right\}, \tag{2.31}$$

where $\mathbf{C}$ is a DCM, $\mathbf{v}$ is the velocity, and $\mathbf{r}$ is the position. The associated Lie algebra, $\mathfrak{se}_2(3)$ is given by

$$\mathfrak{se}_2(3) = \left\{ \boldsymbol{\Xi} = \boldsymbol{\xi}^\wedge \in \mathbb{R}^{5\times 5} \mid \boldsymbol{\xi} \in \mathbb{R}^9 \right\}, \tag{2.32}$$

where

$$\boldsymbol{\xi}^\wedge = \begin{bmatrix} \boldsymbol{\xi}^{\phi} \\ \boldsymbol{\xi}^{v} \\ \boldsymbol{\xi}^{r} \end{bmatrix}^\wedge = \begin{bmatrix} \boldsymbol{\xi}^{\phi^\times} & \boldsymbol{\xi}^{v} & \boldsymbol{\xi}^{r} \\ \mathbf{0} & 0 & 0 \\ \mathbf{0} & 0 & 0 \end{bmatrix}. \tag{2.33}$$

The exponential map from $\mathfrak{se}_2(3)$ to $SE_2(3)$ is

$$\exp_{SE_2(3)}\left(\boldsymbol{\xi}^\wedge\right) = \begin{bmatrix} \exp_{SO(3)}\left(\boldsymbol{\xi}^{\phi^\times}\right) & \mathbf{J}\left(\boldsymbol{\xi}^{\phi}\right)\boldsymbol{\xi}^{v} & \mathbf{J}\left(\boldsymbol{\xi}^{\phi}\right)\boldsymbol{\xi}^{r} \\ \mathbf{0} & 0 & 0 \\ \mathbf{0} & 0 & 0 \end{bmatrix}, \tag{2.34}$$

where $\mathbf{J}(\boldsymbol{\xi}^{\phi})$ is the left Jacobian given by (2.29). The logarithmic map from $SE_2(3)$ to $\mathfrak{se}_2(3)$ is

$$\log_{SE_2(3)}(\mathbf{X}) = \begin{bmatrix} \log_{SO(3)}(\mathbf{C}) & \mathbf{J}\left(\boldsymbol{\xi}^{\phi}\right)^{-1}\mathbf{v} & \mathbf{J}\left(\boldsymbol{\xi}^{\phi}\right)^{-1}\mathbf{r} \\ \mathbf{0} & 0 & 0 \\ \mathbf{0} & 0 & 0 \end{bmatrix}, \tag{2.35}$$

where $\mathbf{J}(\boldsymbol{\xi}^{\phi})^{-1}$ is the inverse of the left Jacobian given by (2.30).

2.4 Differential Flatness

Consider a nonlinear system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \tag{2.36}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}), \tag{2.37}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, and $\mathbf{y} \in \mathbb{R}^m$. The system is called differentially flat if a set of outputs $\mathbf{z} \in \mathbb{R}^m$ of the form [12]

$$\mathbf{z} = \boldsymbol{\zeta}\left(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, \cdots, \mathbf{u}^{(l)}\right) \tag{2.38}$$

can be found such that

$$\mathbf{x} = \mathbf{x}\left(\mathbf{z}, \dot{\mathbf{z}}, \cdots, \mathbf{z}^{(l)}\right), \tag{2.39}$$

$$\mathbf{u} = \mathbf{u}\left(\mathbf{z}, \dot{\mathbf{z}}, \cdots, \mathbf{z}^{(l)}\right). \tag{2.40}$$

The outputs $\mathbf{z}$ are called the *flat outputs*. Note from (2.38) that the flat outputs must be a function of the state $\mathbf{x}$, as well as the control input $\mathbf{u}$, and any number of derivatives of $\mathbf{u}$. The complete system state and control inputs can be derived from the flat outputs and derivatives of the flat outputs, as indicated by (2.39) and (2.40). Therefore, the flat outputs completely characterize the behaviour of the system.

This property, first introduced in [13], is especially useful in trajectory generation, where a trajectory must only be planned in terms of the flat outputs. The complete set of state and control input trajectories can be directly derived from the flat outputs. Moreover, the derived state and control input trajectories are guaranteed to follow the system dynamics given by (2.36).

2.5 Optimization

A standard optimization problem is expressed as [14]

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{f}(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}) \leq 0, \end{aligned} \tag{2.41}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the *optimization variable*, $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}$ is the *objective function*, and $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^m$ is the *constraint function*. The solution to (2.41), denoted $\mathbf{x}^\star$, is the value of $\mathbf{x}$ which minimizes the objective function while satisfying the constraints.

2.5.1 Convex Functions

A function $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ is convex if it satisfies Jensen's inequality [14]

$$f(\alpha\mathbf{x} + \beta\mathbf{y}) \leq \alpha f(\mathbf{x}) + \beta f(\mathbf{y}), \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \quad \forall \alpha, \beta \in \mathbb{R}, \tag{2.42}$$

with $\alpha + \beta = 1,\ \alpha \geq 0,\ \beta \geq 0.$

2.5.2 Convex Optimization

A convex optimization problem is of identical form to that of (2.41), where the functions $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ are convex [14]. A solution to a convex optimization problem is also a global solution.

2.5.3 Quadratic Programming

A quadratic program (QP) is a subset of convex optimization problems where the objective function is quadratic and the constraint functions are affine [14]. A QP is expressed as

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \tfrac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{Q}\mathbf{x} + \mathbf{c}^\mathsf{T}\mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b}, \end{aligned} \tag{2.43}$$

where $\mathbf{Q} = \mathbf{Q}^\mathsf{T} > 0$. There are many numerical methods for solving QPs including interior-point methods [15] and active set methods [16].

Chapter 3

Tandem-rotor Dynamics

In this chapter, the tandem-rotor helicopter is introduced along with a simple dynamic model used to represent the system in the subsequent guidance and control algorithms. An augmented $SE_2(3)$ state error definition is presented, followed by the linearization and discretization procedure for the nonlinear rigid-body equations of motions. Lastly, the equations of motion used to model wind disturbances in simulation are detailed.

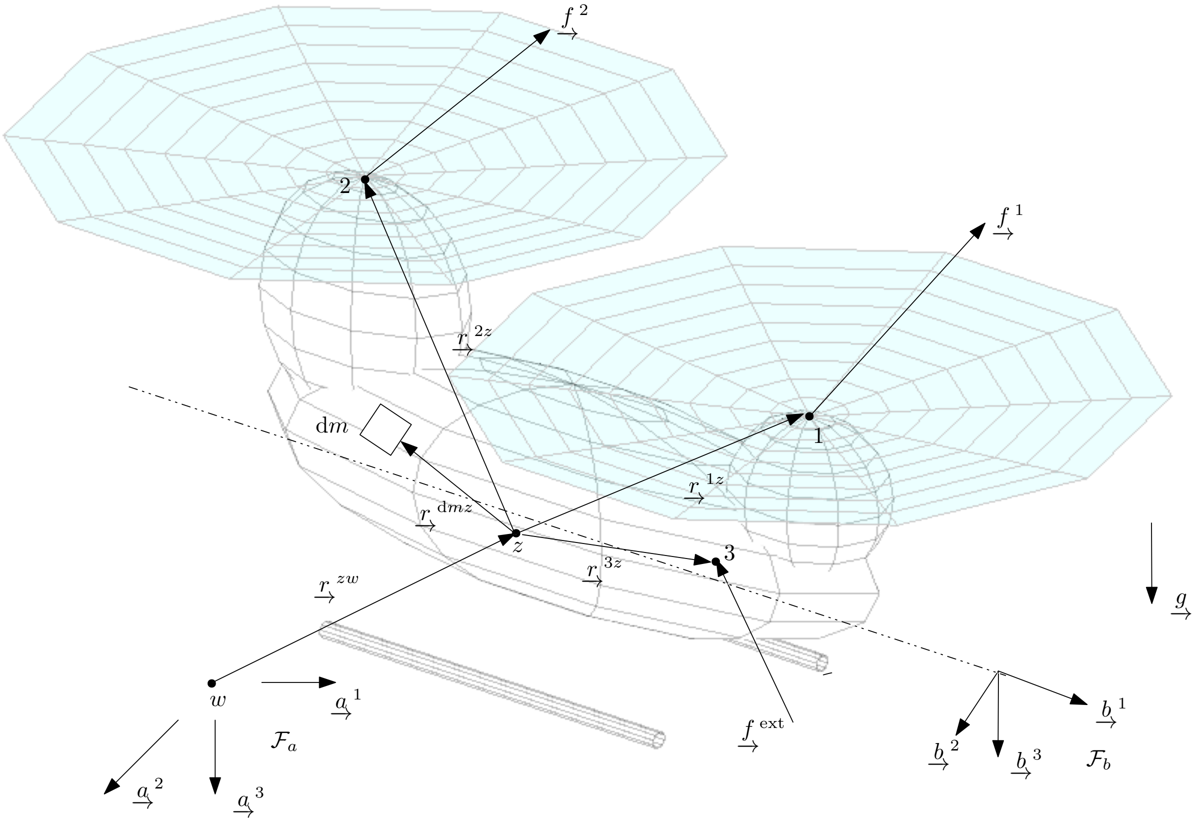
3.1 Tandem-rotor Helicopters

The tandem-rotor helicopter, as detailed in [3], is a type of twin-main-rotor helicopter where the two main-rotors are mounted along the longitudinal axis. Typically, the front and rear rotor disks are overlapped by approximately 35% and are therefore separated vertically. While single-rotor helicopters utilize a tail rotor to counteract the reaction torque from the main-rotor, the tandem-rotor helicopter uses counter-rotating main-rotors which cancel out the reaction torque from each other. In terms of the dynamic behaviour, the main difference between the tandem-rotor and single-rotor helicopters is in the longitudinal and yaw dynamics. The tandem-rotor helicopter achieves pitch control using differential collective, and yaw control using differential lateral cyclic. Roll control and vertical control are achieved through lateral cyclic, and collective of both rotors, respectively.

Tandem-rotor helicopters have traditionally been used in heavy lift and cargo operations due to their ability to lift larger payloads with smaller rotor diameters [17]. They generally have a larger center of mass range and good longitudinal stability characteristics. The most well-known application of the tandem-rotor design is the Boeing CH-46 Sea Knight and CH-47 Chinook, the latter of which is used worldwide for a variety of applications and has been produced for over 60 years [18].

3.2 Dynamic Model

The tandem-rotor helicopter is modeled as a rigid body subject to thrust, gravitational, and drag forces as shown in Figure 3.1.



$\underset{\rightarrow}{f}^{2}$
2
$\underset{\rightarrow}{f}^{1}$
$\underset{\rightarrow}{r}^{2z}$
dm
1
$\underset{\rightarrow}{r}^{1z}$
$\underset{\rightarrow}{r}^{\mathrm{d}mz}$
z
3
$\underset{\rightarrow}{r}^{3z}$
$\underset{\rightarrow}{r}^{zw}$
$\underset{\rightarrow}{g}$
w
$\underset{\rightarrow}{a}^{1}$
$\mathcal{F}_a$
$\underset{\rightarrow}{f}^{\mathrm{ext}}$
$\underset{\rightarrow}{b}^{1}$
$\underset{\rightarrow}{a}^{2}$
$\underset{\rightarrow}{a}^{3}$
$\underset{\rightarrow}{b}^{2}$
$\underset{\rightarrow}{b}^{3}$
$\mathcal{F}_b$

Figure 3.1: Rigid body model of tandem-rotor helicopter system.

Let z be a point collocated with the center of the mass of the helicopter. The kinematics are [9]

$$\dot{\mathbf{C}}_{ab} = \mathbf{C}_{ab}\boldsymbol{\omega}_b^{ba^\times}, \tag{3.1}$$

$$\dot{\mathbf{r}}_a^{zw} = \mathbf{v}_a^{zw/a}, \tag{3.2}$$

where $\mathbf{C}_{ab}$ is the orientation of $\mathcal{F}_a$ relative to $\mathcal{F}_b$, $\boldsymbol{\omega}_b^{ba}$ is the angular velocity of $\mathcal{F}_b$ relative to $\mathcal{F}_a$ resolved in $\mathcal{F}_b$, $\mathbf{r}_a^{zw}$ is the position of point z relative to point w resolved in $\mathcal{F}_a$, and $\mathbf{v}_a^{zw/a}$ is the velocity of point z relative to point w with respect to $\mathcal{F}_a$, resolved in $\mathcal{F}_a$.

The dynamics are [9]

$$m_{\mathcal{B}}\dot{\mathbf{v}}_a^{zw/a} = \mathbf{C}_{ab}\mathbf{f}_b^{\mathcal{B}z}, \tag{3.3}$$

$$\mathbf{J}_b^{\mathcal{B}z}\dot{\boldsymbol{\omega}}_b^{ba} = \mathbf{m}_b^{\mathcal{B}z} - \boldsymbol{\omega}_b^{ba^\times}\mathbf{J}_b^{\mathcal{B}z}\boldsymbol{\omega}_b^{ba}, \tag{3.4}$$

where $m_{\mathcal{B}}$ is the helicopter mass, and $\mathbf{J}_b^{\mathcal{B}z}$ is the helicopter's second moment of mass resolved in $\mathcal{F}_b$. The forces acting on the helicopter are

$$\mathbf{f}_b^{\mathcal{B}z} = \mathbf{f}_b^p + \mathbf{C}_{ab}^\mathsf{T}\mathbf{f}_a^a + \mathbf{C}_{ab}^\mathsf{T}\mathbf{f}_a^g, \tag{3.5}$$

where $\mathbf{f}_b^p = [0 \ 0 \ f]^\mathsf{T}$ is the propulsion force, $f = f^1 + f^2$ is the total thrust force from the rotors, $\mathbf{f}_a^a = \mathbf{C}_{ab}\mathbf{D}\mathbf{C}_{ab}^\mathsf{T}\mathbf{v}_a^{zw/a}$ is the aerodynamic drag force, $\mathbf{D} = \mathrm{diag}(d_x, d_y, d_z)$ is a constant matrix composed of rotor drag coefficients [19], and $\mathbf{f}_a^g = [0 \ 0 \ m_{\mathcal{B}}g]^\mathsf{T}$ is the gravitational force, where $g = 9.81$ m/s^2. The torques acting on the helicopter are

$$\mathbf{m}_b^{\mathcal{B}z} = \mathbf{m}_b + \mathbf{m}_b^a, \tag{3.6}$$

where $\mathbf{m}_b$ is the total control moment from the rotors, and $\mathbf{m}_b^a = -\mathbf{E}\mathbf{C}_{ab}^\mathsf{T}\mathbf{v}_a^{zw/a} - \mathbf{F}\boldsymbol{\omega}_b^{ba}$ is the parasitic torque from the rotor drag, where $\mathbf{E}$ and $\mathbf{F}$ are constant drag matrices [19].

3.3 Control Objective

The objective of the controller is to generate actuator commands that allow the tandem rotor helicopter to follow a defined reference trajectory. Denote the reference frame by $\mathcal{F}_r$, the reference attitude trajectory $\mathbf{C}_{ar}$, the reference velocity trajectory $\mathbf{v}_a^{z_rw/a}$, and the reference position trajectory $\mathbf{r}_a^{z_rw}$. The reference states at time step k are cast into elements of $SE_2(3)$ as

$$\mathbf{X}_k^r = \begin{bmatrix} \mathbf{C}_{ar_k} & \mathbf{v}_{a_k}^{z_rw/a} & \mathbf{r}_{a_k}^{z_rw} \\ \mathbf{0} & 1 & 0 \\ \mathbf{0} & 0 & 1 \end{bmatrix} \in SE_2(3). \tag{3.7}$$

The tracking error is defined using a left-invariant multiplicative error definition [10],

$$\delta\mathbf{X}_k = \mathbf{X}_k^{r^{-1}}\mathbf{X}_k = \begin{bmatrix} \delta\mathbf{C}_k & \delta\mathbf{v}_k & \delta\mathbf{r}_k \\ \mathbf{0} & 1 & 0 \\ \mathbf{0} & 0 & 1 \end{bmatrix} \in SE_2(3), \tag{3.8}$$

where the individual errors $\delta\mathbf{C}_k$, $\delta\mathbf{v}_k$, and $\delta\mathbf{r}_k$ are given by

$$\delta\mathbf{C}_k = \mathbf{C}_{ar_k}^{\mathsf{T}}\mathbf{C}_{ab_k}, \tag{3.9}$$

$$\delta\mathbf{v}_k = \mathbf{C}_{ar_k}^{\mathsf{T}}\left(\mathbf{v}_{a_k}^{zw/a} - \mathbf{v}_{a_k}^{z_r w/a}\right), \tag{3.10}$$

$$\delta\mathbf{r}_k = \mathbf{C}_{ar_k}^{\mathsf{T}}\left(\mathbf{r}_{a_k}^{zw} - \mathbf{r}_{a_k}^{z_r w}\right). \tag{3.11}$$

The tracking error is expressed in terms of the Lie algebra as

$$\delta\mathbf{X}_k = \exp\left(\delta\boldsymbol{\xi}_k^{\wedge}\right) = \begin{bmatrix} \exp_{SO(3)}\left(\delta\boldsymbol{\xi}^{\phi^{\times}}\right) & \mathbf{J}\left(\delta\boldsymbol{\xi}^{\phi}\right)\delta\boldsymbol{\xi}^{v} & \mathbf{J}\left(\delta\boldsymbol{\xi}^{\phi}\right)\delta\boldsymbol{\xi}^{r} \\ \mathbf{0} & 1 & 0 \\ \mathbf{0} & 0 & 1 \end{bmatrix}, \tag{3.12}$$

where $\mathbf{J}(\cdot)$ is the $SO(3)$ left Jacobian given by (2.29).

Note that the matrix Lie group $SE_2(3)$ does not include angular velocity as a state. If the system state is defined by a purely $SE_2(3)$ error definition, then the rotational dynamics given by (3.4) are no longer included in the system. Instead, the angular velocity becomes a control input. In order to incorporate the rotational dynamics, the state is augmented with the angular momentum tracking error $\delta\mathbf{h}$, defined as

$$\delta\mathbf{h}_k = \delta\mathbf{C}_k\mathbf{h}_{b_k}^{\mathcal{B}z/a} - \mathbf{h}_{r_k}^{\mathcal{B}z_r/a}, \tag{3.13}$$

where $\mathbf{h}_r^{\mathcal{B}z_r} = \mathbf{J}_r^{\mathcal{B}z_r}\boldsymbol{\omega}_r^{ra}$ is the reference angular momentum trajectory, and $\mathbf{h}_b^{\mathcal{B}z} = \mathbf{J}_b^{\mathcal{B}z}\boldsymbol{\omega}_b^{ba}$ is the true angular momentum. Note that the angular momentum error, $\delta\mathbf{h}$, is used in the state instead of the angular velocity error, $\delta\boldsymbol{\omega}$, because the Jacobians of the resulting linearized system are simpler. The augmented state $\delta\mathbf{x}_k$ is therefore

$$\delta\mathbf{x}_k = \begin{bmatrix} \delta\boldsymbol{\xi}_k^{\phi} \\ \delta\boldsymbol{\xi}_k^{v} \\ \delta\boldsymbol{\xi}_k^{r} \\ \delta\mathbf{h}_k \end{bmatrix} \in \mathbb{R}^{15}. \tag{3.14}$$

By including the rotational dynamics in the system, the angular velocity is now part of the state, and the rotor torque, $\mathbf{m}_b$ is now a control input. The control objective is to drive the tracking error to zero such that $\delta\mathbf{x}_k = \mathbf{0}$.

3.4 Control Inputs

Unlike [20], where a lower level inner-loop controller is used to generate torque commands and track the desired angular velocity, the proposed controller outputs torque commands directly. This allows input constraints to be enforced at the torque command level. Therefore the control inputs are the total thrust force, f, and the total rotor torque, $\mathbf{m}_b$,

$$\mathbf{u}_k = \begin{bmatrix} f_k \\ \mathbf{m}_{b_k} \end{bmatrix} \in \mathbb{R}^4. \tag{3.15}$$

Denote the reference thrust input trajectory f^r and the reference torque input $\mathbf{m}_r^r$. The reference control input vector is then

$$\mathbf{u}_k^r = \begin{bmatrix} f_k^r \\ \mathbf{m}_{r_k}^r \end{bmatrix} \in \mathbb{R}^4. \tag{3.16}$$

The input error is defined as

$$\delta\mathbf{u}_k = \begin{bmatrix} \delta f_k \\ \delta\mathbf{m}_k \end{bmatrix} = \begin{bmatrix} f_k - f_k^r \\ \mathbf{m}_{b_k} - \delta\mathbf{C}_k^{\mathsf{T}}\mathbf{m}_{r_k}^r \end{bmatrix}. \tag{3.17}$$

Note that the reference torque is resolved in $\mathcal{F}_r$ and must be multiplied by $\delta\mathbf{C}_k^{\mathsf{T}}$ to resolve it in $\mathcal{F}_b$.

3.5 Linearization of Dynamics

Consider the following first-order approximations, valid for small $\delta\boldsymbol{\xi}$ [21],

$$\delta\mathbf{C} \approx \mathbf{1} + \delta\boldsymbol{\xi}^{\phi^\times}, \tag{3.18a}$$

$$\delta\mathbf{C}^{\mathsf{T}} \approx \mathbf{1} - \delta\boldsymbol{\xi}^{\phi^\times}, \tag{3.18b}$$

$$\mathbf{J}\left(\delta\boldsymbol{\xi}^{\phi}\right) \approx \mathbf{1} + \tfrac{1}{2}\delta\boldsymbol{\xi}^{\phi^\times}. \tag{3.18c}$$

Using (3.18), and the error definitions from (3.8), (3.13), and (3.17), the continuous-time equations of motion can be linearized about the reference trajectory. The linearization procedure for each element of $\delta\mathbf{X}_k$ is presented next.

3.5.1 Attitude Error Dynamics

Starting with the attitude error dynamics, the time rate of change of the attitude error is found by differentiating (3.9) as

$$\begin{aligned}
\delta\dot{\mathbf{C}} &= \dot{\mathbf{C}}_{ar}^{\mathsf{T}}\mathbf{C}_{ab} + \mathbf{C}_{ar}^{\mathsf{T}}\dot{\mathbf{C}}_{ab} \qquad (3.19)\\
&= \left(\mathbf{C}_{ar}\boldsymbol{\omega}_r^{ra^\times}\right)^{\mathsf{T}}\mathbf{C}_{ab} + \mathbf{C}_{ar}^{\mathsf{T}}\mathbf{C}_{ab}\boldsymbol{\omega}_b^{ba^\times} \qquad (3.20)\\
&= -\boldsymbol{\omega}_r^{ra^\times}\mathbf{C}_{ar}^{\mathsf{T}}\mathbf{C}_{ab} + \mathbf{C}_{ar}^{\mathsf{T}}\mathbf{C}_{ab}\boldsymbol{\omega}_b^{ba^\times} \qquad (3.21)\\
&= -\boldsymbol{\omega}_r^{ra^\times}\delta\mathbf{C} + \delta\mathbf{C}\boldsymbol{\omega}_b^{ba^\times}. \qquad (3.22)
\end{aligned}$$

The angular velocity, $\boldsymbol{\omega}_b^{ba}$, can be written as

$$\begin{aligned}
\boldsymbol{\omega}_b^{ba} &= \mathbf{J}_b^{\mathcal{B}z^{-1}}\mathbf{h}_b^{\mathcal{B}z/a} \qquad (3.23)\\
&= \left(\mathbf{C}_{br}\mathbf{J}_r^{\mathcal{B}z_r}\mathbf{C}_{rb}\right)^{-1}\delta\mathbf{C}^{\mathsf{T}}\left(\delta\mathbf{h} + \mathbf{h}_r^{\mathcal{B}z_r/a}\right) \qquad (3.24)\\
&= \left(\delta\mathbf{C}^{\mathsf{T}}\mathbf{J}_r^{\mathcal{B}z_r}\delta\mathbf{C}\right)^{-1}\delta\mathbf{C}^{\mathsf{T}}\left(\delta\mathbf{h} + \mathbf{h}_r^{\mathcal{B}z_r/a}\right) \qquad (3.25)\\
&= \delta\mathbf{C}^{\mathsf{T}}\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\left(\delta\mathbf{h} + \mathbf{h}_r^{\mathcal{B}z_r/a}\right) \qquad (3.26)\\
&= \delta\mathbf{C}^{\mathsf{T}}\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\delta\mathbf{h} + \delta\mathbf{C}^{\mathsf{T}}\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\mathbf{h}_r^{\mathcal{B}z_r/a} \qquad (3.27)\\
&= \delta\mathbf{C}^{\mathsf{T}}\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\delta\mathbf{h} + \delta\mathbf{C}^{\mathsf{T}}\boldsymbol{\omega}_r^{ra}. \qquad (3.28)
\end{aligned}$$

Substituting (3.28) into (3.22),

$$\delta\dot{\mathbf{C}} = -\boldsymbol{\omega}_r^{ra^\times}\delta\mathbf{C} + \delta\mathbf{C}\left(\delta\mathbf{C}^{\mathsf{T}}\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\delta\mathbf{h} + \delta\mathbf{C}^{\mathsf{T}}\boldsymbol{\omega}_r^{ra}\right)^{\times}. \qquad (3.29)$$

Linearizing (3.29) using (3.18) and dropping higher-order terms,

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{1} + \delta\boldsymbol{\xi}^{\phi^\times}\right) &= -\boldsymbol{\omega}_r^{ra^\times}\left(\mathbf{1} + \delta\boldsymbol{\xi}^{\phi^\times}\right) + \left(\mathbf{1} + \delta\boldsymbol{\xi}^{\phi^\times}\right)\Bigg(\left(\mathbf{1} - \delta\boldsymbol{\xi}^{\phi^\times}\right)\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\delta\mathbf{h}\\
&\qquad + \left(\mathbf{1} - \delta\boldsymbol{\xi}^{\phi^\times}\right)\boldsymbol{\omega}_r^{ra}\Bigg)^{\times} \qquad (3.30)\\
\delta\dot{\boldsymbol{\xi}}^{\phi^\times} &= -\boldsymbol{\omega}_r^{ra^\times} - \boldsymbol{\omega}_r^{ra^\times}\delta\boldsymbol{\xi}^{\phi^\times} + \left(\mathbf{1} + \delta\boldsymbol{\xi}^{\phi^\times}\right)\left(\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\delta\mathbf{h} + \boldsymbol{\omega}_r^{ra} - \delta\boldsymbol{\xi}^{\phi^\times}\boldsymbol{\omega}_r^{ra}\right)^{\times} \qquad (3.31)\\
&= -\boldsymbol{\omega}_r^{ra^\times} - \boldsymbol{\omega}_r^{ra^\times}\delta\boldsymbol{\xi}^{\phi^\times} + \left(\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\delta\mathbf{h} + \boldsymbol{\omega}_r^{ra} - \delta\boldsymbol{\xi}^{\phi^\times}\boldsymbol{\omega}_r^{ra}\right)^{\times} + \delta\boldsymbol{\xi}^{\phi^\times}\boldsymbol{\omega}_r^{ra^\times} \qquad (3.32)\\
&= -\boldsymbol{\omega}_r^{ra^\times}\delta\boldsymbol{\xi}^{\phi^\times} + \left(\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\delta\mathbf{h}\right)^{\times} - \left(\delta\boldsymbol{\xi}^{\phi^\times}\boldsymbol{\omega}_r^{ra}\right)^{\times} + \delta\boldsymbol{\xi}^{\phi^\times}\boldsymbol{\omega}_r^{ra^\times}. \qquad (3.33)
\end{aligned}$$

Making use of (2.24),

$$\begin{aligned}\delta\dot{\boldsymbol{\xi}}^{\phi^\times} &= \left(\delta\boldsymbol{\xi}^{\phi^\times}\boldsymbol{\omega}_r^{ra}\right)^\times - \left(\delta\boldsymbol{\xi}^{\phi^\times}\boldsymbol{\omega}_r^{ra}\right)^\times + \left(\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\delta\mathbf{h}\right)^\times && (3.34)\\ &= \left(\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\delta\mathbf{h}\right)^\times. && (3.35)\end{aligned}$$

Uncrossing both sides,

$$\delta\dot{\boldsymbol{\xi}}^{\phi} = \mathbf{J}_r^{\mathcal{B}z_r^{-1}}\delta\mathbf{h}. \tag{3.36}$$

3.5.2 Velocity Error Dynamics

Next, the velocity error dynamics are linearized. The time rate of change of the velocity error is found by differentiating (3.10) as

$$\begin{aligned}\delta\dot{\mathbf{v}} &= \dot{\mathbf{C}}_{ar}^{\mathsf{T}}\left(\mathbf{v}^{zw/a} - \mathbf{v}^{z_r w/a}\right) + \mathbf{C}_{ar}^{\mathsf{T}}\left(\dot{\mathbf{v}}^{zw/a} - \dot{\mathbf{v}}^{z_r w/a}\right) && (3.37)\\ &= -\boldsymbol{\omega}_r^{ra^\times}\mathbf{C}_{ar}^{\mathsf{T}}\left(\mathbf{v}^{zw/a} - \mathbf{v}^{z_r w/a}\right) + \mathbf{C}_{ar}^{\mathsf{T}}\Bigg(\mathbf{g}_a - \frac{\mathbf{C}_{ab}f\mathbf{1}_3}{m_B} - \frac{1}{m_{\mathcal{B}}}\mathbf{C}_{ab}\mathbf{D}\mathbf{C}_{ab}^{\mathsf{T}}\mathbf{v}_a^{zw/a} \\ &\qquad - \mathbf{g}_a + \frac{\mathbf{C}_{ar}f^r\mathbf{1}_3}{m_B} + \frac{1}{m_{\mathcal{B}}}\mathbf{C}_{ar}\mathbf{D}\mathbf{C}_{ar}^{\mathsf{T}}\mathbf{v}_a^{z_r w/a}\Bigg) && (3.38)\\ &= -\boldsymbol{\omega}_r^{ra^\times}\delta\mathbf{v} + \mathbf{C}_{ar}^{\mathsf{T}}\Bigg(- \frac{\mathbf{C}_{ab}f\mathbf{1}_3}{m_B} + \frac{\mathbf{C}_{ar}f^r\mathbf{1}_3}{m_B} - \frac{1}{m_{\mathcal{B}}}\mathbf{C}_{ab}\mathbf{D}\mathbf{C}_{ab}^{\mathsf{T}}\mathbf{v}_a^{zw/a} \\ &\qquad + \frac{1}{m_{\mathcal{B}}}\mathbf{C}_{ar}\mathbf{D}\mathbf{C}_{ar}^{\mathsf{T}}\mathbf{v}_a^{z_r w/a}\Bigg) && (3.39)\\ &= -\boldsymbol{\omega}_r^{ra^\times}\delta\mathbf{v} + \frac{1}{m_B}\left(-\delta\mathbf{C}f\mathbf{1}_3 + f^r\mathbf{1}_3 - \delta\mathbf{C}\mathbf{D}\mathbf{C}_{ab}^{\mathsf{T}}\mathbf{v}_a^{zw/a} + \mathbf{D}\mathbf{C}_{ar}^{\mathsf{T}}\mathbf{v}_a^{z_r w/a}\right). && (3.40)\end{aligned}$$

Substituting using (3.9), (3.10), (3.12) and (3.17),

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{J}(\delta\boldsymbol{\xi}^{\phi})\delta\boldsymbol{\xi}^{v}\right) = -\boldsymbol{\omega}_r^{ra^\times}\mathbf{J}(\delta\boldsymbol{\xi}^{\phi})\delta\boldsymbol{\xi}^{v} + \frac{1}{m_B}\Bigg(&- \delta\mathbf{C}\left(f^r + \delta f\right)\mathbf{1}_3 + f^r\mathbf{1}_3 \\ &- \delta\mathbf{C}\mathbf{D}\delta\mathbf{C}^{\mathsf{T}}\mathbf{C}_{ar}^{\mathsf{T}}\left(\mathbf{C}_{ar}\mathbf{J}\left(\delta\boldsymbol{\xi}^{\phi}\right)\delta\boldsymbol{\xi}^{v} + \mathbf{v}_a^{z_r w/a}\right) + \mathbf{D}\mathbf{C}_{ar}^{\mathsf{T}}\mathbf{v}_a^{z_r w/a}\Bigg).\end{aligned} \tag{3.41}$$

Linearizing using (3.18) and dropping higher order terms,

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left(\left(\mathbf{1}+\tfrac{1}{2}\delta\boldsymbol{\xi}^{\phi^\times}\right)\delta\boldsymbol{\xi}^v\right) &= -\boldsymbol{\omega}_r^{ra^\times}\left(\mathbf{1}+\tfrac{1}{2}\delta\boldsymbol{\xi}^{\phi^\times}\right)\delta\boldsymbol{\xi}^v+\frac{1}{m_B}\Bigg(-\left(\mathbf{1}+\delta\boldsymbol{\xi}^{\phi^\times}\right)\Big(f^r \\ &\quad +\delta f\Big)\mathbf{1}_3+f^r\mathbf{1}_3-\left(\mathbf{1}+\delta\boldsymbol{\xi}^{\phi^\times}\right)\mathbf{D}\Big(\mathbf{1} \\ &\quad -\delta\boldsymbol{\xi}^{\phi^\times}\Big)\mathbf{C}_{ar}^\mathsf{T}\Big(\mathbf{C}_{ar}\Big(\mathbf{1}+\tfrac{1}{2}\delta\boldsymbol{\xi}^{\phi^\times}\Big)\delta\boldsymbol{\xi}^v+\mathbf{v}_a^{z_rw/a}\Big) \\ &\quad +\mathbf{D}\mathbf{C}_{ar}^\mathsf{T}\mathbf{v}_a^{z_rw/a}\Bigg)\end{aligned} \tag{3.42}$$

$$\begin{aligned}\delta\dot{\boldsymbol{\xi}}^v &= -\boldsymbol{\omega}_r^{ra^\times}\delta\boldsymbol{\xi}^v+\frac{1}{m_B}\Bigg(-\delta f\mathbf{1}_3-\delta\boldsymbol{\xi}^{\phi^\times}f^r\mathbf{1}_3-\mathbf{D}\delta\boldsymbol{\xi}^v \\ &\quad +\mathbf{D}\delta\boldsymbol{\xi}^{\phi^\times}\mathbf{C}_{ar}^\mathsf{T}\mathbf{v}_a^{z_rw/a}-\delta\boldsymbol{\xi}^{\phi^\times}\mathbf{D}\mathbf{C}_{ar}^\mathsf{T}\mathbf{v}_a^{z_rw/a}\Bigg).\end{aligned} \tag{3.43}$$

Using (2.23),

$$\begin{aligned}\delta\dot{\boldsymbol{\xi}}^v &= -\boldsymbol{\omega}_r^{ra^\times}\delta\boldsymbol{\xi}^v+\frac{1}{m_\mathcal{B}}\Bigg(\left(f^r\mathbf{1}_3\right)^\times\delta\boldsymbol{\xi}^\phi-\mathbf{D}\left(\mathbf{C}_{ar}^\mathsf{T}\mathbf{v}_a^{z_rw/a}\right)^\times\delta\boldsymbol{\xi}^\phi \\ &\quad +\left(\mathbf{D}\mathbf{C}_{ar}^\mathsf{T}\mathbf{v}_a^{z_rw/a}\right)^\times\delta\boldsymbol{\xi}^\phi-\mathbf{D}\delta\boldsymbol{\xi}^v-\mathbf{1}_3\delta f\Bigg)\end{aligned} \tag{3.44}$$

$$\begin{aligned}&= -\left(\frac{1}{m_\mathcal{B}}\mathbf{D}+\boldsymbol{\omega}_r^{ra^\times}\right)\delta\boldsymbol{\xi}^v+\frac{1}{m_\mathcal{B}}\Bigg(\left(\mathbf{D}\mathbf{C}_{ar}^\mathsf{T}\mathbf{v}_a^{z_rw/a}\right)^\times-\mathbf{D}\left(\mathbf{C}_{ar}^\mathsf{T}\mathbf{v}_a^{z_rw/a}\right)^\times \\ &\quad +\left(f^r\mathbf{1}_3\right)^\times\Bigg)\delta\boldsymbol{\xi}^\phi-\frac{1}{m_\mathcal{B}}\mathbf{1}_3\delta f.\end{aligned} \tag{3.45}$$

3.5.3 Position Error Dynamics

Next, the position error dynamics are linearized. The time rate of change of the position error is found by differentiating (3.11) as

$$\delta\dot{\mathbf{r}} = \dot{\mathbf{C}}_{ar}^\mathsf{T}\left(\mathbf{r}^{z_rw}-\mathbf{r}^{zw}\right)+\mathbf{C}_{ar}^\mathsf{T}\left(\dot{\mathbf{r}}^{z_rw}-\dot{\mathbf{r}}^{zw}\right) \tag{3.46}$$

$$= -\boldsymbol{\omega}_r^{ra^\times}\mathbf{C}_{ar}^\mathsf{T}\left(\mathbf{r}^{z_rw}-\mathbf{r}^{zw}\right)+\delta\mathbf{v} \tag{3.47}$$

$$= -\boldsymbol{\omega}_r^{ra^\times}\delta\mathbf{r}+\delta\mathbf{v}. \tag{3.48}$$

Substituting using (3.12) and (3.17),

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{J}(\delta\boldsymbol{\xi}^{\phi})\delta\boldsymbol{\xi}^{r}\right) = -\boldsymbol{\omega}_r^{ra^\times}\mathbf{J}(\delta\boldsymbol{\xi}^{\phi})\delta\boldsymbol{\xi}^{r} + \mathbf{J}(\delta\boldsymbol{\xi}^{\phi})\delta\boldsymbol{\xi}^{v}. \tag{3.49}$$

Linearizing using (3.18) and dropping higher order terms,

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\left(\left(\mathbf{1} + \tfrac{1}{2}\delta\boldsymbol{\xi}^{\phi^\times}\right)\delta\boldsymbol{\xi}^{r}\right) &= -\boldsymbol{\omega}_r^{ra^\times}\left(\mathbf{1} + \tfrac{1}{2}\delta\boldsymbol{\xi}^{\phi^\times}\right)\delta\boldsymbol{\xi}^{r} + \left(\mathbf{1} + \tfrac{1}{2}\delta\boldsymbol{\xi}^{\phi^\times}\right)\delta\boldsymbol{\xi}^{v} && (3.50)\\ \delta\dot{\boldsymbol{\xi}}^{r} &= -\boldsymbol{\omega}_r^{ra^\times}\delta\boldsymbol{\xi}^{r} + \delta\boldsymbol{\xi}^{v}. && (3.51) \end{aligned}$$

3.5.4 Angular Momentum Error Dynamics

Lastly, the angular momentum error dynamics are linearized. The time rate of change of the angular momentum error is found by differentiating (3.13) as

$$\begin{aligned} \delta\dot{\mathbf{h}} &= \delta\dot{\mathbf{C}}\mathbf{h}_b^{\mathcal{B}z/a} + \delta\mathbf{C}\dot{\mathbf{h}}_b^{\mathcal{B}z/a} - \dot{\mathbf{h}}_r^{\mathcal{B}z_r/a} && (3.52)\\ &= \delta\dot{\mathbf{C}}\mathbf{h}_b^{\mathcal{B}z/a} + \delta\mathbf{C}\left(\mathbf{m}_b - \mathbf{E}\mathbf{C}_{ab}^{\mathsf{T}}\mathbf{v}_a^{zw/a} - \mathbf{F}\boldsymbol{\omega}_b^{ba} - \boldsymbol{\omega}_b^{ba^\times}\mathbf{h}_b^{\mathcal{B}z/a}\right) \\ &\quad - \left(\mathbf{m}_r^{r} - \mathbf{E}\mathbf{C}_{ar}^{\mathsf{T}}\mathbf{v}_a^{z_r w/a} - \mathbf{F}\boldsymbol{\omega}_r^{ra} - \boldsymbol{\omega}_r^{ra^\times}\mathbf{h}_r^{\mathcal{B}z_r/a}\right). && (3.53) \end{aligned}$$

Substituting using (3.12), (3.13), (3.17), and (3.28),

$$\begin{aligned} \delta\dot{\mathbf{h}} = \delta\dot{\mathbf{C}}\delta\mathbf{C}^{\mathsf{T}}&\left(\delta\mathbf{h} + \mathbf{h}_r^{\mathcal{B}z_r/a}\right) + \delta\mathbf{C}\Bigg(\delta\mathbf{m} + \delta\mathbf{C}^{\mathsf{T}}\mathbf{m}_r^{r} \\ &- \mathbf{E}\delta\mathbf{C}^{\mathsf{T}}\mathbf{C}_{ar}^{\mathsf{T}}\left(\mathbf{C}_{ar}\mathbf{J}\left(\delta\boldsymbol{\xi}^{\phi}\right)\delta\boldsymbol{\xi}^{v} + \mathbf{v}_a^{z_r w/a}\right) - \mathbf{F}\left(\delta\mathbf{C}^{\mathsf{T}}\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\delta\mathbf{h} + \delta\mathbf{C}^{\mathsf{T}}\boldsymbol{\omega}_r^{ra}\right) \\ &- \left(\delta\mathbf{C}^{\mathsf{T}}\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\delta\mathbf{h} + \delta\mathbf{C}^{\mathsf{T}}\boldsymbol{\omega}_r^{ra}\right)^{\times}\delta\mathbf{C}^{\mathsf{T}}\left(\delta\mathbf{h} + \mathbf{h}_r^{\mathcal{B}z_r/a}\right)\Bigg) && (3.54)\\ &- \mathbf{m}_r^{r} + \mathbf{E}\mathbf{C}_{ar}^{\mathsf{T}}\mathbf{v}_a^{z_r w/a} + \mathbf{F}\boldsymbol{\omega}_r^{ra} + \boldsymbol{\omega}_r^{ra^\times}\mathbf{h}_r^{\mathcal{B}z_r/a} \\ = \delta\dot{\mathbf{C}}\delta\mathbf{C}^{\mathsf{T}}&\left(\delta\mathbf{h} + \mathbf{h}_r^{\mathcal{B}z_r/a}\right) + \delta\mathbf{C}\delta\mathbf{m} - \delta\mathbf{C}\mathbf{E}\delta\mathbf{C}^{\mathsf{T}}\mathbf{C}_{ar}^{\mathsf{T}}\left(\mathbf{C}_{ar}\mathbf{J}\left(\delta\boldsymbol{\xi}^{\phi}\right)\delta\boldsymbol{\xi}^{v} + \mathbf{v}_a^{z_r w/a}\right) \\ &- \delta\mathbf{C}\mathbf{F}\left(\delta\mathbf{C}^{\mathsf{T}}\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\delta\mathbf{h} + \delta\mathbf{C}^{\mathsf{T}}\boldsymbol{\omega}_r^{ra}\right) \\ &- \delta\mathbf{C}\left(\delta\mathbf{C}^{\mathsf{T}}\mathbf{J}_r^{\mathcal{B}z^{-1}}\delta\mathbf{h} + \delta\mathbf{C}^{\mathsf{T}}\boldsymbol{\omega}_r^{ra}\right)^{\times}\delta\mathbf{C}^{\mathsf{T}}\left(\delta\mathbf{h} + \mathbf{h}_r^{\mathcal{B}z_r/a}\right) && (3.55)\\ &+ \mathbf{E}\mathbf{C}_{ar}^{\mathsf{T}}\mathbf{v}_a^{z_r w/a} + \mathbf{F}\boldsymbol{\omega}_r^{ra} + \boldsymbol{\omega}_r^{ra^\times}\mathbf{h}_r^{\mathcal{B}z_r/a}. \end{aligned}$$

Linearizing using (3.18) and dropping higher order terms,

$$
\begin{aligned}
\delta\dot{\mathbf{h}} = {} & \delta\dot{\boldsymbol{\xi}}^{\phi^\times}\left(\mathbf{1}-\delta\boldsymbol{\xi}^{\phi^\times}\right)\left(\delta\mathbf{h}+\mathbf{h}_r^{\mathcal{B}z_r/a}\right)+\left(\mathbf{1}+\delta\boldsymbol{\xi}^{\phi^\times}\right)\delta\mathbf{m} \\
& -\left(\mathbf{1}+\delta\boldsymbol{\xi}^{\phi^\times}\right)\mathbf{E}\left(\mathbf{1}-\delta\boldsymbol{\xi}^{\phi^\times}\right)\mathbf{C}_{ar}^{\mathsf{T}}\left(\mathbf{C}_{ar}\left(\mathbf{1}+\tfrac{1}{2}\delta\boldsymbol{\xi}^{\phi^\times}\right)\delta\boldsymbol{\xi}^{v}+\mathbf{v}_a^{z_rw/a}\right) \\
& -\left(\mathbf{1}+\delta\boldsymbol{\xi}^{\phi^\times}\right)\mathbf{F}\left(\left(\mathbf{1}-\delta\boldsymbol{\xi}^{\phi^\times}\right)\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\delta\mathbf{h}+\left(\mathbf{1}-\delta\boldsymbol{\xi}^{\phi^\times}\right)\boldsymbol{\omega}_r^{ra}\right) \\
& -\left(\mathbf{1}+\delta\boldsymbol{\xi}^{\phi^\times}\right)\Bigg(\left(\mathbf{1}-\delta\boldsymbol{\xi}^{\phi^\times}\right)\mathbf{J}_r^{\mathcal{B}z^{-1}}\delta\mathbf{h} \\
& \qquad +\left(\mathbf{1}-\delta\boldsymbol{\xi}^{\phi^\times}\right)\boldsymbol{\omega}_r^{ra}\Bigg)^{\times}\left(\mathbf{1}-\delta\boldsymbol{\xi}^{\phi^\times}\right)\left(\delta\mathbf{h}+\mathbf{h}_r^{\mathcal{B}z_r/a}\right) \\
& +\mathbf{E}\mathbf{C}_{ar}^{\mathsf{T}}\mathbf{v}_a^{z_rw/a}+\mathbf{F}\boldsymbol{\omega}_r^{ra}+\boldsymbol{\omega}_r^{ra^\times}\mathbf{h}_r^{\mathcal{B}z_r/a}
\end{aligned}
\tag{3.56}
$$

$$
\begin{aligned}
= {} & \delta\dot{\boldsymbol{\xi}}^{\phi^\times}\mathbf{h}_r^{\mathcal{B}z_r/a}+\delta\mathbf{m}-\mathbf{E}\delta\boldsymbol{\xi}^{v}+\mathbf{E}\delta\boldsymbol{\xi}^{\phi^\times}\mathbf{C}_{ar}^{\mathsf{T}}\mathbf{v}_a^{z_rw/a}-\mathbf{F}\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\delta\mathbf{h}+\mathbf{F}\delta\boldsymbol{\xi}^{\phi^\times}\boldsymbol{\omega}_r^{ra} \\
& -\left(\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\delta\mathbf{h}\right)^{\times}\mathbf{h}_r^{\mathcal{B}z_r/a}-\boldsymbol{\omega}_r^{ra^\times}\delta\mathbf{h}+\boldsymbol{\omega}_r^{ra^\times}\delta\boldsymbol{\xi}^{\phi^\times}\mathbf{h}_a^{\mathcal{B}z_r/a}+\left(\delta\boldsymbol{\xi}^{\phi^\times}\boldsymbol{\omega}_r^{ra}\right)^{\times}\mathbf{h}_a^{\mathcal{B}z_r/a} \\
& -\delta\boldsymbol{\xi}^{\phi^\times}\mathbf{E}\mathbf{C}_{ar}^{\mathsf{T}}\mathbf{v}_a^{z_rw/a}-\delta\boldsymbol{\xi}^{\phi^\times}\mathbf{F}\boldsymbol{\omega}_r^{ra}-\delta\boldsymbol{\xi}^{\phi^\times}\boldsymbol{\omega}_r^{ra^\times}\mathbf{h}_a^{\mathcal{B}z_r/a}.
\end{aligned}
\tag{3.57}
$$

Substituting in the previously derived expression for $\delta\dot{\boldsymbol{\xi}}^{\phi}$ from (3.36) and using (2.24),

$$
\begin{aligned}
\delta\dot{\mathbf{h}} = {} & \delta\mathbf{m}-\mathbf{E}\delta\boldsymbol{\xi}^{v}+\mathbf{E}\delta\boldsymbol{\xi}^{\phi^\times}\mathbf{C}_{ar}^{\mathsf{T}}\mathbf{v}_a^{z_rw/a}-\mathbf{F}\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\delta\mathbf{h}+\mathbf{F}\delta\boldsymbol{\xi}^{\phi^\times}\boldsymbol{\omega}_r^{ra} \\
& -\boldsymbol{\omega}_r^{ra^\times}\delta\mathbf{h}-\delta\boldsymbol{\xi}^{\phi^\times}\mathbf{E}\mathbf{C}_{ar}^{\mathsf{T}}\mathbf{v}_a^{z_rw/a}-\delta\boldsymbol{\xi}^{\phi^\times}\mathbf{F}\boldsymbol{\omega}_r^{ra}
\end{aligned}
\tag{3.58}
$$

$$
\begin{aligned}
= {} & \left(\left(\mathbf{E}\mathbf{C}_{ar}^{\mathsf{T}}\mathbf{v}_a^{z_rw/a}\right)^{\times}-\mathbf{E}\left(\mathbf{C}_{ar}^{\mathsf{T}}\mathbf{v}_a^{z_rw/a}\right)^{\times}+\left(\mathbf{F}\boldsymbol{\omega}_r^{ra}\right)^{\times}-\mathbf{F}\boldsymbol{\omega}_r^{ra^\times}\right)\delta\boldsymbol{\xi}^{\phi} \\
& -\mathbf{E}\delta\boldsymbol{\xi}^{v}+\left(-\boldsymbol{\omega}_r^{ra^\times}-\mathbf{F}\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\right)\delta\mathbf{h}+\delta\mathbf{m}.
\end{aligned}
\tag{3.59}
$$

Combining (3.36), (3.45), (3.51), and (3.59), the continuous-time, linearized error dynamics can be written in state-space form as

$$
\underbrace{\begin{bmatrix}\delta\dot{\boldsymbol{\xi}}^{\phi}\\ \delta\dot{\boldsymbol{\xi}}^{v}\\ \delta\dot{\boldsymbol{\xi}}^{r}\\ \delta\dot{\mathbf{h}}\end{bmatrix}}_{\delta\dot{\mathbf{x}}}
=
\underbrace{\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{J}_r^{\mathcal{B}z_r^{-1}}\\ \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{1} & -\boldsymbol{\omega}_r^{ra^\times} & \mathbf{0}\\ \mathbf{A}_{41} & -\mathbf{E} & \mathbf{0} & \mathbf{A}_{44}\end{bmatrix}}_{\mathbf{A}}
\underbrace{\begin{bmatrix}\delta\boldsymbol{\xi}^{\phi}\\ \delta\boldsymbol{\xi}^{v}\\ \delta\boldsymbol{\xi}^{r}\\ \delta\mathbf{h}\end{bmatrix}}_{\delta\mathbf{x}}
+
\underbrace{\begin{bmatrix}\mathbf{0} & \mathbf{0}\\ -1/m_{\mathcal{B}}\mathbf{1}_3 & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{1}\end{bmatrix}}_{\mathbf{B}}
\underbrace{\begin{bmatrix}\delta f\\ \delta\mathbf{m}\end{bmatrix}}_{\delta\mathbf{u}}
\tag{3.60}
$$

where

$$\mathbf{A}_{21} = \frac{1}{m_{\mathcal{B}}}\left(\left(\mathbf{D}\mathbf{C}_{ar}^{\mathsf{T}}\mathbf{v}_a^{z_r w/a}\right)^{\times} - \mathbf{D}\left(\mathbf{C}_{ar}^{\mathsf{T}}\mathbf{v}_a^{z_r w/a}\right)^{\times} + (f^r\mathbf{1}_3)^{\times}\right), \tag{3.61}$$

$$\mathbf{A}_{22} = -\boldsymbol{\omega}_r^{ra^{\times}} - \frac{1}{m_{\mathcal{B}}}\mathbf{D}, \tag{3.62}$$

$$\mathbf{A}_{41} = \left(\mathbf{E}\mathbf{C}_{ar}^{\mathsf{T}}\mathbf{v}_a^{z_r w/a}\right)^{\times} - \mathbf{E}\left(\mathbf{C}_{ar}^{\mathsf{T}}\mathbf{v}_a^{z_r w/a}\right)^{\times} + (\mathbf{F}\boldsymbol{\omega}_r^{ra})^{\times} - \mathbf{F}\boldsymbol{\omega}_r^{ra^{\times}}, \tag{3.63}$$

$$\mathbf{A}_{44} = -\boldsymbol{\omega}_r^{ra^{\times}} - \mathbf{F}\mathbf{J}_r^{\mathcal{B}z_r^{-1}}. \tag{3.64}$$

If the drag terms **D**, **E**, and **F** are assumed to be zero, then the dependence on the reference attitude, $\mathbf{C}_{ar}$, disappears and the Jacobians only depend on the reference thrust, f^r, and reference angular velocity, $\boldsymbol{\omega}_r^{ra}$.

3.6 Augmenting the System With an Integrator

To reduce steady-state error due to unmodeled dynamics or constant disturbances, the linearized dynamics can be augmented with integral control on the position and velocity states. The integrator is of the form [22]

$$\boldsymbol{\xi}^i = \int_0^t (c_1\delta\mathbf{r} + c_2\delta\mathbf{v})\,\mathrm{d}\tau, \tag{3.65}$$

where $c_1, c_2 > 0$. The integrator is linearized using (3.12) and (3.18) as

$$\delta\dot{\boldsymbol{\xi}}^i = c_1\mathbf{J}(\delta\boldsymbol{\xi}^{\phi})\delta\boldsymbol{\xi}^r + c_2\mathbf{J}(\delta\boldsymbol{\xi}^{\phi})\delta\boldsymbol{\xi}^v \tag{3.66}$$

$$\approx c_1\delta\boldsymbol{\xi}^r + c_2\delta\boldsymbol{\xi}^v. \tag{3.67}$$

Therefore, the augmented system dynamics are

$$\delta\dot{\bar{\mathbf{x}}} = \bar{\mathbf{A}}\delta\bar{\mathbf{x}} + \bar{\mathbf{B}}\delta\mathbf{u}, \tag{3.68}$$

where the augmented state is

$$\delta\bar{\mathbf{x}} = \begin{bmatrix} \delta\mathbf{x} \\ \delta\boldsymbol{\xi}^i \end{bmatrix} \in \mathbb{R}^{18}, \tag{3.69}$$

and matrices $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ are given by

$$\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{J}_r^{\mathcal{B}z_r^{-1}} & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & -\boldsymbol{\omega}_r^{ra^\times} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{41} & -\mathbf{E} & \mathbf{0} & \mathbf{A}_{44} & \mathbf{0} \\ \mathbf{0} & c_2\mathbf{1} & c_1\mathbf{1} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \bar{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}. \tag{3.70}$$

3.7 Discretization

The continuous-time linearized system is discretized using the matrix exponential, as presented in [23] and [24]. First, the matrix $\boldsymbol{\Xi}$ is constructed using the linearized continuous-time $\mathbf{A}$ and $\mathbf{B}$ matrices from (3.60), or the augmented $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ matrices from (3.70),

$$\boldsymbol{\Xi} = \begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{A}^{\mathsf{T}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{3.71}$$

Next, the matrix $\boldsymbol{\Upsilon}$ is found by multiplying $\boldsymbol{\Xi}$ by the sampling period T and taking the matrix exponential,

$$\begin{aligned} \boldsymbol{\Upsilon} &= \exp(T\boldsymbol{\Xi}) && (3.72) \\ &= \begin{bmatrix} \boldsymbol{\Upsilon}_{11} & \star & \star & \star \\ \mathbf{0} & \star & \star & \star \\ \mathbf{0} & \mathbf{0} & \star & \boldsymbol{\Upsilon}_{34} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \star \end{bmatrix}. && (3.73) \end{aligned}$$

The discrete-time matrices $\mathbf{A}_k$ and $\mathbf{B}_k$ are then extracted from $\boldsymbol{\Upsilon}$ as,

$$\mathbf{A}_k = \boldsymbol{\Upsilon}_{11}, \tag{3.74}$$

$$\mathbf{B}_k = \boldsymbol{\Upsilon}_{34}, \tag{3.75}$$

and are used to create the discrete-time linearized system,

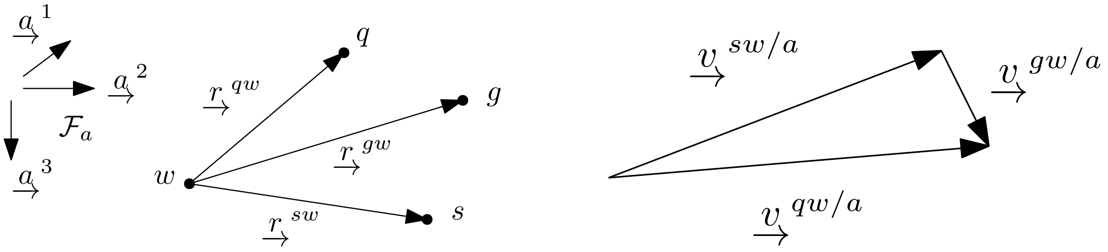
$$\delta\mathbf{x}_{k+1} = \mathbf{A}_k\delta\mathbf{x}_k + \mathbf{B}_k\delta\mathbf{u}_k. \tag{3.76}$$

3.8 Environmental Disturbance Modeling

Environmental disturbances are included in simulations through the use of wind modeling. The total wind field is modeled as a combination of constant ambient wind and stochastic gusts. Consider a particle s that moves with the ambient wind field, a particle g that moves with the wind gusts, and a particle q that moves with the total wind field, as shown in Figure 3.2a. The motion of q relative to w, with respect to $\mathcal{F}_a$ is given by

$$\underset{\rightarrow}{v}^{qw/a} = \underset{\rightarrow}{v}^{sw/a} + \underset{\rightarrow}{v}^{gw/a}. \tag{3.77}$$

This relationship is visualized in Figure 3.2b.



$\underset{\rightarrow}{a}^1$
$\underset{\rightarrow}{a}^2$
$\mathcal{F}_a$
$\underset{\rightarrow}{a}^3$
q
$\underset{\rightarrow}{r}^{qw}$
g
$\underset{\rightarrow}{r}^{gw}$
w
$\underset{\rightarrow}{r}^{sw}$
s
$\underset{\rightarrow}{v}^{sw/a}$
$\underset{\rightarrow}{v}^{gw/a}$
$\underset{\rightarrow}{v}^{qw/a}$

(a) Particle definitions.

(b) Velocity diagram.

Figure 3.2: Wind kinematic definitions.

The wind gusts are modeled using the Dryden spectral representation to add turbulence. The discrete-time Dryden model is given by [25]

$$v_{b_{1k}}^{gw/a} = \left(1 - \frac{V}{L_1}T\right) v_{b_{1k-1}}^{gw/a} + \sqrt{2\frac{V}{L_1}T}\sigma_1\eta_1, \tag{3.78}$$

$$v_{b_{2k}}^{gw/a} = \left(1 - \frac{V}{L_1}T\right) v_{b_{2k-1}}^{gw/a} + \sqrt{2\frac{V}{L_1}T}\sigma_1\eta_2, \tag{3.79}$$

$$v_{b_{3k}}^{gw/a} = \left(1 - \frac{V}{L_2}T\right) v_{b_{3k-1}}^{gw/a} + \sqrt{2\frac{V}{L_2}T}\sigma_2\eta_3, \tag{3.80}$$

where $\mathbf{v}_{b_k}^{gw/a} = \begin{bmatrix} v_{b_{1k}}^{gw/a} & v_{b_{2k}}^{gw/a} & v_{b_{3k}}^{gw/a} \end{bmatrix}^\mathsf{T}$ are the wind gust velocities resolved in the body frame at time k, V is the nominal aircraft velocity in ft/s, T is the timestep, L_1 and L_2 represent the turbulence scale lengths in ft, σ_1 and σ_2 represent the turbulence intensities, and $\eta_{1,2,3} \sim \mathcal{N}(0, 1)$. The low-altitude model for altitudes under 1000 ft is used to set the turbulence length scales and intensities. The turbulence length scales are

$$L_1 = \frac{h}{(0.177 + 0.000823h)^{1.2}}, \qquad L_2 = h, \tag{3.81}$$

where h is the altitude in feet. The turbulence intensities are

$$\sigma_1 = \frac{\sigma_2}{(0.177 + 0.000823h)^{0.4}}, \qquad \sigma_2 = 0.1W_{20}, \tag{3.82}$$

where W_{20} is the wind speed at 20 ft. Typical values for W_{20} are 25 ft/s for light turbulence and 75 ft/s for severe turbulence [26].

Next, the windspeed vector, $\underset{\rightarrow}{v}^{qw/a}$, and groundspeed vector, $\underset{\rightarrow}{v}^{zw/a}$, are used to calculate the airspeed vector, $\underset{\rightarrow}{v}^{zq/a}$, by the relationship

$$\underset{\rightarrow}{v}^{zq/a} = \underset{\rightarrow}{v}^{zw/a} - \underset{\rightarrow}{v}^{qw/a}, \tag{3.83}$$

and is visualized by the wind triangle in Figure 3.3.

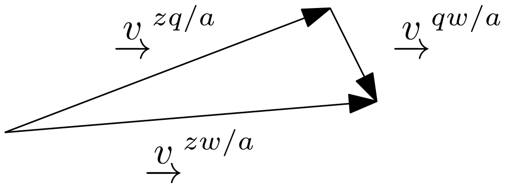

$\underset{\rightarrow}{v}^{zq/a}$
$\underset{\rightarrow}{v}^{qw/a}$
$\underset{\rightarrow}{v}^{zw/a}$

Figure 3.3: Wind triangle relationship.

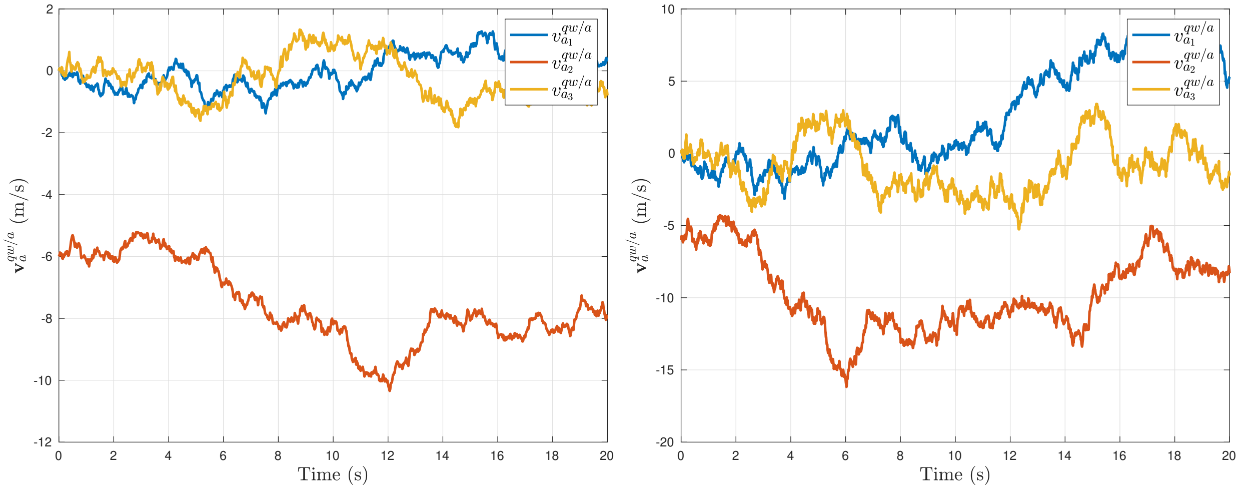
Lastly, the airspeed is used to calculate the external force due to drag. The components of the aerodynamic drag force acting on the vehicle, resolved in the body frame are given by [27]

$$f_{b_i}^{\text{ext}} = \tfrac{1}{2}\rho \left\| \mathbf{v}_b^{zq/a} \right\|^2 S_i C_{D_i}, \quad i = 1, 2, 3, \tag{3.84}$$

where ρ is the air density, $\mathbf{v}_b^{zq/a}$ is the airspeed vector resolved in $\mathcal{F}_b$, S_i is the projected area of the vehicle normal to the $\underset{\rightarrow}{b}^i$ direction, and C_{D_i} is the drag coefficient of the vehicle in the $\underset{\rightarrow}{b}^i$ direction. For simplicity, the vehicle is modeled as a simple cylinder to determine the drag coefficients.

3.8.1 Simulation Results

Simulated wind velocities over a 20 s period in light and heavy turbulence conditions are shown below in Figure 3.4. The nominal wind velocity is set to $\mathbf{v}_a^{sw/a} = \begin{bmatrix} 0 & -6 & 0 \end{bmatrix}^{\mathsf{T}}$ m·s^{-1}, and the gust velocity $\mathbf{v}_a^{gw/a}$ is generated using the Dryden model. These wind velocity plots are representative of the environmental disturbances applied to the vehicle when testing the controller performance in Chapters 5 and 6.



(a) $W_{20} = 25 \text{ ft} \cdot \text{s}^{-1}$.

(b) $W_{20} = 75 \text{ ft} \cdot \text{s}^{-1}$.

Figure 3.4: Simulated wind velocities generated by the Dryden model with low and high levels of turbulence.

Chapter 4

Guidance

In this chapter, the guidance strategy for the tandem-rotor is presented. The objective of the guidance is to use a trajectory planning algorithm to generate a reference trajectory of control inputs and states from the current location to some desired target location.

There are many methods of trajectory generation in the literature that try to strike a balance between computational complexity and the ability to handle aggressive motions in complex environments [28]. For the purposes of this research, the ability to run the trajectory generation algorithm in real-time on limited hardware is of high importance. Additionally, the proposed use case for the tandem-rotor helicopter does not include complex maneuvering. Therefore, simple and computationally lightweight algorithms are preferred.

Direct collocation is a relatively simple and common method of optimal trajectory generation that respects the system dynamics, path constraints, and boundary conditions [29]. However, this method requires solving a nonlinear program (NLP), which may not be feasible in a real-time environment. Additionally, direct collocation and other methods of trajectory generation require prior temporal knowledge about the desired trajectory, such as the duration of the desired trajectory [30]. Given the wide variety of initial conditions and environmental conditions that the tandem-rotor helicopter may operate in, this information may not always be known.

Another class of trajectory generation algorithms exist that exploit the differentially flat dynamics of the system to constrain the trajectory and an optimization problem is solved to minimize some quantity, such as the time [31], or a position derivative [32]. The simplest of these algorithms is a quartic polynomial method [33], which is leveraged in this thesis.

Because the resulting trajectory is coarse and doesn't respect boundary conditions for all states, a method of trajectory refinement is required. Linear quadratic regulator (LQR) has traditionally been used as a method of optimal control. In particular, LQR control on the matrix Lie group $SE_2(3)$ is successfully demonstrated in [20]. The solution to a discrete-time finite-horizon LQR problem is used in this thesis as a method of trajectory generation to refine the quartic reference trajectory and ensure all boundary conditions are satisfied.

4.1 Overview

The proposed two-stage guidance structure is shown in Figure 4.1.

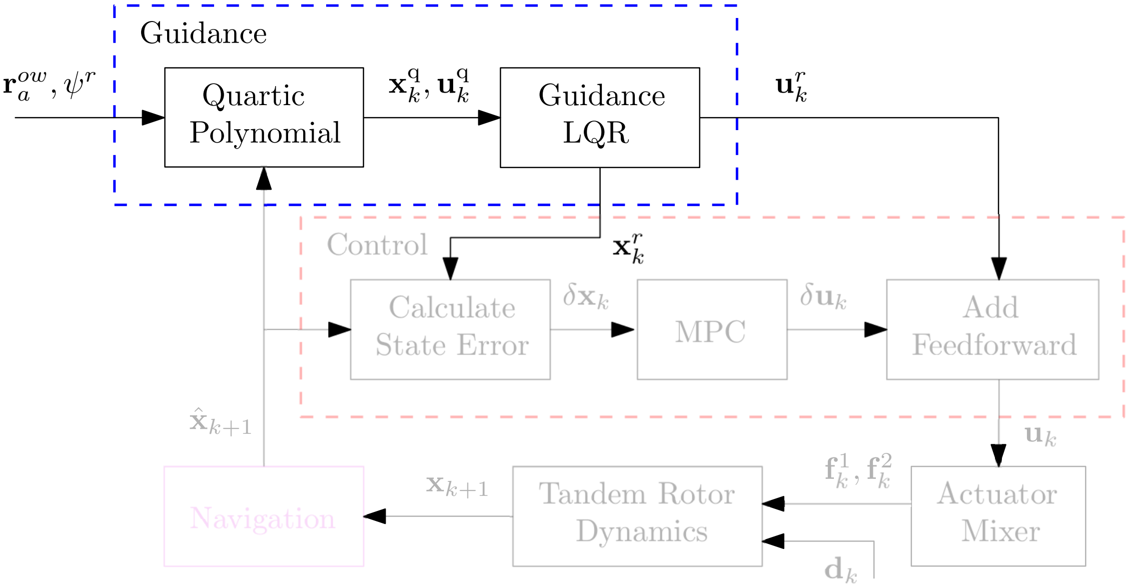


Figure 4.1: Proposed guidance structure.

The first stage uses simple polynomial functions to generate coarse minimum-time position and velocity trajectories from the current vehicle position to a target location. By simplifying the tandem-rotor control inputs to the total thrust and rotor torque, as shown in Section 3.4, the system can be shown to be differentially flat [19]. The differentially flat property of the tandem-rotor helicopter dynamics is leveraged to generate the remaining states and control inputs. To reduce the computational complexity, the trajectory is simplified to two dimensions, horizontal and vertical.

The second guidance stage refines and expands the coarse reference trajectory into a smooth three-dimensional reference trajectory. The linearized tandem-rotor dynamics derived in Section 3.5 are used along with the reference states and control inputs from the quartic trajectory to create a high-quality feedforward term for the controller.

Because the guidance algorithm is computationally simple, it can be recomputed online at any time. This could occur due to a large disturbance pushing the vehicle off the planned trajectory. Additionally, to reduce memory usage, the reference trajectory can be planned for a smaller fixed duration and replanned in real-time as necessary.

4.2 Quartic Trajectory Generation

A quartic guidance law is generated to provide a minimum-time position and velocity trajectory from the current position to a target location using the method from [33]. Because the flat outputs are the position and heading, the remaining state, $\mathbf{x}_k^q$, and control input trajectories, $\mathbf{u}_k^q$, can be found as a function of the quartic polynomials and a specified heading [19].

4.2.1 Quartic Polynomials

Consider an initial location denoted by the point z_0 and a target location denoted by point z_f. The track angle ψ^t from the initial location to the target is given by

$$\psi^t = \text{atan2}\left(r_{a_2}^{z_0 w} - r_{a_2}^{z_f w},\;\; r_{a_1}^{z_0 w} - r_{a_1}^{z_f w}\right) + \pi. \tag{4.1}$$

An intermediate frame $\mathcal{F}_q$ is defined such that $\underrightarrow{q}^1$ is aligned with ψ^t and $\underrightarrow{q}^3$ is aligned with $\underrightarrow{a}^3$. The DCM $\mathbf{C}_{qa}$ is defined using a $\mathbf{C}_3$ rotation

$$\mathbf{C}_{qa} = \begin{bmatrix} \cos\psi^t & \sin\phi^t & 0 \\ -\sin\psi^t & \cos\psi^t & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.2}$$

A coarse reference trajectory for position, velocity, acceleration, jerk, and snap is generated from the current position to the target along a single axis using a series of fourth-order polynomials as [33]

$$r_{q_i}^{z_q w} = r_{q_i}^{z_f w} + v_{q_i}^{z_f w/a} t + \frac{1}{2}\dot{v}_{q_i}^{z_f w/a} t^2 + \frac{1}{6}\ddot{v}_{q_i}^{z_f w/a} t^3 + \frac{1}{24}\dddot{v}_{q_i}^{z_f w/a} t^4, \tag{4.3}$$

$$v_{q_i}^{z_q w/a} = v_{q_i}^{z_f w/a} + \dot{v}_{q_i}^{z_f w/a} t + \frac{1}{2}\ddot{v}_{q_i}^{z_f w/a} t^2 + \frac{1}{6}\dddot{v}_{q_i}^{z_f w/a} t^3, \tag{4.4}$$

$$\dot{v}_{q_i}^{z_q w/a} = \dot{v}_{q_i}^{z_f w/a} + \ddot{v}_{q_i}^{z_f w/a} t + \frac{1}{2}\dddot{v}_{q_i}^{z_f w/a} t^2, \tag{4.5}$$

$$\ddot{v}_{q_i}^{z_q w/a} = \ddot{v}_{q_i}^{z_f w/a} + \dddot{v}_{q_i}^{z_f w/a} t, \tag{4.6}$$

$$\dddot{v}_{q_i}^{z_q w/a} = \dddot{v}_{q_i}^{z_f w/a}. \tag{4.7}$$

The values for the coefficients $r_{q_i}^{z_f w}$, $v_{q_i}^{z_f w/a}$, and $\dot{v}_{q_i}^{z_f w/a}$ are selected based on the desired state at the end of the trajectory. In the case of a landing trajectory,

$$r_{q_i}^{z_f w} = 0, \tag{4.8}$$

$$v_{q_i}^{z_f w/a} = 0, \tag{4.9}$$

$$\dot{v}_{q_i}^{z_f w/a} = 0, \qquad i = 1, 2, 3. \tag{4.10}$$

The values of $\ddot{v}_{q_i}^{z_f w/a}$ and $\dddot{v}_{q_i}^{z_f w/a}$ are chosen based on the current position and velocity of the helicopter to ensure the generated trajectory is smooth,

$$\begin{aligned} \ddot{v}_{q_i}^{z_f w/a} = &\frac{-24}{t_i^3}\left(\left(r_{q_i}^{z_f w} - r_{q_i}^{z_0 w}\right) + v_{q_i}^{z_f w/a} t_i + \frac{1}{2}\dot{v}_{q_i}^{z_f w/a} t_i^2\right) \\ &+ \frac{6}{t_i^2}\left(v_{q_i}^{z_f w/a} - v_{q_i}^{z_0 f w/a} + \dot{v}_{q_i}^{z_f w/a} t\right), \end{aligned} \tag{4.11}$$

$$\begin{aligned} \dddot{v}_{q_i}^{z_f w/a} = &\frac{72}{t_i^4}\left(\left(r_{q_i}^{z_f w} - r_{q_i}^{z_0 w}\right) + v_{q_i}^{z_f w/a} t_i + \frac{1}{2}\dot{v}_{q_i}^{z_f w/a} t_i^2\right) \\ &- \frac{24}{t_i^3}\left(v_{q_i}^{z_f w/a} - v_{q_i}^{z_0 f w/a} + \dot{v}_{q_i}^{z_f w/a} t\right), \end{aligned} \tag{4.12}$$

where point z_0 is collocated with the current position of the helicopter's center of mass. The polynomials given by (4.3) to (4.7) are stated in terms of the negative time to target t. Therefore t_i is the initial negative time to target and $t_f = 0$. The value of t_i is chosen such that the resulting velocity profile is monotonic.

The existence of a monotonic velocity profile is guaranteed if the conditions

$$v_{q_1}^{z_0 w/a}\left(r_{q_1}^{z_f w} - r_{q_1}^{z_0 w}\right) > 0, \tag{4.13}$$

$$\frac{-4\left(r_{q_1}^{z_f w} - r_{q_1}^{z_0 w}\right)}{v_{q_1}^{z_0 w/a}} \leq t_i \leq \frac{-2\left(r_{q_1}^{z_f w} - r_{q_1}^{z_0 w}\right)}{v_{q_1}^{z_0 w/a}}, \tag{4.14}$$

are both satisfied in the $\underset{\rightarrow}{q}^1$ direction. Additionally, to avoid a negative thrust command, the condition

$$t_i \leq \frac{3\left(v_{q_3}^{z_f w/a} + v_{q_3}^{z_0 w/a}\right)}{g}\left(1 - \sqrt{1 + \frac{4g\left(r_{q_3}^{z_f w} - r_{q_3}^{z_0 w}\right)}{3\left(v_{q_3}^{z_f w/a} + v_{q_3}^{z_0 w/a}\right)^2}}\right), \tag{4.15}$$

must be met in the $\underset{\rightarrow}{q}^3$ direction. The polynomials are generated both in the $\underset{\rightarrow}{q}^1$ and $\underset{\rightarrow}{q}^3$ directions. In order to have equal trajectory times for the horizontal and vertical profiles, the initial negative time to target is selected as

$$t_i = \min\left(\frac{-2\left(r_{q_1}^{z_f w} - r_{q_1}^{z_0 w}\right)}{v_{q_1}^{z_0 w/a}},\ \frac{3\left(v_{q_3}^{z_f w/a} + v_{q_3}^{z_0 w/a}\right)}{g}\left(1 - \sqrt{1 + \frac{4g\left(r_{q_3}^{z_f w} - r_{q_3}^{z_0 w}\right)}{3\left(v_{q_3}^{z_f w/a} + v_{q_3}^{z_0 w/a}\right)^2}}\right)\right). \tag{4.16}$$

Once the polynomials in the horizontal and vertical directions are computed, the position, velocity, acceleration, jerk, and snap trajectories can be expressed in $\mathcal{F}_a$ as

$$\mathbf{r}_a^{z_q w} = \mathbf{C}_{qa}^{\mathsf{T}} \mathbf{r}_q^{z_q w}, \tag{4.17}$$

$$\mathbf{v}_a^{z_q w/a} = \mathbf{C}_{qa}^{\mathsf{T}} \mathbf{v}_q^{z_q w/a}, \tag{4.18}$$

$$\dot{\mathbf{v}}_a^{z_q w/a} = \mathbf{C}_{qa}^{\mathsf{T}} \dot{\mathbf{v}}_q^{z_q w/a}, \tag{4.19}$$

$$\ddot{\mathbf{v}}_a^{z_q w/a} = \mathbf{C}_{qa}^{\mathsf{T}} \ddot{\mathbf{v}}_q^{z_q w/a}, \tag{4.20}$$

$$\dddot{\mathbf{v}}_a^{z_q w/a} = \mathbf{C}_{qa}^{\mathsf{T}} \dddot{\mathbf{v}}_q^{z_q w/a}. \tag{4.21}$$

4.2.2 Differential Flatness

The differential flatness property of the tandem-rotor system is leveraged to generate the remaining reference state and control input trajectories. For the tandem-rotor system, the flat outputs are the position $\mathbf{r}_a^{z_q w}$ and the heading angle ψ^q. The position trajectory is given by the previously generated quartic polynomial, and the desired heading can be freely chosen. It is assumed in this formulation that the desired heading is constant over the entire trajectory, however this is not a requirement.

The reference control force $\mathbf{f}_a^q$ is calculated using (3.3) and velocity trajectory from (4.18),

$$\mathbf{f}_{a_k}^q = m_{\mathcal{B}} \left(\dot{\mathbf{v}}_{a_k}^{z_q w/a} - \mathbf{g}_a \right). \tag{4.22}$$

The $\underset{\rightarrow}{q}^3$ vector resolved in $\mathcal{F}_a$ is given by

$$\mathbf{q}_{a_k}^3 = \frac{\mathbf{f}_{a_k}^q}{\|\mathbf{f}_{a_k}^q\|}. \tag{4.23}$$

Two intermediate vectors $\mathbf{c}_a^1$ and $\mathbf{c}_a^2$ are formed using the reference heading

$$\mathbf{c}_{a_k}^1 = \begin{bmatrix} \cos(\psi_k^q) \\ \sin(\psi_k^q) \\ 0 \end{bmatrix}, \qquad \mathbf{c}_{a_k}^2 = \begin{bmatrix} -\sin(\psi_k^q) \\ \cos(\psi_k^q) \\ 0 \end{bmatrix}. \tag{4.24}$$

Then the remaining bases vectors of $\mathcal{F}_q$, resolved in $\mathcal{F}_a$, are found by

$$\mathbf{q}_{a_k}^2 = \frac{\mathbf{q}_{a_k}^{3^\times} \mathbf{c}_{a_k}^1}{\left\| \mathbf{q}_{a_k}^{3^\times} \mathbf{c}_{a_k}^1 \right\|}, \tag{4.25}$$

$$\mathbf{q}_{a_k}^1 = \mathbf{q}_{a_k}^{2^\times} \mathbf{q}_{a_k}^3. \tag{4.26}$$

The reference attitude $\mathbf{C}_{aq}$ can be constructed using the bases vectors as

$$\mathbf{C}_{aq_k} = \begin{bmatrix} \mathbf{q}_{a_k}^1 & \mathbf{q}_{a_k}^2 & \mathbf{q}_{a_k}^3 \end{bmatrix}. \tag{4.27}$$

Next, the reference thrust force f^q is computed,

$$f_k^q = -\mathbf{1}_3 \mathbf{C}_{aq_k}^{\mathsf{T}} \mathbf{f}_{a_k}^q, \tag{4.28}$$

while the components of the reference angular velocity $\boldsymbol{\omega}_q^{qa}$ are found by [19]

$$\omega_{q_1}^{qa} = \frac{-\mathcal{B}_1\mathcal{C}_2\mathcal{D}_3 + \mathcal{B}_1\mathcal{C}_3\mathcal{D}_2 - \mathcal{B}_3\mathcal{C}_1\mathcal{D}_2 + \mathcal{B}_3\mathcal{C}_2\mathcal{D}_1}{\mathcal{A}_2\left(\mathcal{B}_1\mathcal{C}_3 - \mathcal{B}_3\mathcal{C}_1\right)}, \tag{4.29}$$

$$\omega_{q_2}^{qa} = \frac{-\mathcal{C}_1\mathcal{D}_3 + \mathcal{C}_3\mathcal{D}_1}{\mathcal{B}_1\mathcal{C}_3 - \mathcal{B}_3\mathcal{C}_1}, \tag{4.30}$$

$$\omega_{q_3}^{qa} = \frac{\mathcal{B}_1\mathcal{D}_3 - \mathcal{B}_3\mathcal{D}_1}{\mathcal{B}_1\mathcal{C}_3 - \mathcal{B}_3\mathcal{C}_1}, \tag{4.31}$$

where

$$\begin{aligned}
\mathcal{B}_1 &= -\frac{f_k^q}{m_{\mathcal{B}}}\mathbf{1}_3 - (d_z - d_x)\left(\left(\mathbf{C}_{aq_k}\mathbf{1}_1\right)^{\mathsf{T}}\mathbf{v}_{a_k}^{z_q w/a}\right), && (4.32)\\
\mathcal{C}_1 &= -\left(d_x - d_y\right)\left(\left(\mathbf{C}_{aq_k}\mathbf{1}_2\right)^{\mathsf{T}}\mathbf{v}_{a_k}^{z_q w/a}\right), && (4.33)\\
\mathcal{D}_1 &= \left(\mathbf{C}_{aq_k}\mathbf{1}_1\right)^{\mathsf{T}}\ddot{\mathbf{v}}_{a_k}^{z_q w/a} + d_x\left(\mathbf{C}_{aq_k}\mathbf{1}_1\right)^{\mathsf{T}}\dot{\mathbf{v}}_{a_k}^{z_q w/a}, && (4.34)\\
\mathcal{A}_2 &= -\frac{f_k^q}{m_{\mathcal{B}}}\mathbf{1}_3 - \left(d_y - d_z\right)\left(\left(\mathbf{C}_{aq_k}\mathbf{1}_1\right)^{\mathsf{T}}\mathbf{v}_{a_k}^{z_q w/a}\right), && (4.35)\\
\mathcal{C}_2 &= \left(d_x - d_y\right)\left(\mathbf{C}_{aq_k}\mathbf{1}_1\right)^{\mathsf{T}}\mathbf{v}_{a_k}^{z_q w/a}, && (4.36)\\
\mathcal{D}_2 &= -\left(\mathbf{C}_{aq_k}\mathbf{1}_2\right)^{\mathsf{T}}\ddot{\mathbf{v}}_{a_k}^{z_q w/a} - d_y\left(\mathbf{C}_{aq_k}\mathbf{1}_2\right)^{\mathsf{T}}\dot{\mathbf{v}}_{a_k}^{z_q w/a}, && (4.37)\\
\mathcal{B}_3 &= -\mathbf{c}_{a_k}^{2^{\mathsf{T}}}\mathbf{C}_{aq_k}\mathbf{1}_3, && (4.38)\\
\mathcal{C}_3 &= \left\|\mathbf{c}_{a_k}^{2^{\times}}\mathbf{C}_{aq_k}\mathbf{1}_3\right\|, && (4.39)\\
\mathcal{D}_3 &= \dot{\psi}_k^q\mathbf{c}_{a_k}^{1^{\mathsf{T}}}\mathbf{C}_{aq_k}\mathbf{1}_1. && (4.40)
\end{aligned}$$

The components of the reference angular acceleration $\dot{\boldsymbol{\omega}}_q^{qa}$ are found by [19]

$$\begin{aligned}
\dot{\omega}_{q_1}^{qa} &= \frac{-\mathcal{B}_1\mathcal{C}_2\mathcal{E}_3 + \mathcal{B}_1\mathcal{C}_3\mathcal{E}_2 - \mathcal{B}_3\mathcal{C}_1\mathcal{E}_2 + \mathcal{B}_3\mathcal{C}_2\mathcal{E}_1}{\mathcal{A}_2\left(\mathcal{B}_1\mathcal{C}_3 - \mathcal{B}_3\mathcal{C}_1\right)}, && (4.41)\\
\dot{\omega}_{q_2}^{qa} &= \frac{-\mathcal{C}_1\mathcal{E}_3 + \mathcal{C}_3\mathcal{E}_1}{\mathcal{B}_1\mathcal{C}_3 - \mathcal{B}_3\mathcal{C}_1}, && (4.42)\\
\dot{\omega}_{q_3}^{qa} &= \frac{\mathcal{B}_1\mathcal{E}_3 - \mathcal{B}_3\mathcal{E}_1}{\mathcal{B}_1\mathcal{C}_3 - \mathcal{B}_3\mathcal{C}_1}, && (4.43)
\end{aligned}$$

where

$$\begin{aligned}
\mathcal{E}_1 &= \left(\mathbf{C}_{aq_k}\mathbf{1}_1\right)^{\mathsf{T}}\dddot{\mathbf{v}}_{a_k}^{z_q w/a} - 2\left(\frac{1}{m_{\mathcal{B}}}\dot{f}_k^q\mathbf{1}_3\right)\omega_{q2_k}^{qa} + \frac{f_k^q}{m_{\mathcal{B}}}\mathbf{1}_3\omega_{q1_k}^{qa}\omega_{q3_k}^{qa} + \left(\mathbf{C}_{aq_k}\mathbf{1}_1\right)^{\mathsf{T}}\boldsymbol{\Gamma}_k, && (4.44)\\
\mathcal{E}_2 &= -\left(\mathbf{C}_{aq_k}\mathbf{1}_2\right)^{\mathsf{T}}\dddot{\mathbf{v}}_{a_k}^{z_q w/a} - 2\left(\frac{1}{m_{\mathcal{B}}}\dot{f}_k^q\mathbf{1}_3\right)\omega_{q1_k}^{qa} - \frac{f_k^q}{m_{\mathcal{B}}}\mathbf{1}_3\omega_{q2_k}^{qa}\omega_{q3_k}^{qa} - \left(\mathbf{C}_{aq_k}\mathbf{1}_2\right)^{\mathsf{T}}\boldsymbol{\Gamma}_k, && (4.45)\\
\mathcal{E}_3 &= \ddot{\psi}_k^q\mathbf{c}_{a_k}^{1^{\mathsf{T}}}\mathbf{C}_{aq_k}\mathbf{1}_1 + 2\dot{\psi}_k^q\omega_{q3_k}^{qa}\mathbf{c}_{a_k}^{1^{\mathsf{T}}}\mathbf{C}_{aq_k}\mathbf{1}_2 - 2\dot{\psi}_k^q\omega_{q2_k}^{qa}\mathbf{c}_{a_k}^{1^{\mathsf{T}}}\mathbf{C}_{aq_k}\mathbf{1}_3\\
&\quad - \omega_{q1_k}^{qa}\omega_{q2_k}^{qa}\mathbf{c}_{a_k}^{2^{\mathsf{T}}}\mathbf{C}_{aq_k}\mathbf{1}_2 - \omega_{q1_k}^{qa}\omega_{q3_k}^{qa}\mathbf{c}_{a_k}^{2^{\mathsf{T}}}\mathbf{C}_{aq_k}\mathbf{1}_3, && (4.46)
\end{aligned}$$

and [19]

$$
\begin{aligned}
\frac{1}{m_{\mathcal{B}}} \dot{f}_k^q \mathbf{1}_3 = {} & \left(\mathbf{C}_{aq_k} \mathbf{1}_3\right)^{\mathsf{T}} \ddot{\mathbf{v}}_{a_k}^{z_q w/a} + \omega_{q_{1_k}}^{qa} \left(d_y - d_z\right) \left(\mathbf{C}_{aq_k} \mathbf{1}_2\right)^{\mathsf{T}} \mathbf{v}_{a_k}^{z_q w/a} \\
& + \omega_{q_{2_k}}^{qa} \left(d_z - d_x\right) \left(\mathbf{C}_{aq_k} \mathbf{1}_1\right)^{\mathsf{T}} \mathbf{v}_{a_k}^{z_q w/a} + d_z \left(\mathbf{C}_{aq_k} \mathbf{1}_3\right)^{\mathsf{T}} \dot{\mathbf{v}}_{a_k}^{z_q w/a},
\end{aligned} \tag{4.47}
$$

$$
\begin{aligned}
\mathbf{\Gamma}_k = {} & \mathbf{C}_{aq_k} \left(\left(\boldsymbol{\omega}_{q_k}^{qa^\times}\right)^2 \mathbf{D} + \mathbf{D} \left(\boldsymbol{\omega}_{q_k}^{qa^\times}\right)^2 - 2\boldsymbol{\omega}_{q_k}^{qa^\times} \mathbf{D} \boldsymbol{\omega}_{q_k}^{qa^\times} \right) \mathbf{C}_{aq_k}^{\mathsf{T}} \mathbf{v}_{a_k}^{z_q w/a} \\
& + 2\mathbf{C}_{aq_k} \left(\boldsymbol{\omega}_{q_k}^{qa^\times} \mathbf{D} - \mathbf{D} \boldsymbol{\omega}_{q_k}^{qa^\times} \right) \mathbf{C}_{aq_k}^{\mathsf{T}} \dot{\mathbf{v}}_{a_k}^{z_q w/a} + \mathbf{C}_{aq_k} \mathbf{D} \mathbf{C}_{aq_k}^{\mathsf{T}} \ddot{\mathbf{v}}_{a_k}^{z_q w/a}.
\end{aligned} \tag{4.48}
$$

Finally, the torque input $\mathbf{m}_{q_k}^q$ can be calculated as [19]

$$
\mathbf{m}_{q_k}^q = \mathbf{J}_q^{\mathcal{B} z_q} \dot{\boldsymbol{\omega}}_{q_k}^{qa} + \boldsymbol{\omega}_{q_k}^{qa^\times} \mathbf{J}_q^{\mathcal{B} z_q} \boldsymbol{\omega}_{q_k}^{qa} + \mathbf{E} \mathbf{C}_{aq_k}^{\mathsf{T}} \mathbf{v}_{a_k}^{z_q w/a} + \mathbf{F} \boldsymbol{\omega}_{q_k}^{qa}. \tag{4.49}
$$

The quartic reference state trajectory, $\mathbf{x}^q$, at time k is

$$
\mathbf{x}_k^q = \begin{bmatrix} \mathbf{r}_{a_k}^{z_q w} \\ \mathbf{q}_k^{qa} \\ \mathbf{v}_{a_k}^{z_q w/a} \\ \boldsymbol{\omega}_{q_k}^{qa} \end{bmatrix} \in \mathbb{R}^{18}, \tag{4.50}
$$

where $\mathbf{q}_k^{qa}$ is a parameterization of the DCM $\mathbf{C}_{aq_k}$ given by (2.9). The quartic reference control input trajectory, $\mathbf{u}^q$, at time k is

$$
\mathbf{u}_k^q = \begin{bmatrix} f_k^q \\ \mathbf{m}_{q_k}^q \end{bmatrix} \in \mathbb{R}^4. \tag{4.51}
$$

4.3 Linear Quadratic Regulator Trajectory Generation

The quartic trajectory is used to warm start a standard discrete-time, finite-horizon LQR problem [34]. While LQR has traditionally been used as a strategy for optimal control, as demonstrated in [20], in this thesis LQR is used as a guidance strategy. The objective is to refine the coarse quartic reference trajectory and generate a high quality feedforward term for the MPC. The discrete-time, finite-horizon LQR problem formulation is similar to that of the MPC problem, allowing for the same augmented $SE_2(3)$ error definition to be leveraged.

4.3.1 Optimal Gain Sequence

The first step in the LQR trajectory generation process is to compute the optimal LQR gain sequence along the desired trajectory. Consider the system dynamics, linearized about the quartic reference trajectory $\mathbf{x}^q$, and control inputs $\mathbf{u}^q$, to give the linearized state-space model

$$\delta\dot{\bar{\mathbf{x}}}^r = \bar{\mathbf{A}}^r\delta\bar{\mathbf{x}}^r + \bar{\mathbf{B}}^r\delta\mathbf{u}^r, \tag{4.52}$$

where the Jacobians $\bar{\mathbf{A}}^r$ and $\bar{\mathbf{B}}^r$ are given by

$$\bar{\mathbf{A}}^r = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{J}_q^{\mathcal{B}z_q^{-1}} & \mathbf{0} \\ \bar{\mathbf{A}}_{21}^r & \bar{\mathbf{A}}_{22}^r & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & -\boldsymbol{\omega}_q^{qa^\times} & \mathbf{0} & \mathbf{0} \\ \bar{\mathbf{A}}_{41}^r & -\mathbf{E} & \mathbf{0} & \bar{\mathbf{A}}_{44}^r & \mathbf{0} \\ \mathbf{0} & c_2\mathbf{1} & c_1\mathbf{1} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \bar{\mathbf{B}}^r = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -1/m_{\mathcal{B}}\mathbf{1}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{4.53}$$

and the matrices $\bar{\mathbf{A}}_{21}^r$ and $\bar{\mathbf{A}}_{22}^r$ are given by

$$\bar{\mathbf{A}}_{21}^r = \frac{1}{m_{\mathcal{B}}}\left(\left(\mathbf{D}\mathbf{C}_{aq}^{\mathsf{T}}\mathbf{v}_a^{z_qw/a}\right)^\times - \mathbf{D}\left(\mathbf{C}_{aq}^{\mathsf{T}}\mathbf{v}_a^{z_qw/a}\right)^\times + \left(f^q\mathbf{1}_3\right)^\times\right), \tag{4.54}$$

$$\bar{\mathbf{A}}_{22}^r = -\boldsymbol{\omega}_q^{qa^\times} - \frac{1}{m_{\mathcal{B}}}\mathbf{D}, \tag{4.55}$$

$$\bar{\mathbf{A}}_{41}^r = \left(\mathbf{E}\mathbf{C}_{aq}^{\mathsf{T}}\mathbf{v}_a^{z_qw/a}\right)^\times - \mathbf{E}\left(\mathbf{C}_{aq}^{\mathsf{T}}\mathbf{v}_a^{z_qw/a}\right)^\times + \left(\mathbf{F}\boldsymbol{\omega}_q^{qa}\right)^\times - \mathbf{F}\boldsymbol{\omega}_q^{qa^\times}, \tag{4.56}$$

$$\bar{\mathbf{A}}_{44}^r = -\boldsymbol{\omega}_q^{qa^\times} - \mathbf{F}\mathbf{J}_q^{\mathcal{B}z_q^{-1}}. \tag{4.57}$$

Note that the linearized system has been augmented with an integrator as shown in Section 3.6, and that the form of these Jacobians are identical to those given by (3.70). Recall that the script q refers to the reference frame $\mathcal{F}_q$ associated with the quartic trajectory. The script r is used to refer to the reference frame $\mathcal{F}_r$ associated with the guidance LQR trajectory.

The augmented, linearized, continuous-time dynamics are discretized as shown in Section 3.7 to give the discrete-time dynamics

$$\delta\mathbf{x}_{k+1}^r = \bar{\mathbf{A}}_k^r\delta\bar{\mathbf{x}}_k^r + \bar{\mathbf{B}}_k^r\delta\mathbf{u}_k^r. \tag{4.58}$$

Now the relevant reference states from the quartic trajectory, $\mathbf{x}^q$, can be used to populate (4.53) and create a sequence of $\bar{\mathbf{A}}_k^r$ and $\bar{\mathbf{B}}_k^r$ Jacobians.

The LQR cost function is of the form [34]

$$J\left(\delta\mathbf{u}_0^r, \cdots, \delta\mathbf{u}_{N-1}^r\right) = \tfrac{1}{2}\delta\bar{\mathbf{x}}_N^{r^\mathsf{T}}\mathbf{S}\delta\bar{\mathbf{x}}_N^r + \tfrac{1}{2}\sum_{k=0}^{N-1}\left(\delta\bar{\mathbf{x}}_k^{r^\mathsf{T}}\mathbf{Q}\delta\bar{\mathbf{x}}_k^r + \delta\mathbf{u}_k^{r^\mathsf{T}}\mathbf{R}\delta\mathbf{u}_k^r\right), \tag{4.59}$$

where N is the horizon length, $\mathbf{S} = \mathbf{S}^\mathsf{T} \geq 0$ is the terminal state error penalty, $\mathbf{Q} = \mathbf{Q}^\mathsf{T} \geq 0$ is the state error penalty, and $\mathbf{R} = \mathbf{R}^\mathsf{T} > 0$ is the control input penalty. The optimal control input is given by the solution to the discrete-time, finite-horizon LQR problem,

$$\delta\mathbf{u}_k^r = -\mathbf{K}_k\delta\bar{\mathbf{x}}_k^r, \tag{4.60}$$

where the LQR gain $\mathbf{K}_k$ is given by

$$\mathbf{K}_k = \bar{\mathbf{R}}_{k+1}^{-1}\bar{\mathbf{B}}_k^{r^\mathsf{T}}\mathbf{P}_{k+1}\bar{\mathbf{A}}_k^r, \tag{4.61}$$

and

$$\bar{\mathbf{R}}_{k+1} = \mathbf{R} + \bar{\mathbf{B}}_k^{r^\mathsf{T}}\mathbf{P}_{k+1}\bar{\mathbf{B}}_k^r. \tag{4.62}$$

The sequence of $\mathbf{P}_k$, for $k = 0$ to $k = N$ is found by solving the discrete-time Riccati equation backwards in time. Starting from the terminal condition $\mathbf{P}_N = \mathbf{S}$, each successive $\mathbf{P}_k$ is solved by

$$\mathbf{P}_k = \bar{\mathbf{A}}_k^{r^\mathsf{T}}\left(\mathbf{P}_{k+1} - \mathbf{P}_{k+1}\bar{\mathbf{B}}_k^r\bar{\mathbf{R}}_{k+1}^{-1}\bar{\mathbf{B}}_k^{r^\mathsf{T}}\mathbf{P}_{k+1}\right)\bar{\mathbf{A}}_k^r + \mathbf{Q}. \tag{4.63}$$

Once $\mathbf{P}_k$ is found, $\bar{\mathbf{R}}_{k+1}$ is found using (4.62), and $\mathbf{K}_k$ is found using (4.61). This process is repeated N times to calculate the sequence of gains $\mathbf{K}_k$ across the entire horizon.

4.3.2 Reference State and Control Input Trajectory

The sequence of reference control inputs, $\mathbf{u}^r$ and states $\mathbf{x}^r$ are then generated by propagating the closed-loop dynamics. At each timestep, the current vehicle attitude, $\mathbf{C}_{ar}$, velocity, $\mathbf{v}_a^{z_r w/a}$, and position, $\mathbf{r}_a^{z_r w}$, are cast into an element of $SE_2(3)$,

$$\mathbf{X}^r = \begin{bmatrix} \mathbf{C}_{ar} & \mathbf{v}_a^{z_r w/a} & \mathbf{r}_a^{z_r w} \\ \mathbf{0} & 1 & 0 \\ \mathbf{0} & 0 & 1 \end{bmatrix} \in SE_2(3). \tag{4.64}$$

Similarly, the desired vehicle attitude, $\mathbf{C}_{aq}$, velocity, $\mathbf{v}_a^{z_q w/a}$, and position, $\mathbf{r}_a^{z_q w}$, states from the quartic trajectory are cast into an element of $SE_2(3)$,

$$\mathbf{X}^q = \begin{bmatrix} \mathbf{C}_{aq} & \mathbf{v}_a^{z_q w/a} & \mathbf{r}_a^{z_q w} \\ \mathbf{0} & 1 & 0 \\ \mathbf{0} & 0 & 1 \end{bmatrix} \in SE_2(3). \tag{4.65}$$

The left-invariant tracking error is given by

$$\delta \mathbf{X}_k^r = \mathbf{X}_k^{q^{-1}} \mathbf{X}_k^r = \begin{bmatrix} \delta \mathbf{C}_k^r & \delta \mathbf{v}_k^r & \mathbf{r}_k^r \\ \mathbf{0} & 1 & 0 \\ \mathbf{0} & 0 & 1 \end{bmatrix}. \tag{4.66}$$

The tracking error is then cast to the Lie algebra as

$$\delta \boldsymbol{\xi}_k^r = \log \left(\delta \mathbf{X}_k^r \right)^{\vee}, \tag{4.67}$$

and the angular momentum error is given by

$$\delta \mathbf{h}_k^r = \delta \mathbf{C}_k^r \mathbf{h}_{q_k}^{\mathcal{B} z_q a} - \mathbf{h}_{r_k}^{\mathcal{B} z_r a}. \tag{4.68}$$

Therefore, the complete error state is given by

$$\delta \bar{\mathbf{x}}_k^r = \begin{bmatrix} \delta \boldsymbol{\xi}_k^r \\ \delta \mathbf{h}_k^r \\ \delta \boldsymbol{\xi}_k^i \end{bmatrix} = \begin{bmatrix} \delta \boldsymbol{\xi}_k^{\phi^r} \\ \delta \boldsymbol{\xi}_k^{v^r} \\ \delta \boldsymbol{\xi}_k^{r^r} \\ \delta \mathbf{h}_k^r \\ \delta \boldsymbol{\xi}_k^i \end{bmatrix} \in \mathbb{R}^{18}. \tag{4.69}$$

Next, the optimal incremental control input $\delta \mathbf{u}_k^r$ is computed though (4.60) using the optimal LQR gain, $\mathbf{K}_k$ and the current state error, $\delta \bar{\mathbf{x}}_k^r$. The quartic feedforward term, $\mathbf{u}_k^q$, is added to $\delta \mathbf{u}_k^r$ to produce the total reference control input $\mathbf{u}_k^r$,

$$\mathbf{u}_k^r = \begin{bmatrix} f_k^r \\ \mathbf{m}_{r_k}^r \end{bmatrix} = \begin{bmatrix} f_k^q + \delta f_k^r \\ \delta \mathbf{C}_k^{r^{\mathsf{T}}} \mathbf{m}_{q_k}^q + \delta \mathbf{m}_k^r \end{bmatrix} \in \mathbb{R}^4. \tag{4.70}$$

The nonlinear vehicle dynamics given by (3.1) to (3.4) and the integrator dynamics given by (3.65) are integrated using a forward Euler method to generate the reference states, $\mathbf{x}^r_{k+1}$ as

$$\mathbf{C}_{ar_{k+1}} = \mathbf{C}_{ar_k} \exp\left(\left(h\boldsymbol{\omega}^{ra}_{r_k}\right)^\times\right), \tag{4.71}$$

$$\mathbf{r}^{z_r w}_{a_{k+1}} = \mathbf{r}^{z_r w}_{a_k} + h\mathbf{v}^{z_r w/a}_{a_k}, \tag{4.72}$$

$$\mathbf{v}^{z_r w/a}_{a_{k+1}} = \mathbf{v}^{z_r w/a}_{a_k} + h\frac{1}{m_{\mathcal{B}}}\mathbf{f}^{\mathcal{B}z_r}_{a_k}, \tag{4.73}$$

$$\boldsymbol{\omega}^{ra}_{r_{k+1}} = \boldsymbol{\omega}^{ra}_{r_k} + h\mathbf{J}_r^{\mathcal{B}z_r^{-1}}\left(\mathbf{m}^{\mathcal{B}z_r}_{r_k} - \boldsymbol{\omega}^{ra^\times}_{r_k}\mathbf{J}^{\mathcal{B}z}_r\boldsymbol{\omega}^{ra}_{r_k}\right), \tag{4.74}$$

$$\boldsymbol{\xi}^i_{k+1} = \boldsymbol{\xi}^i_k + h\left(c_1\delta\mathbf{r}^r_k + c_2\delta\mathbf{v}^r_k\right), \tag{4.75}$$

where h is the timestep.

The steps (4.64) to (4.75) are repeated at every timestep to generate a complete set of refined reference trajectories for the states, $\mathbf{x}^r$, and control inputs, $\mathbf{u}^r$.

4.4 Simulation Results

To demonstrate the proposed two-stage guidance law, a simulation is run to generate a reference trajectory for a landing approach maneuver. The states are initialized as

$$\begin{aligned}
\mathbf{C}_{ab_0} &= \exp_{SO(3)}\left(\begin{bmatrix} 0 & 0 & -\pi/36 \end{bmatrix}^\mathsf{T}\right), \\
\mathbf{r}^{z_0 w}_a &= \begin{bmatrix} -30 & -5 & -20 \end{bmatrix}^\mathsf{T} \text{ m}, \\
\mathbf{v}^{z_0 w/a}_a &= \begin{bmatrix} 5 & -1 & 0.2 \end{bmatrix}^\mathsf{T} \text{ m} \cdot \text{s}^{-1}, \\
\boldsymbol{\omega}^{b_0 a}_b &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\mathsf{T} \text{ rad} \cdot \text{s}^{-1}.
\end{aligned}$$

The parameters used by the guidance LQR algorithm are given in Table 4.1. The generated reference path from the initial position to the target is shown in Figure 4.2. The initial path planned by the quartic guidance law is simply defined by the fourth-order polynomial given by (4.3). The guidance LQR algorithm refines the path based on the LQR cost function and the initial conditions.

The generated state and input trajectories are shown in Figures 4.3 and 4.4, respectively. The polynomial function used by the quartic guidance law can be seen in the position and velocity trajectories. Note that while the algorithm can always perfectly match the initial vehicle position state, it is not always possible to match the initial velocity state. Recall from

Table 4.1: Guidance LQR Parameters Used in Trajectory Generation Simulation

Parameter	Value	Units
dt	0.02	s
$\mathbf{Q}^{\phi}$	$\text{diag}(1\text{e}3, 1\text{e}3, 1\text{e}6)$	-
$\mathbf{Q}^{v}$	$\text{diag}(1\text{e}1, 1\text{e}1, 1\text{e}1)$	-
$\mathbf{Q}^{r}$	$\text{diag}(1\text{e}2, 1\text{e}2, 1\text{e}2)$	-
$\mathbf{Q}^{h}$	$\text{diag}(1\text{e}2, 1\text{e}2, 1\text{e}2)$	-
$\mathbf{Q}^{i}$	$\text{diag}(1, 1, 1)$	-
$\mathbf{Q}$	$\text{diag}(\mathbf{Q}^{\phi}, \mathbf{Q}^{v}, \mathbf{Q}^{r}, \mathbf{Q}^{h}, \mathbf{Q}^{i})$	-
$\mathbf{S}$	$10 \cdot \mathbf{Q}$	-
$\mathbf{R}$	$\text{diag}(1\text{e} - 3, 1, 1, 1)$	-
c_1	1	-
c_2	1	-

Section 4.2.1 that the condition given by (4.13) must be satisfied to generate a trajectory with a monotonic velocity profile. This condition is only satisfied if the vehicle's initial velocity is in the direction of the target, which is not guaranteed. In the case that (4.13) is violated, the quartic guidance law assumes a valid initial velocity that allows the trajectory to be generated. The guidance LQR is then able to smooth the error between the reference velocity from the quartic guidance and true initial velocity. Similarly, the initial attitude and angular velocity states are not considered by the quartic guidance, however the guidance LQR ensures that all boundary conditions are matched, and that the final reference trajectories sent to the controller are smooth.

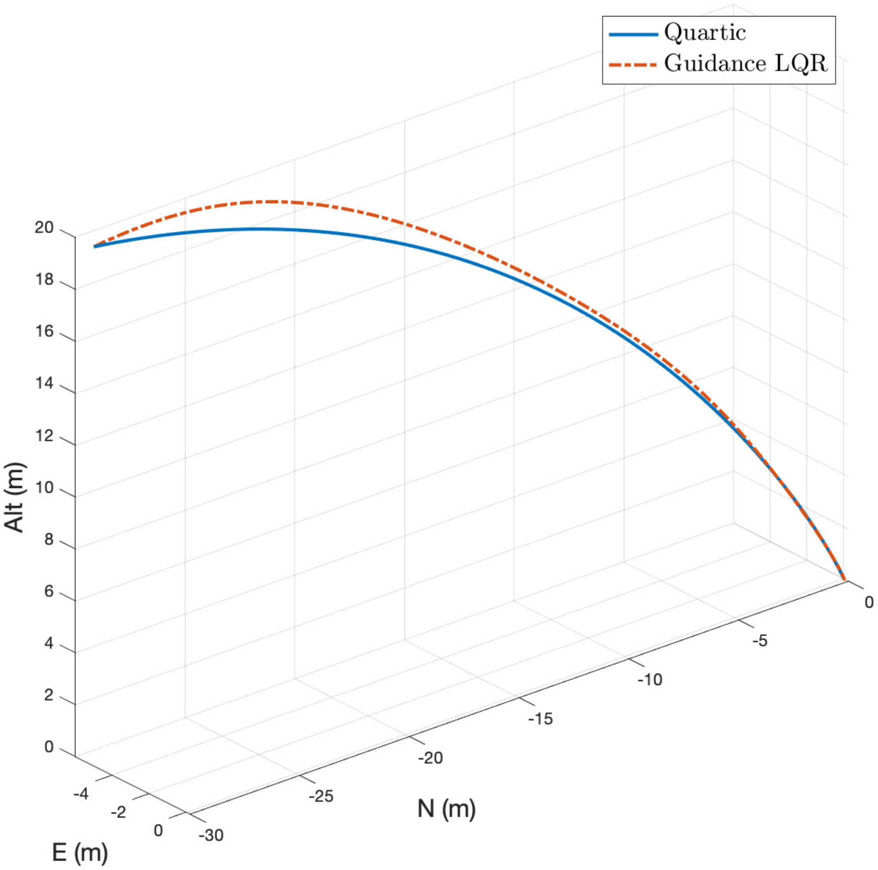
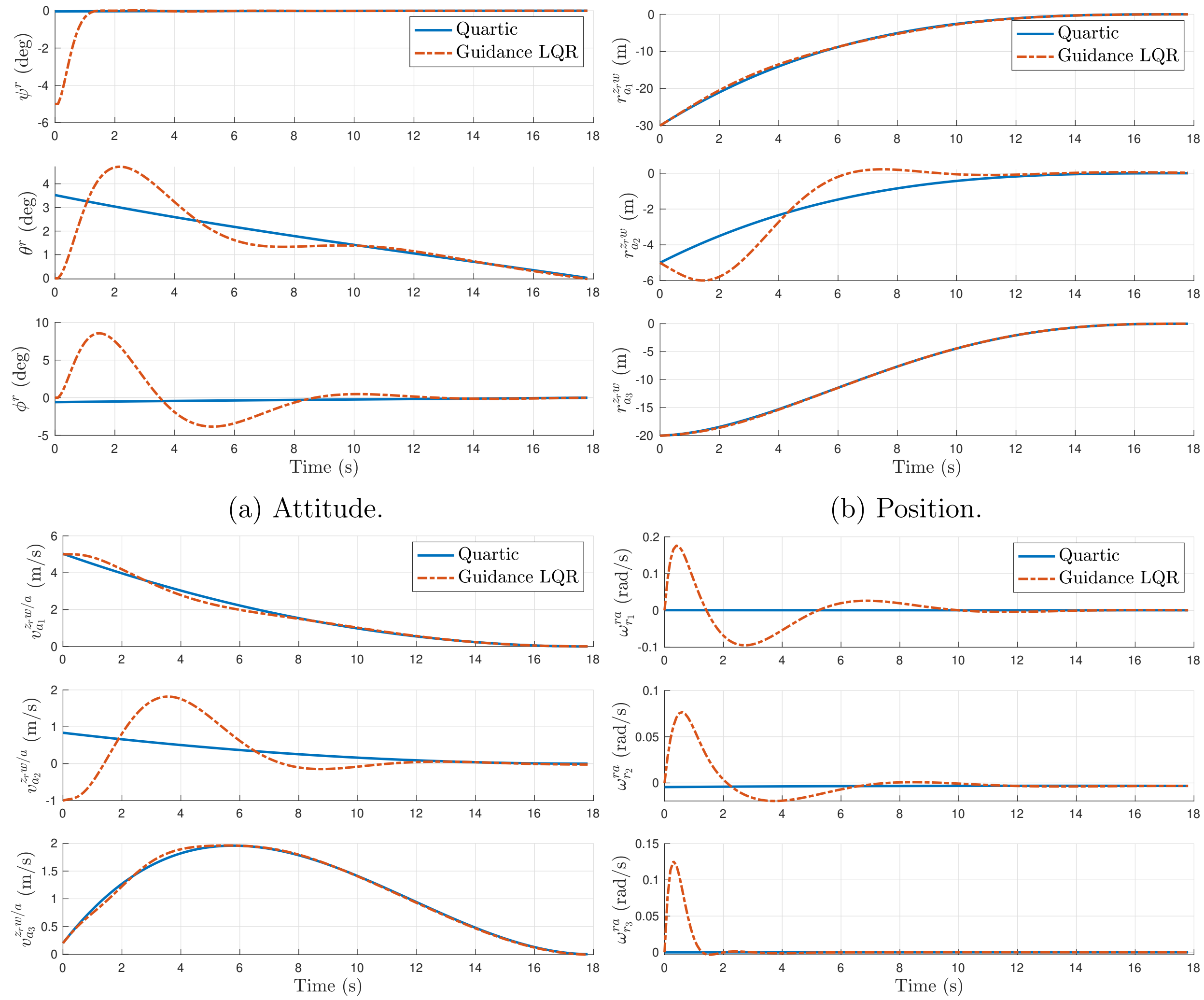


Figure 4.2: Simulated landing approach path generated by two-stage guidance.



(a) Attitude.

(b) Position.

(c) Velocity.

(d) Angular Velocity.

Figure 4.3: Simulated landing approach state trajectories generated by two-stage guidance.

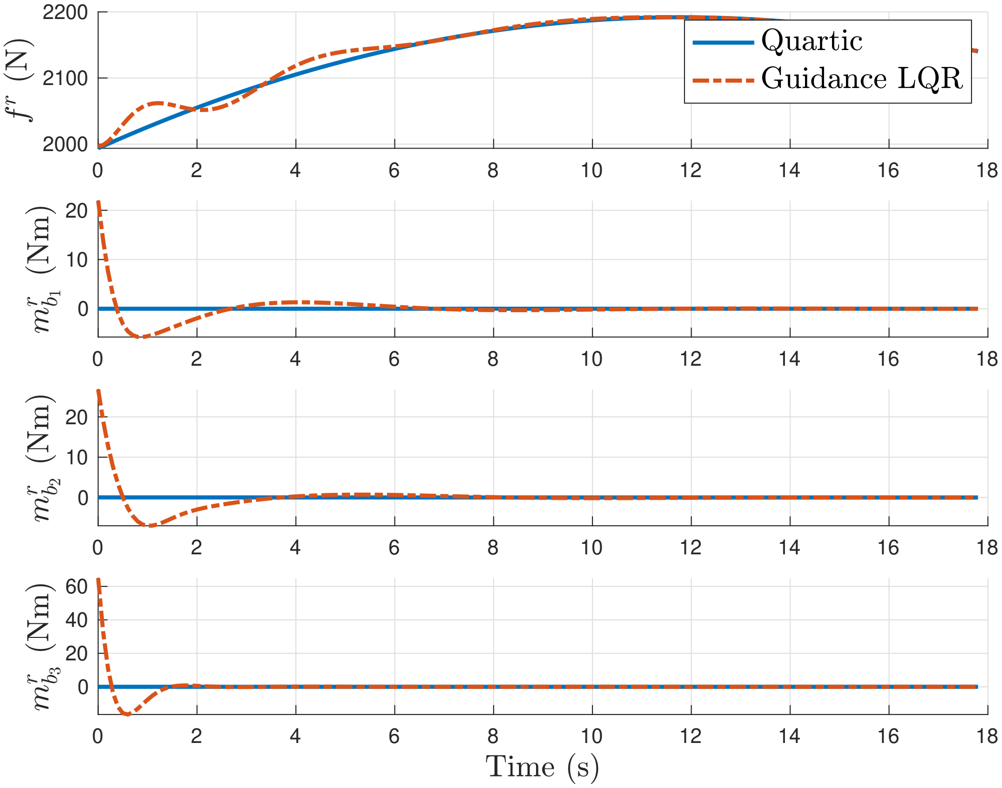


Figure 4.4: Simulated landing approach control input trajectories generated by two-stage guidance.

Chapter 5

Model Predictive Control

Now that a guidance strategy has been developed, a control strategy is necessary to generate actuator inputs that will allow the tandem-rotor helicopter to follow the desired reference trajectory. Although not commonly studied, several examples exist in the literature of control strategies applied to tandem-rotor helicopters. In [35], a proportional-integral (PI) velocity and heading controller is developed for a Chinook helicopter autoland system, while a nonlinear Lyapunov algorithm for hover control is demonstrated in [36]. Lastly, an $\mathcal{L}_1$ adaptive controller is designed using a linearized model of the system dynamics in [37].

In this chapter, an MPC approach for tandem-rotor control is presented. The MPC framework offers a number of advantages over other methods of control. It is relatively simple to design, particularly for systems with many states and control inputs. More importantly, MPC provides a framework that allows state and control inputs to be directly constrained, which is not always straightforward with other control methods. Historically, MPC was developed in the 1960s and was primarily used in the refining and petrochemical industries until the early 2000s [6]. Since then, MPC has found widespread applications in a range of industries including automotive and aerospace control, information technology, energy, and finance [5].

MPC in its most basic form has several that must be addressed in order to be successfully implemented on the tandem-rotor system. They are as follows.

1. MPC can be unsuitable for large real-time systems with fast dynamics and many constraints.
2. Nonlinear systems cannot be posed as a QP.

Additionally, the tracking performance of the controller is dependent on the accuracy of the selected dynamic model and state estimates provided to the controller. Although this is not

MPC specific, and is in fact a weakness of any model-based scheme, it is an issue that must be addressed to ensure good performance. This chapter aims to develop an MPC strategy for the tandem-rotor helicopter that addresses each of these weaknesses.

5.1 Overview

Matrix Lie groups can be used to compactly and accurately represent vehicle attitude, pose, or extended pose, which in turn can be leveraged in state estimation and control problems [20, 38–42]. In particular, an invariant linear quadratic Gaussian controller defined on $SE(2)$ is used to control a simplified car in [40], and invariant LQR using an error defined on $SE_2(3)$ is used to control a quadrotor in [20]. The use of an invariant error definition, instead of a traditional multiplicative error definition [23], along with a particular type of process model, results in Jacobians that are state-independent. This yields improved robustness to initial conditions for state-estimation and control. MPC on matrix Lie groups has been explored in [41] and [42] for spacecraft attitude control on $SO(3)$.

In this chapter, an invariant error definition [38, 39] is used to develop the MPC strategy. The attitude, velocity, and position states are cast into an element of the matrix Lie group $SE_2(3)$ [39]. The nonlinear dynamics from Chapter 3 are linearized about the reference trajectory from Chapter 4, using the linearization process from Chapter 3. This allows the MPC optimization problem to be posed as a QP. The QP is subject to various linear constraints on the control inputs and states. The dynamics used for control design do not exactly fit the invariant framework. Nevertheless, the invariant approach to control is followed due to the straightforward linearization process and the reduced dependence of the Jacobians on attitude [20, 40].

Unlike [20], where the angular velocity is a control input to the plant, the proposed MPC algorithm is able to control and constrain the body force and torque directly. This is accomplished by augmenting the $SE_2(3)$ state with the vehicle angular momentum, eliminating the need for a separate inner-loop controller to generate torque commands. A challenge with this approach is that the resulting process model is multi-timescale, necessitating a small controller timestep and long MPC prediction horizon [43]. This combination is computationally burdensome and is remedied by introducing a non-uniformly spaced prediction horizon, much like [44] and [45].

5.2 Introduction to MPC

MPC is a method of optimal control that uses a simplified dynamic model of the system to predict its likely future evolution and chooses an optimal control action to apply at the current time [5]. An MPC algorithm consists of three steps. At each time k,

1. The estimate of the current state, $\mathbf{x}_k$, is updated.
2. An optimization problem is solved to get the sequence of optimal control inputs, $\boldsymbol{\mu}_k^\star$.
3. The first optimal control input from the sequence, $\mathbf{u}_{0|k}^\star$, is applied to the system.

These steps are repeated continuously until the control objective is reached.

The proposed control structure is highlighted in Figure 5.2a and expanded to show the detailed MPC pipeline in Figure 5.2b. First, the nonlinear system dynamics are linearized about the reference trajectory from the guidance and discretized. Next, the MPC prediction and constraint matrices are created to formulate the QP. A QP solver then solves the optimization problem yielding the sequence of optimal control inputs. Lastly, the feedforward control input from the guidance is added to the optimal feedback control input to form the total input. The remainder of this chapter explains these steps in detail.

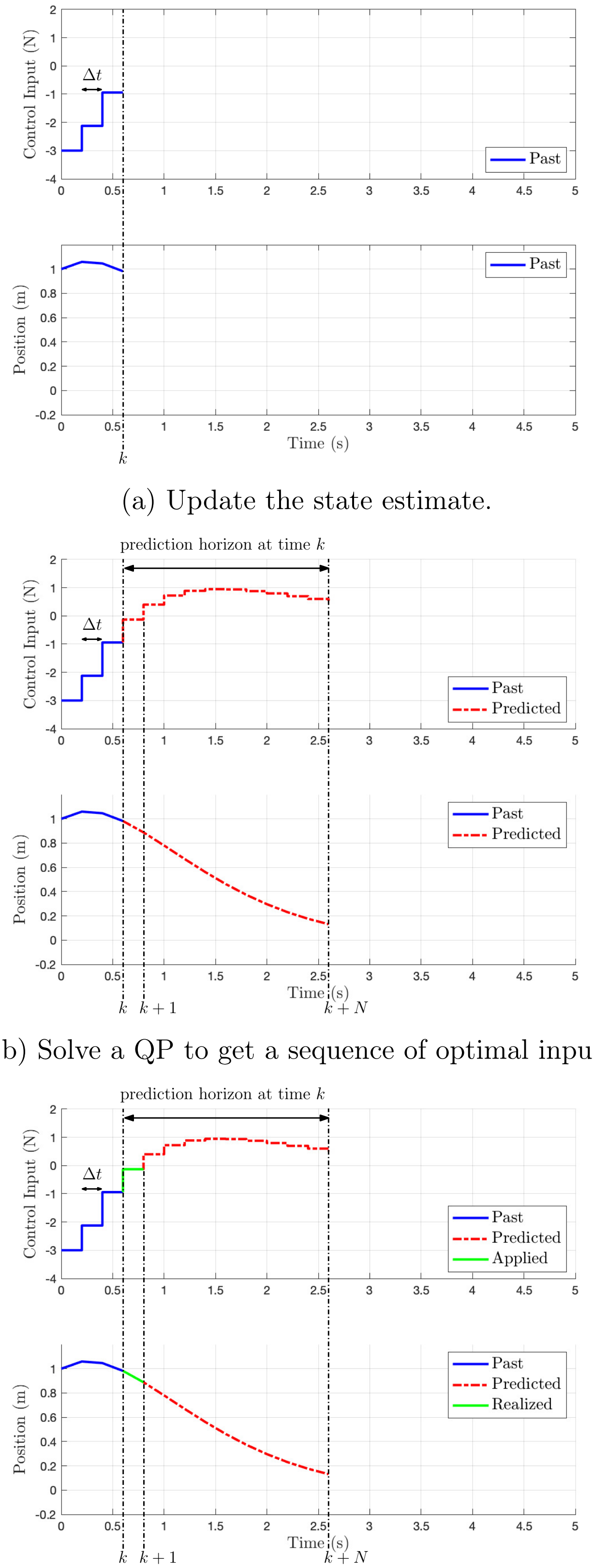
5.3 Finite Horizon MPC for Linear Time-varying Systems

For the discrete-time-linearized system (3.76), the state predictions are written as

$$\delta\mathbf{x}_{i|k} = \boldsymbol{\mathcal{A}}_i(k)\delta\mathbf{x}_k + \boldsymbol{\mathcal{C}}_i(k)\delta\boldsymbol{\mu}_k, \qquad i = 0, \dots, N_p, \tag{5.1}$$

where $\delta\mathbf{x}_{i|k}$ is the state vector given by (3.14) at time $k+i$ predicted at time k, N_p is the prediction horizon length, and $\delta\boldsymbol{\mu}_k$ is the predicted input sequence,

$$\delta\boldsymbol{\mu}_k = \begin{bmatrix} \delta\mathbf{u}_{0|k} \\ \delta\mathbf{u}_{1|k} \\ \vdots \\ \delta\mathbf{u}_{N_p-1|k} \end{bmatrix}. \tag{5.2}$$

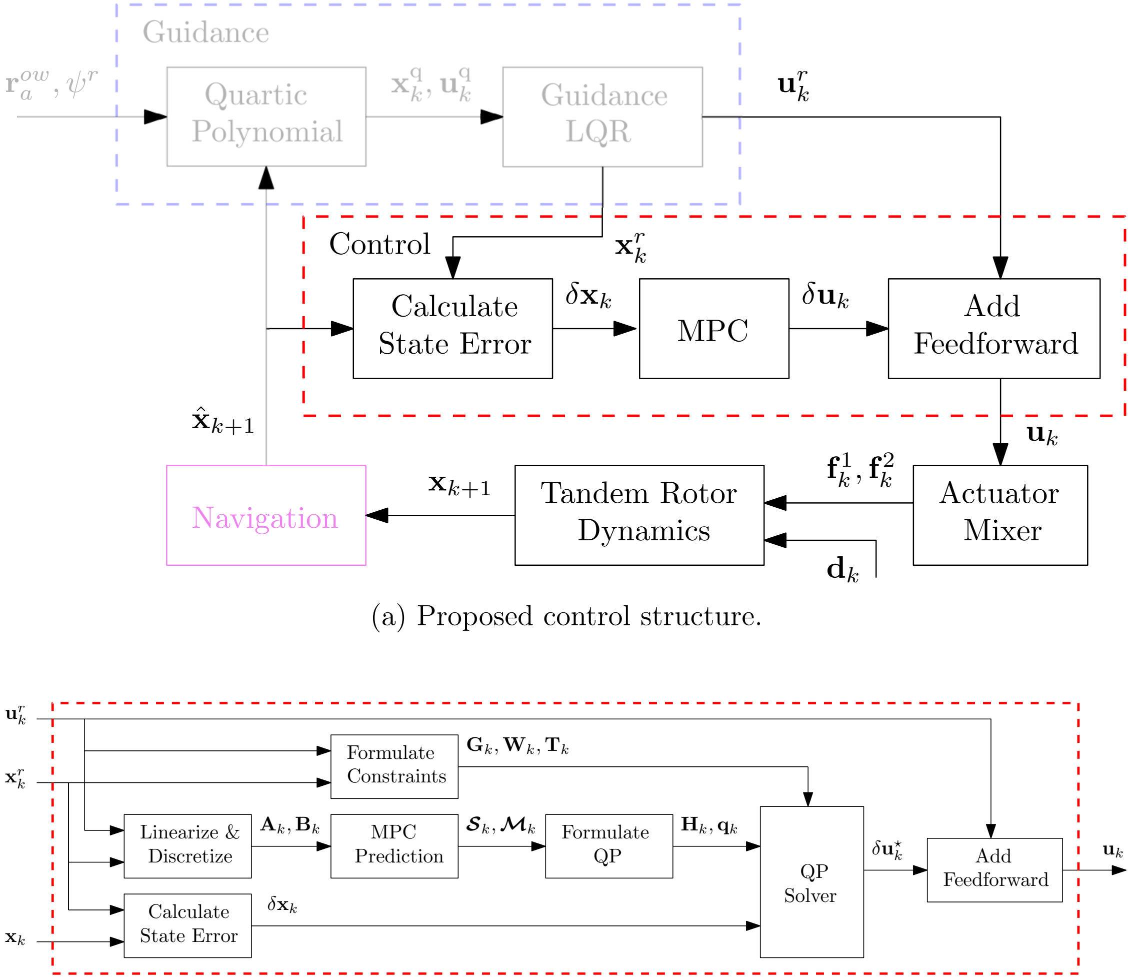


(a) Update the state estimate.

(b) Solve a QP to get a sequence of optimal inputs.

(c) Apply the first input from the sequence.

Figure 5.1: MPC algorithm demonstration.



(a) Proposed control structure.

(b) Detailed MPC pipeline.

Figure 5.2: Proposed control structure and expanded MPC pipeline.

The time-varying state transition matrices, $\boldsymbol{\mathcal{A}}_i(k)$ and $\boldsymbol{\mathcal{C}}_i(k)$, are formed using $\mathbf{A}_k$ and $\mathbf{B}_k$ given by (3.74) and (3.75), and are defined as

$$\boldsymbol{\mathcal{A}}_i(k) = \overset{\curvearrowleft}{\prod}_{j=0}^{i-1} \mathbf{A}_{k+j}, \tag{5.3}$$

$$\boldsymbol{\mathcal{C}}_i(k) = \left[\left(\overset{\curvearrowleft}{\prod}_{j=1}^{i-1} \mathbf{A}_{k+j} \right) \mathbf{B}_k \quad \left(\overset{\curvearrowleft}{\prod}_{j=2}^{i-1} \mathbf{A}_{k+j} \right) \mathbf{B}_{k+1} \quad \cdots \quad \mathbf{B}_{k+i-1} \quad \mathbf{0} \quad \cdots \quad \mathbf{0} \right], \tag{5.4}$$

where $\overset{\curvearrowleft}{\prod}$ indicates that successive terms in the sequence are left-multiplied. The cost matrices $\boldsymbol{\mathcal{S}}_k$ and $\boldsymbol{\mathcal{M}}_k$ are defined by stacking the state transition matrices as

$$\boldsymbol{\mathcal{S}}_k = \begin{bmatrix} \boldsymbol{\mathcal{C}}_1(k) \\ \boldsymbol{\mathcal{C}}_2(k) \\ \vdots \\ \boldsymbol{\mathcal{C}}_{N_p}(k) \end{bmatrix}, \qquad \boldsymbol{\mathcal{M}}_k = \begin{bmatrix} \boldsymbol{\mathcal{A}}_1(k) \\ \boldsymbol{\mathcal{A}}_2(k) \\ \vdots \\ \boldsymbol{\mathcal{A}}_{N_p}(k) \end{bmatrix}. \tag{5.5}$$

Therefore, the predicted state sequence over the prediction horizon, $\delta\boldsymbol{\chi}_k$, can be compactly written as

$$\delta\boldsymbol{\chi}_k = \begin{bmatrix} \delta\mathbf{x}_{1|k} \\ \delta\mathbf{x}_{2|k} \\ \vdots \\ \delta\mathbf{x}_{N_p|k} \end{bmatrix} = \boldsymbol{\mathcal{S}}_k \delta\boldsymbol{\mu}_k + \boldsymbol{\mathcal{M}}_k \delta\mathbf{x}_k. \tag{5.6}$$

Note that $\delta\mathbf{x}_k = \delta\mathbf{x}_{0|k}$ is the current state error at time k. The predicted finite-horizon cost to be optimized is

$$J(\delta\mathbf{x}_k, \delta\mathbf{u}_{i|k}) = \delta\mathbf{x}_{N_p|k}^{\mathsf{T}} \mathbf{P} \delta\mathbf{x}_{N_p|k} + \sum_{i=0}^{N_p-1} \left(\delta\mathbf{x}_{i|k}^{\mathsf{T}} \mathbf{Q}_k \delta\mathbf{x}_{i|k} + \delta\mathbf{u}_{i|k}^{\mathsf{T}} \mathbf{R}_k \delta\mathbf{u}_{i|k} \right), \tag{5.7}$$

where $\mathbf{Q} = \mathbf{Q}^{\mathsf{T}} \geq 0$ is the state penalty matrix, $\mathbf{R} = \mathbf{R}^{\mathsf{T}} > 0$ is the control input penalty matrix, and $\mathbf{P} = \mathbf{P}^{\mathsf{T}} \geq 0$ is the terminal state penalty matrix. The penalty matrices are composed of weights on each of the control inputs and states

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}^{\phi} & & & \\ & \mathbf{Q}^{v} & & \\ & & \mathbf{Q}^{r} & \\ & & & \mathbf{Q}^{h} \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} \mathbf{P}^{\phi} & & & \\ & \mathbf{P}^{v} & & \\ & & \mathbf{P}^{r} & \\ & & & \mathbf{P}^{h} \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} R^{f} & \\ & \mathbf{R}^{m} \end{bmatrix}. \tag{5.8}$$

The stacked state penalty and control input error matrices over the prediction horizon are

$$\bar{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & & \\ & \ddots & & \\ & & \mathbf{Q} & \\ & & & \mathbf{P} \end{bmatrix}, \qquad \bar{\mathbf{R}} = \begin{bmatrix} \mathbf{R} & & \\ & \ddots & \\ & & \mathbf{R} \end{bmatrix}. \tag{5.9}$$

Substituting the definitions from (5.6) and (5.9) into (5.7),

$$\begin{aligned} J(\delta\mathbf{x}_k, \delta\mathbf{u}_{i|k}) &= \delta\boldsymbol{\chi}_k^{\mathsf{T}}\bar{\mathbf{Q}}\delta\boldsymbol{\chi}_k + \delta\boldsymbol{\mu}_k^{\mathsf{T}}\bar{\mathbf{R}}\delta\boldsymbol{\mu}_k + \delta\mathbf{x}_k^{\mathsf{T}}\mathbf{Q}\delta\mathbf{x}_k && (5.10)\\ &= \left(\boldsymbol{\mathcal{S}}_k\delta\boldsymbol{\mu}_k + \boldsymbol{\mathcal{M}}_k\delta\mathbf{x}_k\right)^{\mathsf{T}}\bar{\mathbf{Q}}\left(\boldsymbol{\mathcal{S}}_k\delta\boldsymbol{\mu}_k + \boldsymbol{\mathcal{M}}_k\delta\mathbf{x}_k\right) + \delta\boldsymbol{\mu}_k^{\mathsf{T}}\bar{\mathbf{R}}\delta\boldsymbol{\mu}_k + \delta\mathbf{x}_k^{\mathsf{T}}\mathbf{Q}\delta\mathbf{x}_k && (5.11)\\ &= \delta\boldsymbol{\mu}_k^{\mathsf{T}}\left(\boldsymbol{\mathcal{S}}_k^{\mathsf{T}}\bar{\mathbf{Q}}\boldsymbol{\mathcal{S}} + \bar{\mathbf{R}}\right)\delta\boldsymbol{\mu}_k + 2\delta\mathbf{x}_k^{\mathsf{T}}\boldsymbol{\mathcal{M}}_k^{\mathsf{T}}\bar{\mathbf{Q}}\boldsymbol{\mathcal{S}}_k\delta\boldsymbol{\mu}_k \\ &\qquad + \delta\mathbf{x}_k^{\mathsf{T}}\left(\boldsymbol{\mathcal{M}}_k^{\mathsf{T}}\bar{\mathbf{Q}}\boldsymbol{\mathcal{M}}_k + \mathbf{Q}\right)\delta\mathbf{x}_k && (5.12)\\ &= \delta\boldsymbol{\mu}_k^{\mathsf{T}}\mathbf{H}\delta\boldsymbol{\mu}_k + 2\mathbf{F}^{\mathsf{T}}\delta\boldsymbol{\mu}_k + \mathbf{c}_k, && (5.13) \end{aligned}$$

where

$$\begin{aligned} \mathbf{H}_k &= \boldsymbol{\mathcal{S}}_k^{\mathsf{T}}\bar{\mathbf{Q}}\boldsymbol{\mathcal{S}}_k + \bar{\mathbf{R}}, && (5.14)\\ \mathbf{F}_k &= \boldsymbol{\mathcal{S}}_k^{\mathsf{T}}\bar{\mathbf{Q}}\boldsymbol{\mathcal{M}}_k, && (5.15)\\ \mathbf{c}_k &= \delta\mathbf{x}_k^{\mathsf{T}}\left(\boldsymbol{\mathcal{M}}_k^{\mathsf{T}}\bar{\mathbf{Q}}\boldsymbol{\mathcal{M}}_k + \mathbf{Q}\right)\delta\mathbf{x}_k. && (5.16) \end{aligned}$$

The optimization problem can then be expressed as a QP,

$$\min_{\delta\boldsymbol{\mu}_k} \quad J(\delta\boldsymbol{\mu}_k) = \frac{1}{2}\delta\boldsymbol{\mu}_k^{\mathsf{T}}\mathbf{H}_k\delta\boldsymbol{\mu}_k + \mathbf{F}_k^{\mathsf{T}}\delta\boldsymbol{\mu}_k. \tag{5.17}$$

Note that the $\mathbf{c}_k$ term has been dropped from (5.17) since it is not a function of the optimization variables, $\delta\boldsymbol{\mu}_k$. Solving the QP gives the optimal control input sequence $\delta\boldsymbol{\mu}_k^{\star}$.

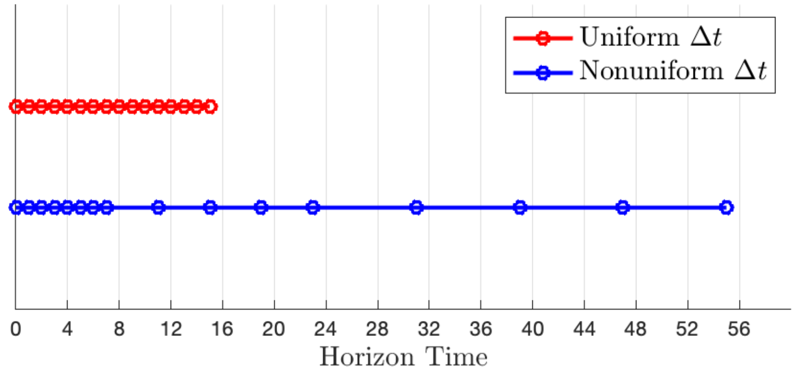


Figure 5.3: Total prediction horizon time using a traditional uniform timestep and a non-uniform timestep with $N_p = 16$.

5.3.1 Non-uniform Prediction Horizon Timestep

The process model given by (3.2) and (3.4) contains “fast” attitude dynamics and “slow” translational dynamics. To accurately predict the dynamics of this multi-timescale system, a large prediction horizon is necessary. To increase the time span of the prediction horizon without increasing the number of optimization variables, a non-uniformly spaced prediction horizon is implemented [44]. The short-term horizon sampling time is kept small to resolve the “fast” attitude dynamics. The long-term horizon sampling time is increased to ensure the “slow” translational dynamics are predicted sufficiently far into the future.

This effect is demonstrated by Figure 5.3. In this example, a total of $N = 16$ steps are used. By splitting the prediction horizon into 3 segments with increasing timestep intervals, the total horizon time is increased by a factor of 3.5 without increasing the number of prediction steps.

The penalty matrices **Q** and **R** are modified to account for the different sampling time at each segment in the horizon such that [45]

$$\mathbf{Q}_i = (\Delta t_i/\Delta t_1)\,\mathbf{Q}, \qquad \mathbf{R}_i = (\Delta t_i/\Delta t_1)\,\mathbf{R}, \tag{5.18}$$

where Δt_i is the sampling time of the individual horizon segment.

5.4 State and Input Constraints

Recall that the reference trajectory generated by the guidance does not enforce any state or control input constraints. This is by design since including constraints in the LQR formulation would require solving a QP instead of an analytic solution, increasing computational complexity [46]. An advantage of MPC is the ability to explicitly embed state and control input constraints in the optimization problem. The desired constraints must be defined in terms of the optimization variables and written as a linear inequality to be included in the QP. In this thesis, three types of constraints are considered.

5.4.1 Attitude Keep-in Zone

The presence of an external disturbance, such as a wind gust, can cause the vehicle to deviate from the planned reference trajectory, therefore the controller must be able to compensate within the desired attitude limits. The attitude of the helicopter is constrained using a keep-in zone [47], defined as

$$\mathbf{x}_b^{1^\mathsf{T}} \mathbf{C}_{ab}^\mathsf{T} \mathbf{y}_a^1 \geq \cos(\alpha) - \epsilon_1, \tag{5.19}$$

where ϵ_1 is a slack variable. The inequality given by (5.19) ensures that the angle made by $\mathbf{x}_b^1$ and $\mathbf{C}_{ab}^\mathsf{T}\mathbf{y}_a^1$ does not exceed α. By setting

$$\mathbf{x}_b^1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad \mathbf{y}_a^1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{5.20}$$

the roll and pitch angles are constrained simultaneously by α. The attitude keep-in zone constraint is visualized by Figure 5.4 where the attitude keep-in zone, represented by the green cone, constrains the $\underrightarrow{b}^{3}$ vector shown in blue.

To incorporate this constraint in the QP, (5.19) must be linearized and written in terms of the optimization variables, $\delta\boldsymbol{\mu}_k$ and ϵ_1. First, (5.19) is linearized using (3.8) and (3.18),

$$\mathbf{x}_b^{1^\mathsf{T}} \delta\mathbf{C}^\mathsf{T}\mathbf{C}_{ar}^\mathsf{T}\mathbf{y}_a^1 \geq \cos(\alpha) - \epsilon_1 \tag{5.21}$$

$$\mathbf{x}_b^{1^\mathsf{T}} \left(\mathbf{1} - \delta\boldsymbol{\xi}^{\phi^\times}\right) \mathbf{C}_{ar}^\mathsf{T}\mathbf{y}_a^1 \geq \cos(\alpha) - \epsilon_1 \tag{5.22}$$

$$\mathbf{x}_b^{1^\mathsf{T}} \left(\mathbf{C}_{ar}^\mathsf{T}\mathbf{y}_a^1\right)^\times \delta\boldsymbol{\xi}^\phi \geq \cos(\alpha) - \mathbf{x}_b^{1^\mathsf{T}}\mathbf{C}_{ar}^\mathsf{T}\mathbf{y}_a^1 - \epsilon_1. \tag{5.23}$$

From the stacked state transition matrix (5.6), the i^{th} predicted attitude in the horizon can be written as

$$\delta\boldsymbol{\xi}_{k|i}^\phi = \mathbf{P}_i^a \left(\boldsymbol{\mathcal{S}}_k \delta\boldsymbol{\mu}_k + \boldsymbol{\mathcal{M}}_k \delta\mathbf{x}_k\right), \tag{5.24}$$

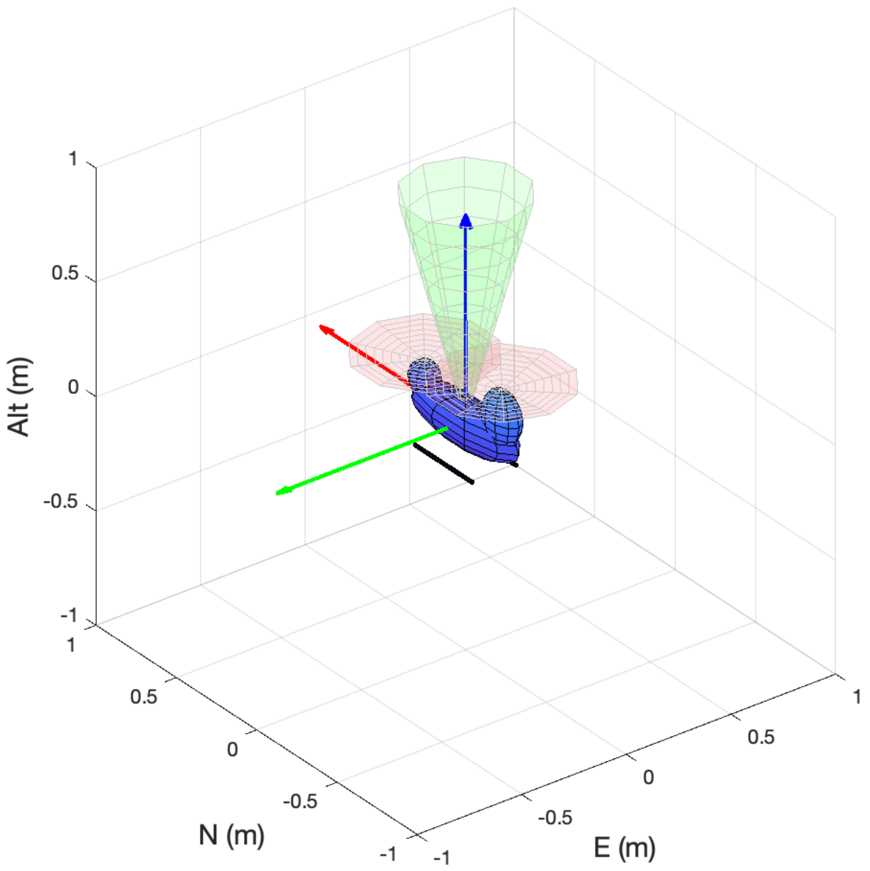


Figure 5.4: Visualization of attitude keep-in zone constraint.

where $\mathbf{P}_i^a$ is a projection matrix. Substituting (5.24) into (5.23) results in

$$\mathbf{x}_b^{1^\mathsf{T}} \left(\mathbf{C}_{ar_{k+i}}^\mathsf{T} \mathbf{y}_a^1\right)^\times \mathbf{P}_i^a \left(\boldsymbol{\mathcal{S}}_k \delta\boldsymbol{\mu}_k + \boldsymbol{\mathcal{M}}_k \delta\mathbf{x}_k\right) \geq \cos(\alpha) - \mathbf{x}_b^{1^\mathsf{T}} \mathbf{C}_{ar_{k+i}}^\mathsf{T} \mathbf{y}_a^1 - \epsilon_1. \tag{5.25}$$

By rearranging (5.25), the linearized keep-in zone constraint can be written in terms of the optimization variables as

$$\begin{aligned}\begin{bmatrix} -\mathbf{x}_b^{1^\mathsf{T}} \left(\mathbf{C}_{ar_{k+i}}^\mathsf{T} \mathbf{y}_a^1\right)^\times \mathbf{P}_i^a \boldsymbol{\mathcal{S}}_k & -\mathbf{1} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{\mu}_k \\ \epsilon_1 \end{bmatrix} \leq & \begin{bmatrix} -\cos(\alpha) + \mathbf{x}_b^{1^\mathsf{T}} \mathbf{C}_{ar_{k+i}}^\mathsf{T} \mathbf{y}_a^1 \end{bmatrix} \\ & + \begin{bmatrix} \mathbf{x}_b^{1^\mathsf{T}} \left(\mathbf{C}_{ar_{k+i}}^\mathsf{T} \mathbf{y}_a^1\right)^\times \mathbf{P}_i^a \boldsymbol{\mathcal{M}}_k \end{bmatrix} \delta\mathbf{x}_k. \end{aligned} \tag{5.26}$$

5.4.2 Attitude Error

Due to the linearization performed to formulate the attitude keep-in zone constraint, (5.26) is only valid for small $\delta\boldsymbol{\xi}^\phi$. Although the controller attempts to drive $\delta\boldsymbol{\xi}^\phi$ to zero, it does not ensure that the size of the attitude error is kept below a desired threshold. Therefore, an additional constraint is imposed on the size of the attitude error to ensure the nonlinear keep-in zone constraint given by (5.23) is respected. While the ℓ_2-norm is a logical choice to constrain the size of the vector $\delta\boldsymbol{\xi}^\phi$, the ℓ_1-norm is used here for two reasons. Firstly, it provides a conservative size constraint since $\|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1$ [48]. Secondly, unlike the ℓ_2-norm, the ℓ_1-norm can be easily written as a series of linear inequalities, which is required to incorporate the constraint in the QP.

The ℓ_1-norm attitude error constraint is written as

$$\left\|\delta\boldsymbol{\xi}^\phi\right\|_1 \leq \gamma + \epsilon_2, \tag{5.27}$$

where γ is a scalar that limits the size of $\delta\boldsymbol{\xi}$, and ϵ_2 is a slack variable. Equation (5.27) is equivalent to the linear inequalities [14]

$$\sum_{i=1}^{3} z_i \leq \gamma + \epsilon_2, \tag{5.28}$$

$$-z_i \leq \delta\boldsymbol{\xi}_i^\phi \leq z_i, \qquad i = 1, 2, 3 \tag{5.29}$$

where z_i, $i = 1, 2, 3$ are additional optimization variables. To incorporate (5.28) and (5.29) into the QP, they must be expressed in terms of the optimization variables $\delta\boldsymbol{\mu}_k$, $\mathbf{z}$, and ϵ_2. Note that (5.28) already satisfies this requirement, however (5.29) must be modified.

Substituting (5.24) into (5.29) results in

$$-\mathbf{z}_i \leq \mathbf{P}_i^a \left(\boldsymbol{\mathcal{S}}_k \delta\boldsymbol{\mu}_k + \boldsymbol{\mathcal{M}}_k \delta\mathbf{x}_k\right) \leq \mathbf{z}_i. \tag{5.30}$$

The compound inequality given by (5.30) can be split into two separate inequalities,

$$-\mathbf{z}_i \leq \mathbf{P}_i^a \left(\boldsymbol{\mathcal{S}}_k \delta\boldsymbol{\mu}_k + \boldsymbol{\mathcal{M}}_k \delta\mathbf{x}_k\right), \tag{5.31}$$

$$-\mathbf{z}_i \leq -\mathbf{P}_i^a \left(\boldsymbol{\mathcal{S}}_k \delta\boldsymbol{\mu}_k + \boldsymbol{\mathcal{M}}_k \delta\mathbf{x}_k\right). \tag{5.32}$$

Therefore, the ℓ_1-norm constraint can be written in matrix form by combining and rearranging (5.28), (5.31), and (5.32),

$$\begin{bmatrix} \mathbf{0} & \mathbf{1}^{1\times 3} & -1 \\ \mathbf{P}_i^a \boldsymbol{\mathcal{S}}_k & -\mathbf{1} & 0 \\ -\mathbf{P}_i^a \boldsymbol{\mathcal{S}}_k & -\mathbf{1} & 0 \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{\mu}_k \\ \mathbf{z} \\ \epsilon_2 \end{bmatrix} \leq \begin{bmatrix} \gamma \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} 0 \\ -\mathbf{P}_i^a \boldsymbol{\mathcal{M}}_k \\ \mathbf{P}_i^a \boldsymbol{\mathcal{M}}_k \end{bmatrix} \delta\mathbf{x}_k. \tag{5.33}$$

5.4.3 Control Inputs

Control input constraints are defined by first setting maximum and minimum total force and torque control efforts,

$$\mathbf{u}_{\max} = \begin{bmatrix} f_{\max} \\ \mathbf{m}_{b_{\max}} \end{bmatrix}, \qquad \mathbf{u}_{\max} = \begin{bmatrix} f_{\min} \\ \mathbf{m}_{b_{\min}} \end{bmatrix}. \tag{5.34}$$

The MPC algorithm operates on the incremental control input, $\delta\mathbf{u}_k$, therefore the control input constraints are defined by subtracting the reference control input, $\mathbf{u}_k^r$, from the maximum and minimum control efforts,

$$\delta\mathbf{u}_{\max_k} = \mathbf{u}_{\max} - \mathbf{u}_k^r, \tag{5.35}$$

$$\delta\mathbf{u}_{\min_k} = \mathbf{u}_{\min} - \mathbf{u}_k^r. \tag{5.36}$$

The constraints on the control inputs are therefore,

$$\delta\mathbf{u}_{\min_k} \leq \delta\mathbf{u}_k \leq \delta\mathbf{u}_{\max_k}. \tag{5.37}$$

Equation (5.37) can be written in terms of the optimization variables $\delta\boldsymbol{\mu}_k$ as,

$$\delta\mathbf{u}_{\min_{i|k}} \leq \mathbf{P}_i^c \delta\boldsymbol{\mu}_k \leq \delta\mathbf{u}_{\max_{i|k}}, \tag{5.38}$$

where $\mathbf{P}_i^c$ is a projection matrix. Note that this is not the same as the projection matrix, $\mathbf{P}_i^a$, from in Sections 5.4.1 and 5.4.2. The compound inequality from (5.38) can be written as two separate inequalities as

$$-\mathbf{P}_i^c \delta\boldsymbol{\mu}_k \leq -\delta\mathbf{u}_{\min_{i|k}}, \tag{5.39}$$

$$\mathbf{P}_i^c \delta\boldsymbol{\mu}_k \leq \delta\mathbf{u}_{\max_{i|k}}. \tag{5.40}$$

By rearranging (5.39) and (5.40), the control input constraints are written in matrix form,

$$\begin{bmatrix} \mathbf{P}_i^c \\ -\mathbf{P}_i^c \end{bmatrix} \delta\boldsymbol{\mu}_k \leq \begin{bmatrix} \delta\mathbf{u}_{\max_{i|k}} \\ -\delta\mathbf{u}_{\min_{i|k}} \end{bmatrix}. \tag{5.41}$$

5.4.4 Formulating the Constrained QP

To incorporate the formulated constraints into the QP, the cost function from (5.7) must be modified to include penalties on the slack variables ϵ_1 and ϵ_2,

$$J(\delta\mathbf{x}_k, \delta\mathbf{u}_{i|k}, \boldsymbol{\epsilon}) = \boldsymbol{\epsilon}^{\mathsf{T}}\boldsymbol{\eta}\boldsymbol{\epsilon} + \delta\mathbf{x}_{N_p|k}^{\mathsf{T}}\mathbf{P}\delta\mathbf{x}_{N_p|k} + \sum_{i=0}^{N_p-1}\left(\delta\mathbf{x}_{i|k}^{\mathsf{T}}\mathbf{Q}_k\delta\mathbf{x}_{i|k} + \delta\mathbf{u}_{i|k}^{\mathsf{T}}\mathbf{R}_k\delta\mathbf{u}_{i|k}\right), \tag{5.42}$$

where $\boldsymbol{\eta} = \operatorname{diag}(\eta_1, \eta_2)$ is a penalty on the slack variables. The constraints (5.26), (5.33), and (5.41) are combined to form matrices than can be included in the QP. First, the control input vector is augmented to include the additional optimization variables from the constraints,

$$\delta\tilde{\boldsymbol{\mu}}_k = \begin{bmatrix} \delta\boldsymbol{\mu}_k \\ \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_{N_c} \\ \epsilon_{1|1} \\ \vdots \\ \epsilon_{1|N_c} \\ \epsilon_{2|1} \\ \vdots \\ \epsilon_{2|N_c} \end{bmatrix}, \tag{5.43}$$

where $N_c \leq N_p$ is the constraint horizon. The length of the constraint horizon can be shrunk to reduce the size of the QP. The matrices $\bar{\mathbf{R}}$ and $\boldsymbol{\mathcal{S}}$ are also augmented to incorporate the additional optimization variables,

$$\tilde{\mathbf{R}} = \begin{bmatrix} \bar{\mathbf{R}} & & & \\ & \mathbf{0} & & \\ & & \eta_1 \mathbf{1} & \\ & & & \eta_2 \mathbf{1} \end{bmatrix}, \qquad \tilde{\boldsymbol{\mathcal{S}}} = \begin{bmatrix} \boldsymbol{\mathcal{S}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{5.44}$$

Therefore, the $\mathbf{H}_k$ and $\mathbf{F}_k$ matrices become

$$\tilde{\mathbf{H}}_k = \tilde{\boldsymbol{\mathcal{S}}}_k^\mathsf{T} \bar{\mathbf{Q}} \tilde{\boldsymbol{\mathcal{S}}}_k + \tilde{\mathbf{R}}, \tag{5.45}$$

$$\tilde{\mathbf{F}}_k = \tilde{\boldsymbol{\mathcal{S}}}_k^\mathsf{T} \bar{\mathbf{Q}} \boldsymbol{\mathcal{M}}_k \delta \mathbf{x}_k. \tag{5.46}$$

Next, the constraints are combined and written in terms of $\delta \bar{\boldsymbol{\mu}}_k$,

$$\mathbf{G}_k = \left[\begin{array}{cccccccccc}
-\mathbf{x}_b^{1^\mathsf{T}} \left(\mathbf{C}_{ar_{k+1}}^\mathsf{T} \mathbf{y}_a^1 \right)^\times \mathbf{P}_1^a \boldsymbol{\mathcal{S}}_k & \mathbf{0} & & & -1 & & & 0 & & \\
\vdots & & \ddots & & & \ddots & & & \ddots & \\
-\mathbf{x}_b^{1^\mathsf{T}} \left(\mathbf{C}_{ar_{k+N_c+1}}^\mathsf{T} \mathbf{y}_a^1 \right)^\times \mathbf{P}_{N_c}^a \boldsymbol{\mathcal{S}}_k & & & \mathbf{0} & & & -1 & & & 0 \\
\mathbf{0} & \mathbf{1}^{1\times 3} & & & 0 & & & -1 & & \\
\vdots & & \ddots & & & \ddots & & & \ddots & \\
\mathbf{0} & & & \mathbf{1}^{1\times 3} & & & 0 & & & -1 \\
\mathbf{P}_1^a \boldsymbol{\mathcal{S}}_k & -\mathbf{1} & & & 0 & & & 0 & & \\
\vdots & & \ddots & & & \ddots & & & \ddots & \\
\mathbf{P}_{N_c}^a \boldsymbol{\mathcal{S}}_k & & & -\mathbf{1} & & & 0 & & & 0 \\
-\mathbf{P}_1^a \boldsymbol{\mathcal{S}}_k & -\mathbf{1} & & & 0 & & & 0 & & \\
\vdots & & \ddots & & & \ddots & & & \ddots & \\
-\mathbf{P}_{N_c}^a \boldsymbol{\mathcal{S}}_k & & & -\mathbf{1} & & & 0 & & & 0 \\
\mathbf{P}_1^c & \mathbf{0} & & & 0 & & & 0 & & \\
\vdots & & \ddots & & & \ddots & & & \ddots & \\
\mathbf{P}_{N_c}^c & & & \mathbf{0} & & & 0 & & & 0 \\
-\mathbf{P}_1^c & \mathbf{0} & & & 0 & & & 0 & & \\
\vdots & & \ddots & & & \ddots & & & \ddots & \\
-\mathbf{P}_{N_c}^c & & & \mathbf{0} & & & 0 & & & 0
\end{array} \right], \tag{5.47}$$

$$\mathbf{W}_k = \begin{bmatrix} -\cos(\alpha) + \mathbf{x}_b^{1^\mathsf{T}} \mathbf{C}^\mathsf{T}_{ar_{k+1}} \mathbf{y}_a^1 \\ \vdots \\ -\cos(\alpha) + \mathbf{x}_b^{1^\mathsf{T}} \mathbf{C}^\mathsf{T}_{ar_{k+N_c+1}} \mathbf{y}_a^1 \\ \gamma \\ \vdots \\ \gamma \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \delta \mathbf{u}_{\max_{1|k}} \\ \vdots \\ \delta \mathbf{u}_{\max_{N_c|k}} \\ -\delta \mathbf{u}_{\min_{1|k}} \\ \vdots \\ -\delta \mathbf{u}_{\min_{N_c|k}} \end{bmatrix}, \quad \mathbf{T}_k \quad = \begin{bmatrix} \mathbf{x}_b^{1^\mathsf{T}} \left(\mathbf{C}^\mathsf{T}_{ar_{k+1}} \mathbf{y}_a^1 \right)^\times \mathbf{P}_1^a \boldsymbol{\mathcal{M}}_k \\ \vdots \\ \mathbf{x}_b^{1^\mathsf{T}} \left(\mathbf{C}^\mathsf{T}_{ar_{k+N_c+1}} \mathbf{y}_a^1 \right)^\times \mathbf{P}_{N_c}^a \boldsymbol{\mathcal{M}}_k \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -\mathbf{P}_1^a \boldsymbol{\mathcal{M}}_k \\ \vdots \\ -\mathbf{P}_{N_c}^a \boldsymbol{\mathcal{M}}_k \\ \mathbf{P}_1^a \boldsymbol{\mathcal{M}}_k \\ \vdots \\ \mathbf{P}_{N_c}^a \boldsymbol{\mathcal{M}}_k \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}. \tag{5.48}$$

The constraint matrices are added to the QP from (5.17) to give the constrained QP,

$$\begin{aligned} \min_{\delta \tilde{\boldsymbol{\mu}}_k} \quad & J(\delta \tilde{\boldsymbol{\mu}}_k) = \frac{1}{2} \delta \tilde{\boldsymbol{\mu}}_k^\mathsf{T} \tilde{\mathbf{H}}_k \delta \tilde{\boldsymbol{\mu}}_k + \tilde{\mathbf{F}}_k^\mathsf{T} \delta \tilde{\boldsymbol{\mu}}_k. \\ \text{s.t.} \quad & \mathbf{G}_k \delta \tilde{\boldsymbol{\mu}}_k \leq \mathbf{W}_k + \mathbf{T}_k \delta \mathbf{x}_k, \end{aligned} \tag{5.49}$$

5.5 MPC Solution

5.5.1 Solving the Constrained QP

The constrained QP given by (5.49) is solved yielding the optimal incremental control input sequence

$$\delta \boldsymbol{\mu}_k^\star = \begin{bmatrix} \delta \mathbf{u}_{0|k}^\star \\ \delta \mathbf{u}_{1|k}^\star \\ \vdots \\ \delta \mathbf{u}_{N-1|k}^\star \end{bmatrix}. \tag{5.50}$$

Several commercial and open-source solvers exist which can efficiently solve QPs. For the simulations performed in this thesis, MATLAB's internal QP solver, `quadprog()`, is used.

5.5.2 Adding the Feedforward Terms

The first control input in the optimal input sequence given by (5.50) is extracted yielding

$$\delta\mathbf{u}^{\star}_{0|k} = \begin{bmatrix} \delta f^{\star}_{0|k} \\ \delta\mathbf{m}^{\star}_{0|k} \end{bmatrix}. \tag{5.51}$$

Finally, the total control input, $\mathbf{u}_k$, is obtained by adding the feedforward input, $\mathbf{u}^r_k$, given by (4.70) to the incremental control input

$$\mathbf{u}_k = \begin{bmatrix} f_k \\ \mathbf{m}_{b_k} \end{bmatrix} = \begin{bmatrix} f^r_k + \delta f^{\star}_{0|k} \\ \delta\mathbf{C}^{\mathsf{T}}_k \mathbf{m}^r_{r_k} + \delta\mathbf{m}^{\star}_{0|k} \end{bmatrix}. \tag{5.52}$$

Note that the reference torque command, $\mathbf{m}^r_{r_k}$, is resolved in $\mathcal{F}_r$. Before being combined with $\delta\mathbf{m}_k$, it must be resolved in $\mathcal{F}_b$ through multiplication by $\delta\mathbf{C}^{\mathsf{T}}_k$.

5.6 Disturbance Estimation

To improve guidance, an estimate of the disturbances acting on the vehicle, particularly a near-constant wind, is needed. Assuming the states are accurately estimated, then the disturbance at the previous timestep can be approximated as

$$\mathbf{d}_{k-1} = \mathbf{x}_k - \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}), \tag{5.53}$$

where $\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$ represents modeled dynamics from (3.2) and (3.4). A simple moving average filter is used to estimate the current disturbance. When the trajectory is replanned, the controller passes the disturbance estimate to the LQR guidance to improve the accuracy of the new reference trajectory. Without this simple disturbance estimator in the guidance, the tracking performance is poor.

5.7 Simulation Implementation

5.7.1 Actuator Mixer

Before being applied to the system, an actuator mixer is used to map the control input commands from the MPC, given by (5.52), to front and rear rotor force components, $\mathbf{f}^1_{b_k}$ and

$\mathbf{f}^2_{b_k}$, respectively. As is common for tandem-rotor helicopter control [3], it is assumed that the $\underset{\rightarrow}{b}^{1}$ rotor force component is strictly used for pitch trimming. Therefore, only the $\underset{\rightarrow}{b}^{2}$, and $\underset{\rightarrow}{b}^{3}$ rotor force components are used for control. The mapping from the control inputs to the rotor force inputs is

$$\begin{bmatrix} f^1_{b_{2k}} \\ f^1_{b_{3k}} \\ f^2_{b_{2k}} \\ f^2_{b_{3k}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ -r^{1z}_{b_3} & r^{1z}_{b_2} & -r^{2z}_{b_3} & r^{2z}_{b_2} \\ 0 & -r^{1z}_{b_1} & 0 & -r^{2z}_{b_1} \\ r^{1z}_{b_1} & 0 & r^{2z}_{b_1} & 0 \end{bmatrix}^{-1} \mathbf{u}_k, \tag{5.54}$$

where $\mathbf{r}^{1z}_b = \begin{bmatrix} r^{1z}_{b_1} & r^{1z}_{b_2} & r^{1z}_{b_3} \end{bmatrix}^{\mathsf{T}}$ and $\mathbf{r}^{2z}_b = \begin{bmatrix} r^{2z}_{b_1} & r^{2z}_{b_2} & r^{2z}_{b_3} \end{bmatrix}^{\mathsf{T}}$ are the positions of the front and rear rotors relative to the vehicle's center of mass, respectively.

5.7.2 Dynamics Propagation

For all simulations, the vehicle dynamics given by (3.1) to (3.4) are propagated using the forward-Euler method as

$$\mathbf{C}_{ab_{k+1}} = \mathbf{C}_{ab_k} \exp\left(\left(h\boldsymbol{\omega}^{ba}_{b_k}\right)^{\times}\right), \tag{5.55}$$

$$\mathbf{r}^{zw}_{a_{k+1}} = \mathbf{r}^{zw}_{a_k} + h\mathbf{v}^{zw/a}_{a_k}, \tag{5.56}$$

$$\mathbf{v}^{zw/a}_{a_{k+1}} = \mathbf{v}^{zw/a}_{a_k} + h\frac{1}{m_{\mathcal{B}}}\mathbf{f}^{\mathcal{B}z}_{a_k}, \tag{5.57}$$

$$\boldsymbol{\omega}^{ba}_{b_{k+1}} = \boldsymbol{\omega}^{ba}_{b_k} + h\mathbf{J}^{\mathcal{B}z^{-1}}_b \left(\mathbf{m}^{\mathcal{B}z}_{b_k} - \boldsymbol{\omega}^{ba^{\times}}_{b_k}\mathbf{J}^{\mathcal{B}z}_b\boldsymbol{\omega}^{ba}_{b_k}\right), \tag{5.58}$$

where h is the timestep.

5.8 Simulation Results

5.8.1 Simulated Landing Trajectory

A series of simulations are performed using the tandem-rotor helicopter parameters shown in Table 5.1 and the equations of motion from (3.2) and (3.4). The simulations are meant to replicate a final approach to a landing site. For each simulation, the target position and velocity are

$$\mathbf{r}^{z_f w}_a = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}} \text{ m},$$

$$\mathbf{v}^{z_f w/a}_a = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}} \text{ m} \cdot \text{s}^{-1}.$$

The initial states are

$$\begin{aligned}
\mathbf{C}_{ab_0} &= \mathbf{1}, \\
\mathbf{r}_a^{z_0 w} &= \begin{bmatrix} -30 & -5 & -20 \end{bmatrix}^{\mathsf{T}} \text{ m}, \\
\mathbf{v}_a^{z_0 w/a} &= \begin{bmatrix} 5 & 0 & -0.5 \end{bmatrix}^{\mathsf{T}} \text{ m} \cdot \text{s}^{-1}, \\
\boldsymbol{\omega}_b^{b_0 a} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}} \text{ rad} \cdot \text{s}^{-1},
\end{aligned}$$

and the reference heading is $\psi^r = 0$ rad. The nominal wind speed is set to

$$\mathbf{v}_a^{sw/a} = \begin{bmatrix} 0 & -5 & 0 \end{bmatrix}^{\mathsf{T}} \text{ m} \cdot \text{s}^{-1},$$

and wind gusts, $\mathbf{v}_a^{gw/a}$, are generated using the Dryden model shown in Section 3.8 with $W_{20} = 12 \text{ m} \cdot \text{s}^{-1}$.

The guidance and control loop is run at 50 Hz. The parameters used by the guidance LQR algorithm to generate the reference trajectory are shown in Table 5.2, while the parameters used by the MPC algorithm are shown in Table 5.3. Note that while the MPC prediction horizon is $N_p = 20$, the control horizon and constraint horizon are both limited to $N_u = N_c = 10$ to reduce computational complexity [7]. A non-uniform prediction horizon is created by splitting the horizon into three distinct timestep intervals. The reference trajectory is replanned if the ℓ_1-norm constraint is active for longer than 0.4 s. In practice a navigation loop would provide state estimates, $\hat{\mathbf{x}}_k$, to the guidance and control algorithms. For these simulations, it is assumed the state estimates are accurate such that $\mathbf{x}_k = \hat{\mathbf{x}}_k$.

Table 5.1: Tandem-rotor Helicopter Parameters Used in Simulation

Parameter	Value	Units
$m_{\mathcal{B}}$	218	kg
$\mathbf{J}_b^{\mathcal{B}z}$	$\text{diag}(26.8, 97.6, 87.2)$	$\text{kg} \cdot \text{m}^2$
$\mathbf{r}_b^{1z}$	$\begin{bmatrix} 1.045 & 0 & -0.514 \end{bmatrix}^{\mathsf{T}}$	m
$\mathbf{r}_b^{2z}$	$\begin{bmatrix} -0.937 & 0 & -0.686 \end{bmatrix}^{\mathsf{T}}$	m

The results from a single simulation are highlighted to demonstrate advantages of the proposed guidance and control structure. The helicopter path from the starting position to the target is shown in Figure 5.5. Dashed lines show reference trajectories, while solid lines show actual trajectories. Changes in line color indicate a replanned trajectory, and occur when the attitude error exceeds the specified threshold. Initial tracking of the reference trajectory is good when the disturbance estimate from the controller is most accurate. As

Table 5.2: Guidance LQR Parameters Used in Simulation

Parameter	Value	Units
dt	0.02	s
$\mathbf{Q}^{\phi}$	$\text{diag}(1\text{e}3, 1\text{e}3, 1\text{e}6)$	-
$\mathbf{Q}^{v}$	$\text{diag}(1\text{e}1, 1\text{e}1, 1\text{e}1)$	-
$\mathbf{Q}^{r}$	$\text{diag}(1\text{e}2, 1\text{e}2, 1\text{e}2)$	-
$\mathbf{Q}^{h}$	$\text{diag}(1\text{e}2, 1\text{e}2, 1\text{e}2)$	-
$\mathbf{Q}^{i}$	$\text{diag}(1\text{e}2, 1\text{e}2, 1\text{e}2)$	-
$\mathbf{Q}$	$\text{diag}(\mathbf{Q}^{\phi}, \mathbf{Q}^{v}, \mathbf{Q}^{r}, \mathbf{Q}^{h}, \mathbf{Q}^{i})$	-
$\mathbf{S}$	$10 \cdot \mathbf{Q}$	-
$\mathbf{R}$	$\text{diag}(1, 1, 1, 1)$	-
c_1	1	-
c_2	1	-

the disturbance evolves, the tracking performance suffers until the trajectory is replanned with an updated disturbance estimate.

The attitude keep-in zone and ℓ_1-norm attitude error constraints are visualized in Figure 5.6a. The attitude keep-in zone constraints are respected if the solid colored lines remain above the dotted black line. Conversely, the ℓ_1-norm attitude error constraint is respected if the solid colored line remains blow the dotted black line. The linearized keep-in zone and attitude error constraints are respected throughout the entire simulation. During periods with larger attitude error, the linearized and nonlinear keep-in zone constraints diverge slightly. In some cases, the nonlinear constraint is marginally violated. This behavior is limited by the attitude error constraint, which maintains the validity of the keep-in zone by limiting the size of $\delta\boldsymbol{\xi}^{\phi}$. The divergence of the nonlinear and linearized keep-in zones can be further limited by reducing γ. However, overly restricting the attitude error results in diminished tracking performance in the presence of large disturbances. It can be seen that the trajectory is replanned when the ℓ_1-norm attitude error constraint becomes active. Once the trajectory is replanned, the attitude error immediately drops to zero since the new reference trajectory is planned from the current state.

The control input constraints are shown in Figure 5.6b. The force and torque inputs are effectively bound by the imposed constraints. When the reference trajectory is replanned, there is a step change in the state error resulting in discontinuous control inputs. In practice, the control inputs would be passed through a low-pass filter before being sent to the actuators to ensure smooth commands.

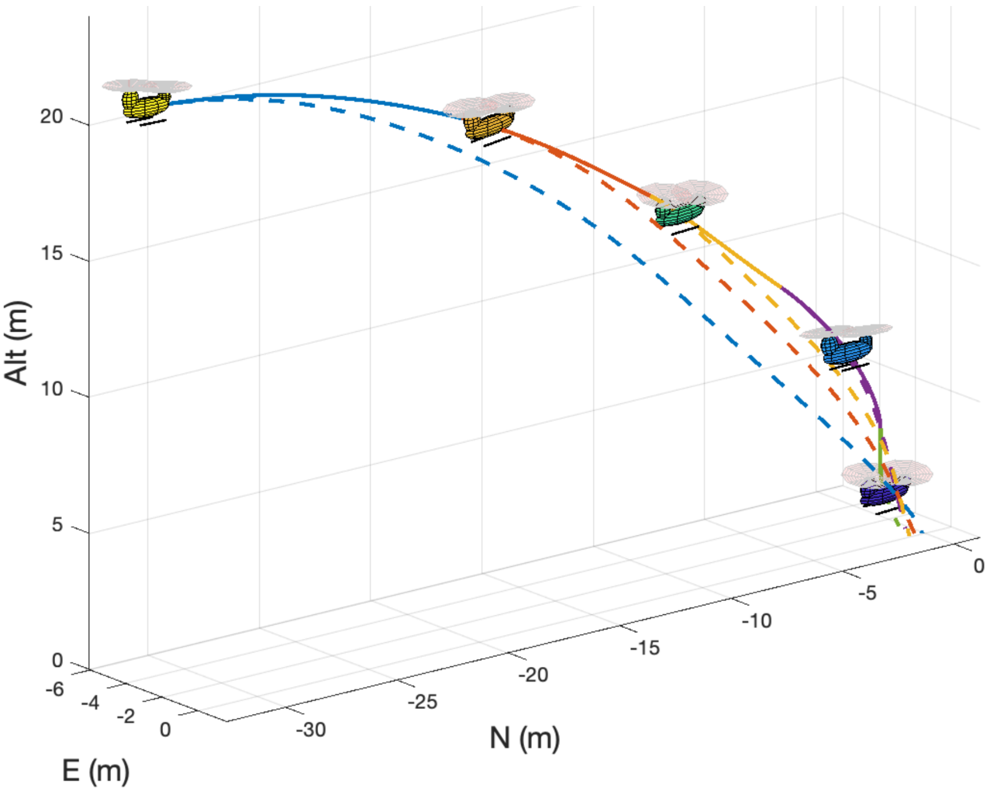
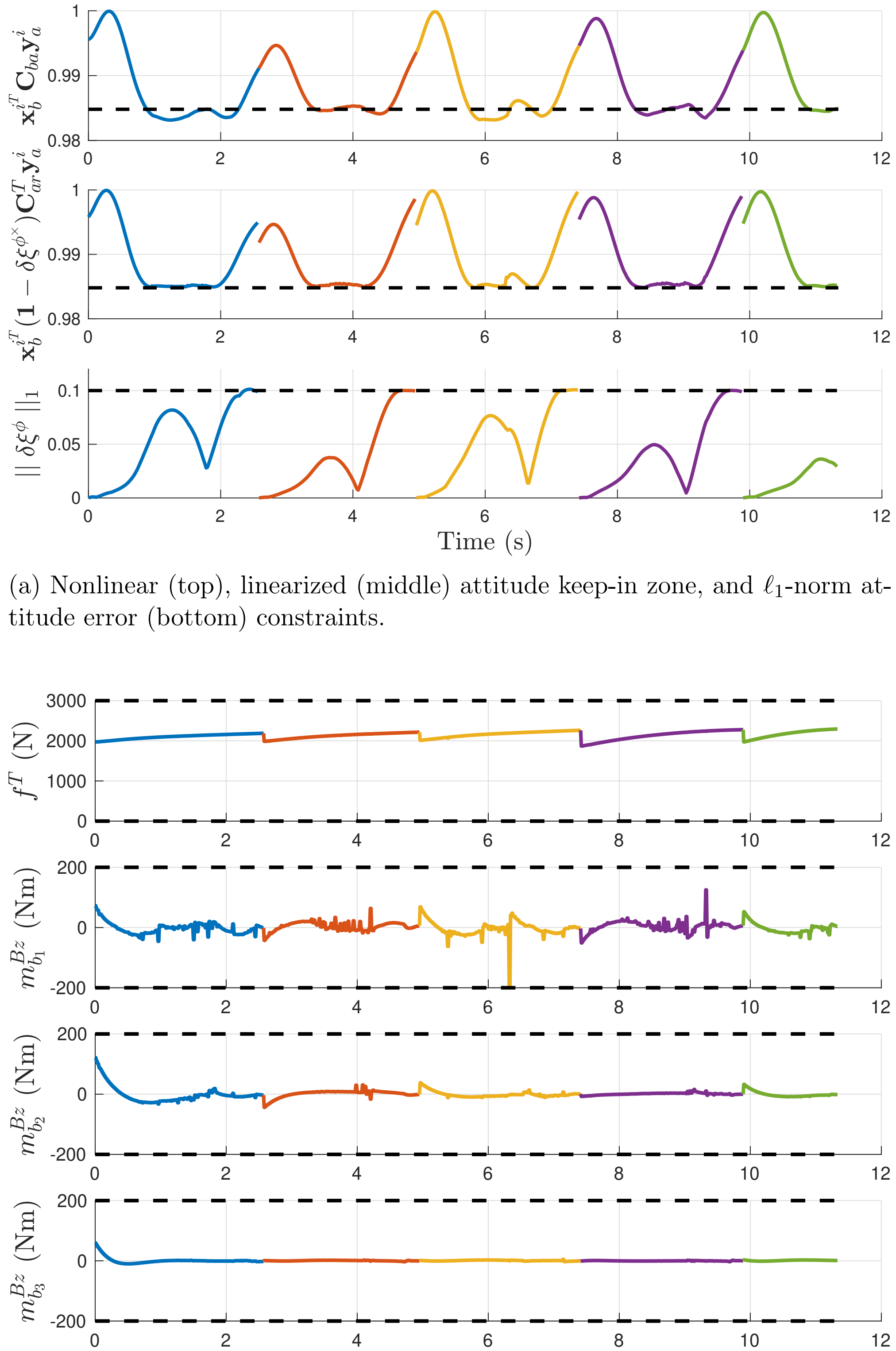


Figure 5.5: Helicopter's progress along the planned path during a single simulation. Dashed lines represent desired trajectory, solid lines represent actual trajectory. Changes in colour represent replanned trajectories.



(a) Nonlinear (top), linearized (middle) attitude keep-in zone, and ℓ_1-norm attitude error (bottom) constraints.

(b) Control input constraints.

Figure 5.6: Attitude and control input constraint performance over a single simulation. Changes in colour represent replanned trajectories.

Table 5.3: MPC Parameters Used in Simulation

Parameter	Value	Units
dt	0.02	s
N_p	48	-
N_u	10	-
N_c	10	-
$\mathbf{Q}^{\phi}$	$\text{diag}(1e3, 1e3, 1e6)$	-
$\mathbf{Q}^{v}$	$\text{diag}(1e1, 1e1, 1e1)$	-
$\mathbf{Q}^{r}$	$\text{diag}(1e2, 1e2, 1e2)$	-
$\mathbf{Q}^{h}$	$\text{diag}(1e2, 1e2, 1e2)$	-
$\mathbf{Q}$	$\text{diag}(\mathbf{Q}^{\phi}, \mathbf{Q}^{v}, \mathbf{Q}^{r}, \mathbf{Q}^{h})$	-
$\mathbf{P}$	$10 \cdot \mathbf{Q}$	-
$\mathbf{R}$	$\text{diag}(1, 1, 1, 1)$	-
η_1	$1e24$	-
η_2	$1e24$	-
Δt_1	0.04	s
Δt_2	0.16	s
Δt_3	0.64	s
N_{p_1}	24	-
N_{p_2}	12	-
N_{p_3}	12	-
$\mathbf{u}_{\text{min}}$	$\begin{bmatrix} 0 & -200 & -200 & -200 \end{bmatrix}^{\mathsf{T}}$	N, N · m
$\mathbf{u}_{\text{max}}$	$\begin{bmatrix} 3000 & 200 & 200 & 200 \end{bmatrix}^{\mathsf{T}}$	N, N · m
α	10	deg
γ	0.1	rad

5.8.2 Monte-Carlo Simulations

Monte-Carlo simulations are performed to test the robustness of the proposed guidance and control structure to initial conditions, environmental disturbances, and model uncertainty. The initial state of the helicopter is randomized such that

$$\begin{aligned}
\boldsymbol{\phi}_0 &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}} + \mathbf{w}_1 \text{ rad}, \\
\mathbf{r}_a^{z_0 w} &= \begin{bmatrix} -30 & -5 & -20 \end{bmatrix}^{\mathsf{T}} + \mathbf{w}_2 \text{ m}, \\
\mathbf{v}_a^{z_0 w/a} &= \begin{bmatrix} 5 & 0 & 0.5 \end{bmatrix}^{\mathsf{T}} + \mathbf{w}_3 \text{ m} \cdot \text{s}^{-1}, \\
\boldsymbol{\omega}_b^{b_0 a} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}} + \mathbf{w}_4 \text{ rad} \cdot \text{s}^{-1},
\end{aligned}$$

where $\boldsymbol{\phi}_0 = \log_{SO(3)}(\mathbf{C}_{ab_0})$, $\mathbf{w}_1 \sim \mathcal{N}(\mathbf{0}, \sigma_\phi^2 \mathbf{1})$, $\mathbf{w}_2 \sim \mathcal{N}(\mathbf{0}, \sigma_r^2 \mathbf{1})$, $\mathbf{w}_3 \sim \mathcal{N}(\mathbf{0}, \sigma_v^2 \mathbf{1})$, and $\mathbf{w}_4 \sim \mathcal{N}(\mathbf{0}, \sigma_\omega^2 \mathbf{1})$, where $\mathbf{x} \sim \mathcal{N}(\mathbf{a}, \mathbf{B})$ indicates that $\mathbf{x}$ is normally distributed with mean $\mathbf{a}$ and variance $\mathbf{B}$.

The nominal wind condition and Dryden gust model are randomized such that

$$\begin{aligned} \mathbf{v}_a^{sw/a} &= \begin{bmatrix} 0 & -3 & 0 \end{bmatrix}^\mathsf{T} + \mathbf{w}_5 \ \mathrm{m} \cdot \mathrm{s}^{-1}, \\ W_{20} &= 10 + w_6 \ \mathrm{m} \cdot \mathrm{s}^{-1}, \end{aligned}$$

where $\mathbf{w}_5 \sim \mathcal{N}(0, \sigma_w^2 \mathbf{1})$, and $w_6 \sim \mathcal{N}(0, \sigma_d^2)$.

Lastly, the estimated mass, $\hat{m}_\mathcal{B}$, and inertia matrix, $\hat{\mathbf{J}}_b^{\mathcal{B}z}$, are perturbed from their true values such that

$$\begin{aligned} \hat{m}_\mathcal{B} &= m_\mathcal{B} + w_7 \ \mathrm{kg}, \\ \hat{\mathbf{J}}_{\hat{b}}^{\mathcal{B}z} &= \mathbf{C}_{b\hat{b}}^\mathsf{T} \mathbf{J}_b^{\mathcal{B}z} \mathbf{C}_{b\hat{b}} \ \mathrm{kg} \cdot \mathrm{m}^2, \end{aligned}$$

where $w_7 \sim \mathcal{N}(0, \sigma_m^2)$, and $\mathbf{C}_{b\hat{b}} = \exp_{SO(3)}(\boldsymbol{\phi})$ is a perturbation DCM, where $\boldsymbol{\phi} \sim \mathcal{N}(\mathbf{0}, \sigma_J^2 \mathbf{1})$. By perturbing $\mathbf{J}_b^{\mathcal{B}z}$ with a DCM, cross-coupling is introduced in the inertia matrix, however the eigenvalues of the inertia matrix are preserved. The standard deviation values used to randomize the states and model parameters are shown in Table 5.4.

Table 5.4: State and Parameter Standard Deviations Used in Monte Carlo Simulations

Parameter	Value	Units
σ_ϕ	0.0873	rad
σ_v	0.333	$\mathrm{m} \cdot \mathrm{s}^{-1}$
σ_r	1	m
σ_ω	0.0291	$\mathrm{rad} \cdot \mathrm{s}^{-1}$
σ_m	10	kg
σ_J	0.0436	$\mathrm{kg} \cdot \mathrm{m}^2$
σ_w	1.333	$\mathrm{m} \cdot \mathrm{s}^{-1}$
σ_d	0.667	$\mathrm{m} \cdot \mathrm{s}^{-1}$

The distribution of root-mean-square error (RMSE) results for 100 Monte-Carlo runs is shown in Figure 5.7 in the form of violin plots. In each case, the helicopter is able to reach the target position. The large distribution on the control input, particularly thrust, is due to the applied mass uncertainty. With reduced mass uncertainty, the variance in δf and $\delta \mathbf{m}$ is smaller, as seen in Figure 5.8.

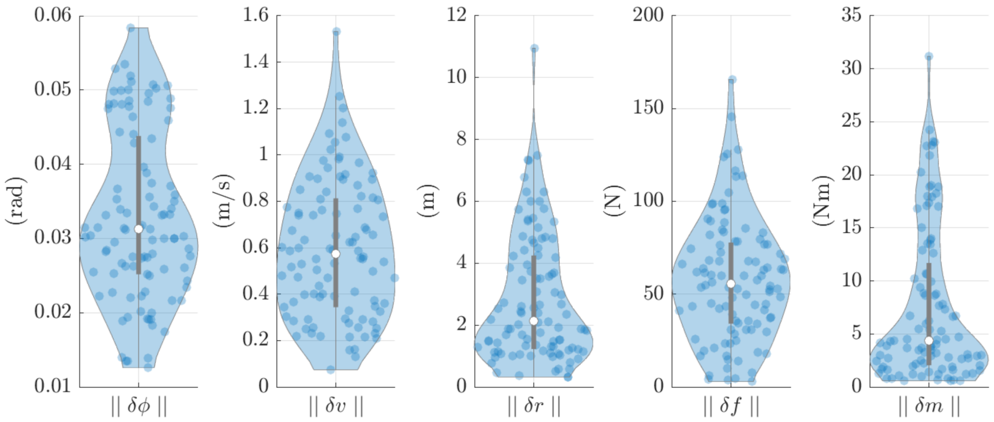


Figure 5.7: State and input RMSE values across 100 Monte-Carlo simulations with random initial conditions, wind gusts, and model uncertainty.

Table 5.5: Prediction Horizon Parameters

Case	Δt_1 (s)	N_{p1}	Δt_2 (s)	N_{p2}	Δt_3 (s)	N_{p3}	T_{pred} (s)
1	0.04	24	0.16	12	0.64	12	10.56
2	0.02	48	-	-	-	-	0.96

5.8.3 Effect of Non-uniform Prediction Horizon

An additional set of Monte-Carlo simulations are run to demonstrate the advantage of the non-uniform prediction horizon. In Case 1, the proposed non-uniform timestep is used, while Case 2 features a uniform fixed-timestep equal to the controller timestep. The prediction horizon parameters for both cases are shown in Table 5.5. In both cases, the prediction horizon contains a total of 48 steps, therefore the problem size is identical. All Monte-Carlo parameters are as previously stated in Section 5.8.1 except for a reduction in the amount of model uncertainty such that $\sigma_m = 3.333$ kg, and $\sigma_J = 0.0436$ rad. Attempts to further increase the model uncertainty result in non-convergence of the MPC in Case 2.

The distribution of RMSE results for 100 Monte-Carlo runs is shown in Figure 5.8. With the longer total prediction horizon, the average attitude, velocity, and position errors in Case 1 are 9%, 62%, and 64% lower, respectively. However, the average thrust and torque input errors in Case 1 are 81% and 32% higher, respectively. Comparing other metrics, the average time to reach the target and the average number of replanned trajectories is 44% and 48% lower respectively in Case 1. Therefore, although the average control effort is higher in Case 1, the longer prediction horizon achieved by the non-uniformly spaced timestep provides substantial benefit in tracking performance.

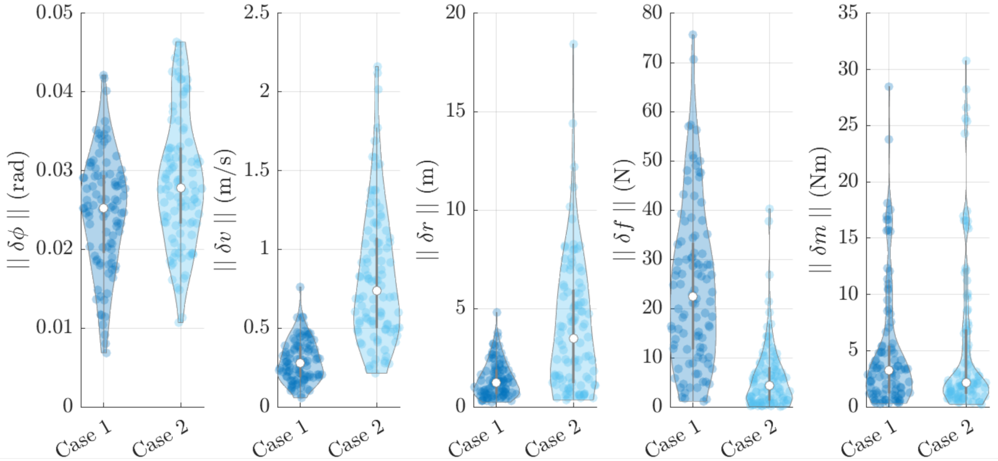


Figure 5.8: State and input RMSE values across 100 Monte-Carlo simulations comparing non-uniform (Case 1) and uniform (Case 2) prediction horizons.

Chapter 6

Cascaded Model Predictive Control

The MPC strategy previously presented in Chapter 5 allows all of the system control inputs to be generated by a single controller, however the resulting control structure is fairly large. The size and complexity of the QP used in the MPC solution grows quickly with the number of states, control inputs, constraints, and horizon length. As an example, consider the MPC structure for the tandem-rotor system with the following features,

- 12 states and 4 control inputs,
- Horizon length of $N_p = N_u = N_c = 20$,
- Control input, attitude keep-in zone, and ℓ_1-norm attitude error constraints.

The resulting QP has 180 optimization variables and 320 constraints. Additionally, the QP must be run at 50 Hz because it includes the "fast" inner-loop dynamics. The combination of a large QP and fast update rate makes real-time implementation on limited hardware infeasible.

A common approach used to avoid solving a QP in real-time is explicit MPC, where the optimal control solutions for all different operating regions are precomputed and stored [49]. Then the controller simply uses a lookup table to determine the current operating region and associated control input. Explicit MPC has a long history of successful implementation as shown in [50] and [51]. It has also been demonstrated on LTV systems as shown in [52] and [53]. Explicit MPC however is only generally practical for small problem sizes [8] because the number of operating regions, and hence size of the lookup table, grows exponentially with the horizon, constraints, and state dimension.

A more practical approach is the cascaded MPC approach where the single MPC is split into separate outer and inner-loop controllers. Cascaded MPC has been commonly used to effectively control multi-timescale systems, such as in [54] where cascaded nonlinear model predictive control (NMPC) is demonstrated on a quadrotor, and in [55] where a cascaded MPC structure is applied to a power system. In this thesis, cascaded MPC provides a method of effectively handling the multi-timescale dynamics of the tandem-rotor helicopter in a less computationally demanding way than the single MPC structure of Chapter 5. Although this approach necessitates solving two QPs in real-time, each MPC has fewer states and less optimization variables, resulting in smaller QPs with less computational complexity.

6.1 Overview

The proposed cascaded MPC structure is shown below in Figure 6.1.

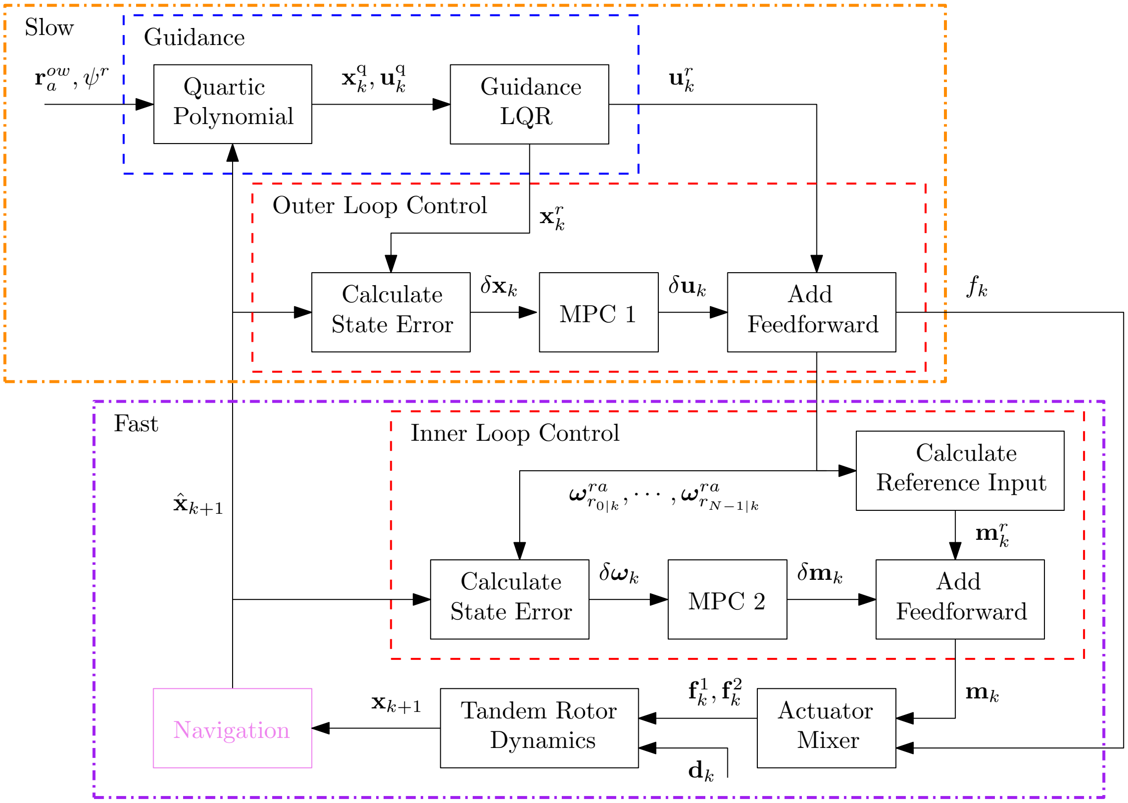


Figure 6.1: Proposed cascaded guidance and control structure.

The inner-loop controller operates on the fast attitude dynamics and continues to run at 50 Hz. However, the outer-loop controller only operates on the slow translational dynamics and can therefore run at a much slower update rate of 10 Hz. In this chapter, the cascaded MPC approach is presented as well as simulation results highlighting the reduction in computation time compared to the single MPC strategy from Chapter 5.

6.2 Outer-loop MPC

The outer-loop MPC operates on the position and velocity dynamics of the tandem-rotor helicopter given by (3.1), (3.2), and (3.3). Instead of the augmented error definition presented in Chapter 5, the elimination of the rotational dynamics allows a purely $SE_2(3)$ error to be used in the outer-loop MPC formulation. Therefore, the tracking error $\delta \mathbf{X}_k$ is as shown in (3.8) and the state is

$$\delta \mathbf{x}_k^o = \begin{bmatrix} \delta \boldsymbol{\xi}_k^{\phi} \\ \delta \boldsymbol{\xi}_k^{v} \\ \delta \boldsymbol{\xi}_k^{r} \end{bmatrix} \in \mathbb{R}^9, \tag{6.1}$$

where the superscript $(\cdot)^o$ refers to a quantity related to the outer-loop MPC. Notice that the state dimension has been reduced from 12 to 9 which decreases the size of the QP. The control inputs are the thrust force and angular velocity commands

$$\mathbf{u}_k^o = \begin{bmatrix} f_k^T \\ \boldsymbol{\omega}_{b_k}^{ba,\text{cmd}} \end{bmatrix} \in \mathbb{R}^4. \tag{6.2}$$

The input error is defined as

$$\delta \mathbf{u}_k^o = \begin{bmatrix} \delta f_k \\ \delta \boldsymbol{\omega}_k \end{bmatrix} = \begin{bmatrix} f_k - f_k^r \\ \boldsymbol{\omega}_{b_k}^{ba,\text{cmd}} - \delta \mathbf{C}_k^{\mathsf{T}} \boldsymbol{\omega}_{r_k}^{ra} \end{bmatrix}. \tag{6.3}$$

6.2.1 Linearization of Dynamics

The linearized dynamics for the outer-loop MPC largely follow Section 3.5, however the linearization of the attitude dynamics is modified slightly. In this case, $\delta \boldsymbol{\omega}$ is a control input instead of a state. Therefore, the attitude error dynamics are linearized as

$$\begin{aligned} \delta \dot{\mathbf{C}} &= \dot{\mathbf{C}}_{ar}^{\mathsf{T}} \mathbf{C}_{ab} + \mathbf{C}_{ar}^{\mathsf{T}} \dot{\mathbf{C}}_{ab}, && (6.4) \\ &= \left(\mathbf{C}_{ar} \boldsymbol{\omega}_r^{ra^\times} \right)^{\mathsf{T}} \mathbf{C}_{ab} + \mathbf{C}_{ar}^{\mathsf{T}} \mathbf{C}_{ab} \boldsymbol{\omega}_b^{ba^\times} && (6.5) \\ &= -\boldsymbol{\omega}_r^{ra^\times} \mathbf{C}_{ar}^{\mathsf{T}} \mathbf{C}_{ab} + \mathbf{C}_{ar}^{\mathsf{T}} \mathbf{C}_{ab} \boldsymbol{\omega}_b^{ba^\times} && (6.6) \\ &= -\boldsymbol{\omega}_r^{ra^\times} \delta \mathbf{C} + \delta \mathbf{C} \boldsymbol{\omega}_b^{ba^\times}. && (6.7) \end{aligned}$$

Using the error definition from (6.3),

$$\delta \dot{\mathbf{C}} = -\boldsymbol{\omega}_r^{ra^\times} \delta \mathbf{C} + \delta \mathbf{C} \left(\delta \boldsymbol{\omega} + \delta \mathbf{C}^{\mathsf{T}} \boldsymbol{\omega}_r^{ra} \right)^{\times}. \tag{6.8}$$

Linearizing (6.8) using (3.18) and dropping higher-order terms,

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{1}+\delta\boldsymbol{\xi}^{\phi^\times}\right) = -\boldsymbol{\omega}_r^{ra^\times}\left(\mathbf{1}+\delta\boldsymbol{\xi}^{\phi^\times}\right)+\left(\mathbf{1}+\delta\boldsymbol{\xi}^{\phi^\times}\right)\left(\delta\boldsymbol{\omega}+\left(\mathbf{1}-\delta\boldsymbol{\xi}^{\phi^\times}\right)\boldsymbol{\omega}_r^{ra}\right)^\times \tag{6.9}$$

$$\delta\dot{\boldsymbol{\xi}}^{\phi^\times} = -\boldsymbol{\omega}_r^{ra^\times}-\boldsymbol{\omega}_r^{ra^\times}\delta\boldsymbol{\xi}^{\phi^\times}+\delta\boldsymbol{\omega}^\times+\boldsymbol{\omega}_r^{ra^\times}-\left(\delta\boldsymbol{\xi}^{\phi^\times}\boldsymbol{\omega}_r^{ra}\right)^\times+\delta\boldsymbol{\xi}^{\phi^\times}\boldsymbol{\omega}_r^{ra^\times} \tag{6.10}$$

$$= \delta\boldsymbol{\omega}^\times-\boldsymbol{\omega}_r^{ra^\times}\delta\boldsymbol{\xi}^{\phi^\times}-\left(\delta\boldsymbol{\xi}^{\phi^\times}\boldsymbol{\omega}_r^{ra}\right)^\times+\delta\boldsymbol{\xi}^{\phi^\times}\boldsymbol{\omega}_r^{ra^\times}. \tag{6.11}$$

Using (2.24),

$$\delta\dot{\boldsymbol{\xi}}^{\phi^\times} = \delta\boldsymbol{\omega}^\times-\boldsymbol{\omega}_r^{ra^\times}\delta\boldsymbol{\xi}^{\phi^\times}-\left(\delta\boldsymbol{\xi}^{\phi^\times}\boldsymbol{\omega}_r^{ra^\times}-\boldsymbol{\omega}_r^{ra^\times}\delta\boldsymbol{\xi}^{\phi^\times}\right)+\delta\boldsymbol{\xi}^{\phi^\times}\boldsymbol{\omega}_r^{ra^\times}, \tag{6.12}$$

$$= \delta\boldsymbol{\omega}^\times. \tag{6.13}$$

Uncrossing both sides,

$$\delta\dot{\boldsymbol{\xi}}^{\phi} = \delta\boldsymbol{\omega} \tag{6.14}$$

The linearized velocity and position error dynamics are identical to those shown in Sections 3.5.2 and 3.5.3. Therefore, the linearized, continuous-time outer-loop dynamics are

$$\underbrace{\begin{bmatrix}\delta\dot{\boldsymbol{\xi}}^{\phi}\\ \delta\dot{\boldsymbol{\xi}}^{v}\\ \delta\dot{\boldsymbol{\xi}}^{r}\end{bmatrix}}_{\delta\dot{\mathbf{x}}^o} = \underbrace{\begin{bmatrix}\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{A}_{21}&\mathbf{A}_{22}&\mathbf{0}\\ \mathbf{0}&\mathbf{1}&-\boldsymbol{\omega}_r^{ra^\times}\end{bmatrix}}_{\mathbf{A}^o}\underbrace{\begin{bmatrix}\delta\boldsymbol{\xi}^{\phi}\\ \delta\boldsymbol{\xi}^{v}\\ \delta\boldsymbol{\xi}^{r}\end{bmatrix}}_{\delta\mathbf{x}^o}+\underbrace{\begin{bmatrix}\mathbf{0}&\mathbf{1}\\ -1/m_{\mathcal{B}}\mathbf{1}_3&\mathbf{0}\\ \mathbf{0}&\mathbf{0}\end{bmatrix}}_{\mathbf{B}^o}\underbrace{\begin{bmatrix}\delta f\\ \delta\boldsymbol{\omega}\end{bmatrix}}_{\delta\mathbf{u}^o}, \tag{6.15}$$

where $\mathbf{A}_{21}$ and $\mathbf{A}_{22}$ are given by (3.61) and (3.62).

6.2.2 Constraints

The attitude keep-in zone, ℓ_1-norm attitude error, and control input constraints are included in the outer-loop MPC and are formulated as shown in Section 5.4. The torque input constraint can be replaced with an identical constraint on the angular velocity input using the method shown in Section 5.4.3.

6.3 Inner-loop MPC

The inner-loop MPC operates on the rotational dynamics of the tandem-rotor helicopter given by (3.4). The state is simply the angular momentum

$$\delta \mathbf{x}^{i}_{k} = \delta \mathbf{h}_{k} \in \mathbb{R}^3, \tag{6.16}$$

where the superscript $(\cdot)^i$ indicates a quantity related to the inner-loop MPC, and the tracking error $\delta \mathbf{h}_k$ is as defined in (3.13). The control inputs are the rotor torque commands

$$\mathbf{u}^{i}_{k} = \mathbf{m}_{b_k} \in \mathbb{R}^3, \tag{6.17}$$

and the input error $\delta \mathbf{m}$ is as defined in (3.17). Note that the state and input dimensions have been reduced to $\mathbb{R}^3$, which significantly reduces the size of the inner-loop QP.

6.3.1 Linearization of Dynamics

The linearized dynamics of the inner-loop rotational dynamics is as shown in Section 3.5.4, therefore the the continuous-time, linearized inner-loop dynamics are

$$\begin{aligned} \delta \dot{\mathbf{h}} = &\left(\left(\mathbf{E} \mathbf{C}^{\mathsf{T}}_{ar} \mathbf{v}^{z_r w/a}_{a} \right)^{\times} - \mathbf{E} \left(\mathbf{C}^{\mathsf{T}}_{ar} \mathbf{v}^{z_r w/a}_{a} \right)^{\times} + \left(\mathbf{F} \boldsymbol{\omega}^{ra}_{r} \right)^{\times} - \mathbf{F} \boldsymbol{\omega}^{ra^{\times}}_{r} \right) \delta \boldsymbol{\xi}^{\phi} \\ &- \mathbf{E} \delta \boldsymbol{\xi}^{v} + \left(- \boldsymbol{\omega}^{ra^{\times}}_{r} - \mathbf{F} \mathbf{J}^{\mathcal{B} z_r^{-1}}_{r} \right) \delta \mathbf{h} + \delta \mathbf{m}. \end{aligned} \tag{6.18}$$

For simplicity, the drag terms in the inner-loop are ignored, resulting in

$$\underbrace{\delta \dot{\mathbf{h}}}_{\delta \dot{\mathbf{x}}^i} = \underbrace{- \boldsymbol{\omega}^{ra^{\times}}_{r}}_{\mathbf{A}^i} \underbrace{\delta \mathbf{h}}_{\delta \mathbf{x}^i} + \underbrace{\mathbf{1}}_{\mathbf{B}^i} \underbrace{\delta \mathbf{m}}_{\delta \mathbf{u}^i} . \tag{6.19}$$

From (6.19), it can be seen that a reference angular velocity trajectory $\boldsymbol{\omega}^{ra}_{r}$ is required in $\mathbf{A}^i$ to linearize the inner-loop dynamics. Recall that the outer-loop MPC generates a sequence of optimal control inputs, $\mathbf{u}^{\star}_{k}$. The sequence of optimal angular velocity commands, $\boldsymbol{\omega}^{\star}_{k}$ are extracted from $\mathbf{u}^{\star}_{k}$ yielding

$$\boldsymbol{\omega}^{\star}_{k} = \begin{bmatrix} \boldsymbol{\omega}^{ra}_{r_{0|k}} \\ \boldsymbol{\omega}^{ra}_{r_{1|k}} \\ \vdots \\ \boldsymbol{\omega}^{ra}_{r_{N-1|k}} \end{bmatrix} . \tag{6.20}$$

Instead of using the reference angular velocity from the guidance, the inner-loop dynamics are linearized about $\boldsymbol{\omega}_k^\star$. The resulting linearization more accurately represents the desired trajectory.

Note that from the definition of $\delta\mathbf{m}$, a reference torque input trajectory, $\mathbf{m}_{r_k}^r$ must also be supplied to the inner-loop MPC. This is accomplished using the differential flatness property of the system and the steps described in Section 4.2.2. First the reference angular acceleration, $\dot{\boldsymbol{\omega}}_{r_k}^{ra}$ is calculated using (4.41) to (4.43). Then, the reference torque input, $\mathbf{m}_{r_k}^r$ can be calculated using (4.49).

6.3.2 LTI Assumption

It can be seen from (6.19) that the linearized inner-loop dynamics are an LTV system because the reference angular velocity in $\mathbf{A}^i$ is time-varying. Consider the cascaded control structure shown in Figure 6.1. Assuming that the outer-loop MPC is tuned such that the control inputs generated by the controller vary slowly, the reference angular velocity trajectory used to linearize the inner-loop dynamics can be approximated as a constant value for each iteration of the outer-loop controller,

$$\boldsymbol{\omega}_k^\star \approx \begin{bmatrix} \boldsymbol{\omega}_{r_{0|k}}^{ra} \\ \vdots \\ \boldsymbol{\omega}_{r_{0|k}}^{ra} \end{bmatrix}. \tag{6.21}$$

Using this assumption, (6.19) becomes an LTI system, and the inner-loop dynamics only need to be linearized and discretized once per iteration of the outer-loop controller. The $\boldsymbol{\mathcal{S}}_k$ and $\boldsymbol{\mathcal{M}}_k$ matrices from (5.5) are simplified to

$$\boldsymbol{\mathcal{S}}_k^i = \begin{bmatrix} \mathbf{B}^i & & & \\ \mathbf{A}^i\mathbf{B}^i & \mathbf{B}^i & & \\ \vdots & \vdots & \ddots & \\ \mathbf{A}^{i^{N-1}}\mathbf{B}^i & \mathbf{A}^{i^{N-2}}\mathbf{B}^i & \cdots & \mathbf{B}^i \end{bmatrix}, \quad \boldsymbol{\mathcal{M}}_k^i = \begin{bmatrix} \mathbf{A}^i \\ \mathbf{A}^{i^2} \\ \vdots \\ \mathbf{A}^{i^N} \end{bmatrix}. \tag{6.22}$$

6.3.3 Constraints

A state constraint on the angular momentum is not necessary since the angular velocity command is already constrained in the outer-loop MPC. Therefore, the only constraint included in the inner-loop MPC is a control input constraint on the torque. This constraint is formulated as shown in Section 5.4.3.

6.4 Simulation Results

Monte-Carlo simulations are performed to demonstrate the performance of the cascaded MPC structure. Case 1 features the single MPC structure presented in Chapter 5. The parameters used in the guidance and control algorithms are identical to those shown in Table 5.2, and Table 5.3. Case 2 features the cascaded MPC structure presented in this chapter. The controller parameters used in the outer-loop and inner-loop MPC algorithms are shown in Table 6.1 and Table 6.2, respectively. The vehicle parameters and standard deviation values used in the Monte-Carlo simulations for both Case 1 and Case 2 are as shown in Table 5.1 and Table 5.4, respectively.

The guidance and outer-loop controller run at 10 Hz, while the inner-loop controller runs at 50 Hz. Note that while a non-uniform horizon is used in the outer-loop MPC to extend the total prediction horizon time, a traditional fixed timestep prediction horizon is used in the inner-loop MPC. This is done because the inner-loop update rate is relatively fast and the inner-loop dynamics operate on a single timescale. Therefore, the benefits to using a non-uniform horizon for the inner-loop MPC are greatly reduced.

The distribution of state and input RMSE values across 100 Monte-Carlo simulations are shown in Figure 6.2. Although the average attitude tracking error is larger in Case 2, the velocity and position tracking performance shows improvement in Case 2. Similarly, the thrust input error is larger in Case 2, however the torque input error is reduced in Case 2. This indicates that the inner-loop MPC is able to track the reference torque command better than the single MPC controller.

While a clear tracking performance benefit is not seen, the cascaded MPC structure shows significant reduction in computational complexity from having two smaller QPs compared to a single larger QP. To quantify this effect, the total CPU time used to solve the constrained QP at each timestep is summed for each simulation. The distribution of total CPU solve time across the 100 Monte-Carlo simulations is shown in Figure 6.3. The average CPU solve time for each simulation is 56.0 s and 5.4 s for Case 1 and Case 2, respectively. Note that the average simulation time for the landing trajectory is 14.6 s. The large QP from the single MPC structure in Case 1 results in long, inconsistent solve times meaning it cannot be implemented in real-time. However, the smaller QPs from the cascaded MPC structure in Case 2 results in shorter, more consistent solve times that are better suited for real-time implementation.

Table 6.1: Outer-loop MPC Parameters Used in Simulation

Parameter	Value	Units
dt	0.1	s
N_p	20	-
N_u	5	-
N_c	5	-
$\mathbf{Q}^\phi$	$\text{diag}(1\text{e}1, 1\text{e}1, 1\text{e}1)$	-
$\mathbf{Q}^v$	$\text{diag}(1\text{e}1, 1\text{e}1, 1\text{e}1)$	-
$\mathbf{Q}^r$	$\text{diag}(1\text{e}1, 1\text{e}1, 1\text{e}1)$	-
$\mathbf{Q}^h$	$\text{diag}(1\text{e}1, 1\text{e}1, 1\text{e}1)$	-
$\mathbf{Q}$	$\text{diag}(\mathbf{Q}^\phi, \mathbf{Q}^v, \mathbf{Q}^r, \mathbf{Q}^h)$	-
$\mathbf{P}$	$10 \cdot \mathbf{Q}$	-
$\mathbf{R}$	$\text{diag}(1\text{e} - 3, 1, 1, 1)$	-
η_1	$1e24$	-
η_2	$1e24$	-
Δt_1	0.2	s
Δt_2	0.4	s
Δt_3	1.6	s
N_{p_1}	10	-
N_{p_2}	5	-
N_{p_3}	5	-
$\mathbf{u}_{\text{min}}$	$\begin{bmatrix} 0 & -2 & -2 & -2 \end{bmatrix}^\intercal$	$\text{N, rad} \cdot \text{s}^{-1}$
$\mathbf{u}_{\text{max}}$	$\begin{bmatrix} 3000 & 2 & 2 & 2 \end{bmatrix}^\intercal$	$\text{N, rad} \cdot \text{s}^{-1}$
α	10	deg
γ	0.1	rad

Table 6.2: Inner-loop MPC Parameters Used in Simulation

Parameter	Value	Units
dt	0.02	s
N_p	10	-
N_u	5	-
N_c	5	-
$\mathbf{Q}$	$\text{diag}(1\text{e}3, 1\text{e}3, 1\text{e}3)$	-
$\mathbf{P}$	$1 \cdot \mathbf{Q}$	-
$\mathbf{R}$	$\text{diag}(1, 1, 1)$	-
$\mathbf{u}_{\text{min}}$	$\begin{bmatrix} -200 & -200 & -200 \end{bmatrix}^\intercal$	$\text{N} \cdot \text{m}$
$\mathbf{u}_{\text{max}}$	$\begin{bmatrix} 200 & 200 & 200 \end{bmatrix}^\intercal$	$\text{N} \cdot \text{m}$

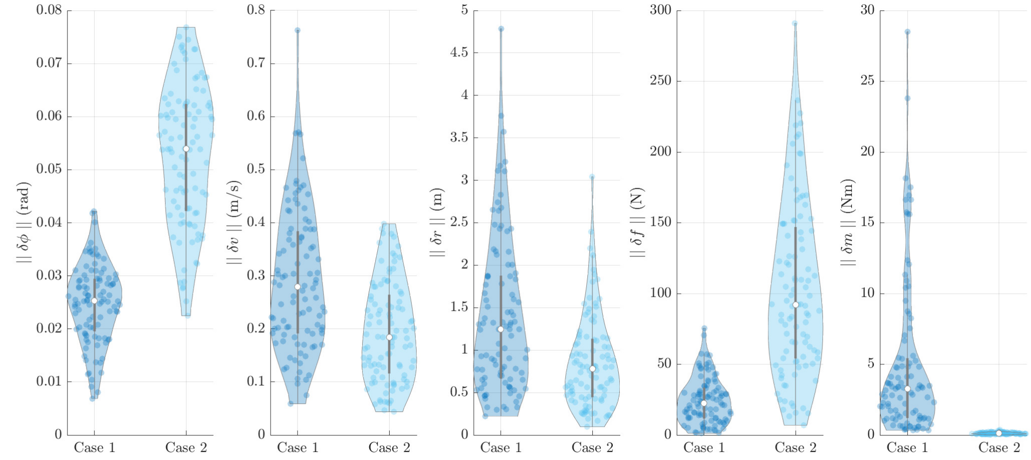


Figure 6.2: State and input RMSE values across 100 Monte-Carlo simulations comparing the single MPC structure (Case 1) and the cascaded MPC structure (Case 2).

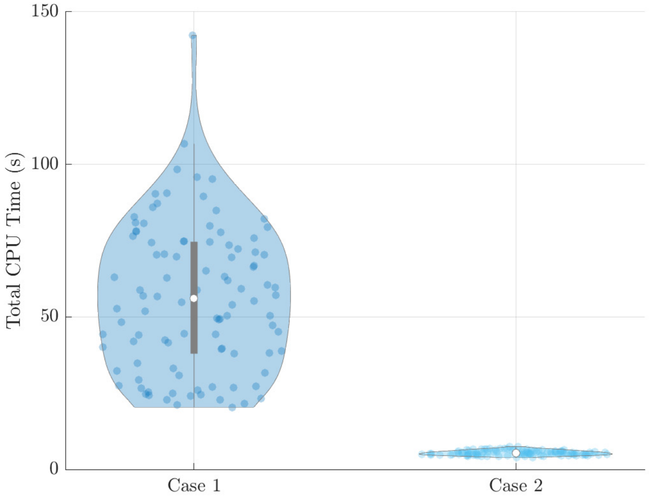


Figure 6.3: Total CPU solve time across 100 Monte-Carlo simulations comparing the single MPC structure (Case 1) and the cascaded MPC structure (Case 2).

Chapter 7

Closing Remarks and Future Work

7.1 Conclusions

This thesis presents a guidance and control strategy for a tandem-rotor helicopter. First, the nonlinear system dynamics are developed and the linearization is presented using an augmented $SE_2(3)$ error definition. Next, a two-stage guidance approach is developed, which leverages the differentially flat property of the dynamics and a discrete-time, finite-horizon LQR algorithm to generate a reference trajectory. This approach is computationally lightweight as it is purely analytical and does not require the solution of an optimization problem. Simulations are performed to demonstrate the ability of the guidance strategy to match the boundary conditions of all states.

For control, an MPC approach is presented. A non-uniformly spaced prediction horizon is used to predict the multi-timescale dynamics while limiting the optimization problem size. A novel method of attitude constraint is presented using a combination of attitude keep-in zone and ℓ_1-norm attitude error constraints. Monte-Carlo simulations demonstrate robustness of the control strategy to random initial conditions, model uncertainty, and environmental disturbances. Additionally, the non-uniform prediction horizon is shown to be beneficial over a traditional fixed prediction horizon. Although a linear MPC formulation is used, the problem size still exceeds the limits of simple real-time computing platforms. Therefore, a cascaded MPC approach is presented as an alternative controller implementation. The output of the outer-loop MPC is used to linearize the inner-loop system dynamics. Further Monte-Carlo simulations show the significant reduction in computational effort required by the cascaded MPC structure and suggest that this method is more suitable for real-time implementation.

7.2 Future Work

One notable avenue for further development is the concept of $\mathcal{H}_\infty$ loop-shaping on MPC, as shown in [56]. This MPC formulation allows the controller to reject constant unmeasured disturbances with zero steady-state error and meet transient performance objectives. At the same time, the closed-loop stability properties of the $\mathcal{H}_\infty$ formulation are guaranteed as long as the constraints are inactive.

Recall that the cascaded MPC structure presented in Chapter 6 was not able to match the attitude tracking performance of the single MPC structure from Chapter 5. Given the clear advantages of the cascaded MPC structure in terms of its suitability for real-time implementation, more effort should be given to improving the tracking performance of the cascaded MPC implementation.

Lastly, given the results of Chapter 4 and 6, an attempt should be made to implement the proposed guidance and cascaded MPC algorithms on an embedded computing platform. The guidance and control structure could be modified with minimal effort to work with an off-the-shelf UAV platform by simply swapping out the tandem-rotor helicopter dynamic model and modifying the control input mapping. Real-world experiments would greatly help validate the potential performance of the guidance and control algorithms.

Bibliography

[1] H. Shakhatreh *et al.*, “Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges”, *IEEE Access*, vol. 7, pp. 48 572–48 634, 2019. arXiv: `1805.00881`.

[2] N. Cherif, W. Jaafar, H. Yanikomeroglu, and A. Yongacoglu, “3D Aerial Highway: The Key Enabler of the Retail Industry Transformation”, *IEEE Commun. Mag.*, pp. 65–71, 2021. arXiv: `2009.09477`.

[3] W. Johnson, *Rotorcraft Aeromechanics*. New York: Cambridge University Press, 2013.

[4] D. Bernstein and W. Haddad, “Nonlinear Controllers for Positive Real Systems With Arbitrary Input Nonlinearities”, *IEEE Trans. Automat. Contr.*, vol. 39, no. 7, pp. 1513–1517, 1994.

[5] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd. Nob Hill Publishing, 2017.

[6] M. Morari and J. H. Lee, “Model Predictive Control: Past, Present and Future”, *Comput. Chem. Eng.*, vol. 23, no. 4-5, pp. 667–682, 1999.

[7] M. Schwenzer, M. Ay, T. Bergs, and D. Abel, “Review on Model Predictive Control: An Engineering Perspective”, *Int. J. Adv. Manuf. Technol.*, vol. 117, no. 5-6, pp. 1327–1349, 2021.

[8] Y. Wang and S. Boyd, “Fast Model Predictive Control Using Online Optimization”, *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 267–278, 2010.

[9] P. C. Hughes, *Spacecraft Attitude Dynamics*. Dover, 2004.

[10] A. Barrau, “Non-linear State Error Based Extended Kalman Filters With Applications to Navigation”, Ph.D. dissertation, Mines Paristech, 2015.

[11] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2017.

[12] M. J. Van Nieuwstadt and R. M. Murray, “Real Time Trajectory Generation for Differentially Flat Systems”, *Act. Disturb. Rejection Control Dyn. Syst.*, vol. 017, 1997.

[13] M. Fliess, P. Martin, and P. Rouchon, "Sur les Systèmes Non Linéaires Différentiellement Plats", *Comptes rendus l'Académie des Sci. Série I, Mathématique*, vol. 315, no. January, pp. 619–624, 1992.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization.* New York: Cambridge University Press, 2004.

[15] S. Mehrotra, "On the Implementation of a Primal-Dual Interior Point Method", *SIAM J. Optim.*, vol. 2, no. 4, pp. 575–601, 1992.

[16] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization.* Academic Press, 1982.

[17] W. Z. Stepniewski, *Rotary-wing Aerodynamics.* Dover Publications, 1984.

[18] Boeing, *Boeing Marks 50 Years of Delivering Chinook Helicopters*, 2012. [Online]. Available: `https://boeing.mediaroom.com/news-releases-statements?item=2389` (visited on 06/14/2022).

[19] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential Flatness of Quadrotor Dynamics Subject to Rotor Drag for Accurate Tracking of High-speed Trajectories", *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 620–626, 2018. arXiv: `1712.02402`.

[20] M. R. Cohen, K. Abdulrahim, and J. R. Forbes, "Finite-Horizon LQR Control of Quadrotors on SE2(3)", *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5748–5755, 2020. arXiv: `2105.13935`.

[21] M. R. Hartley, "Contact-aided State Estimation on Lie Groups for Legged Robot Mapping and Control", Ph.D. dissertation, University of Michigan, 2019.

[22] T. Lee, M. Leok, and N. H. McClamroch, "Geometric Tracking Control of a Quadrotor UAV on SE(3)", in *IEEE Conf. Decis. Control*, Atlanta, 2010, pp. 5420–5425.

[23] J. A. Farrell, *Aided Navigation Systems: GPS and High Rate Sensors.* New York: McGraw-Hill, 2008.

[24] C. F. Van Loan, "Computing Integrals Involving the Matrix Exponential", *IEEE Trans. Automat. Contr.*, vol. AC-23, no. 3, pp. 395–404, 1978.

[25] D. J. Moorhouse and R. J. Woodcock, "Background Information and User Guide for MIL-F-8785C, Military Specification - Flying Qualities of Piloted Airplanes", Air Force Wright Aeronautical Labs, Wright-Patterson AFB, OH, Tech. Rep., 1982.

[26] MathWorks, *Dryden Wind Turbulence Model (Discrete)*, 2022. [Online]. Available: `https://www.mathworks.com/help/aeroblks/drydenwindturbulencemodeldiscrete.html` (visited on 06/12/2022).

[27] R. F. Stengel, *Flight Dynamics.* Princeton University Press, 2004.

[28] M. W. Mueller, M. Hehn, and R. D. Andrea, “A Computationally Efficient Motion Primitive for Quadrocopter Trajectory Generation”, *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1294–1310, 2015.

[29] M. Kelly, “An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation”, *SIAM Rev.*, vol. 59, no. 4, pp. 849–904, 2017.

[30] G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, “Quadrotor Helicopter Trajectory Tracking Control”, *AIAA Guid. Navig. Control Conf. Exhib.*, pp. 1–14, 2008.

[31] M. Hehn and R. D’Andrea, *Quadrocopter Trajectory Generation and Control.* IFAC, 2011, vol. 44, pp. 1485–1491.

[32] D. Mellinger and V. Kumar, “Minimum Snap Trajectory Generation and Control for Quadrotors”, in *IEEE Int. Conf. Robot. Autom.*, 2011, pp. 2520–2525.

[33] J. D. Lafontaine, D. Neveu, and K. Lebel, “Autonomous Planetary Landing With Obstacle Avoidance: The Quartic Guidance Revisited”, in *14th AAS/AIAA Sp. Flight Mech. Conf.*, Maui, 2004.

[34] R. F. Stengel, *Optimal Control and Estimation.* Mineola: Dover Publications, 1994.

[35] D. R. Downing and W. H. Bryant, “Flight Test of a Digital Controller Used in a Helicopter Autoland System”, *Automatica*, vol. 23, no. 3, pp. 295–300, 1987.

[36] A. Dzul, T. Hamel, and R. Lozano, “Nonlinear Control for a Tandem Rotor Helicopter”, *IFAC Proc. Vol.*, vol. 35, no. 1, pp. 229–234, 2002.

[37] L. Gaoyuan, W. Mei, and A. Ashraf, “L1 Adaptive Control for Tandem-Rotor Helicopter with Anti-Disturbance Capability”, in *2018 Int. Conf. Control. Autom. Inf. Sci. (ICCAIS 2018)*, IEEE, 2018, pp. 372–377.

[38] S. Bonnabel, P. Martin, and E. Salaün, “Invariant Extended Kalman filter: Theory and Application to a Velocity-aided Attitude Estimation Problem”, in *IEEE Conf. Decis. Control*, Shanghai: IEEE, 2009, pp. 1298–1304.

[39] A. Barrau and S. Bonnabel, “The Invariant Extended Kalman Filter as a Stable Observer”, *IEEE Trans. Automat. Contr.*, vol. 62, no. 4, pp. 1797–1812, 2017. arXiv: `1410.1465`.

[40] S. Diemer and S. Bonnabel, “An Invariant Linear Quadratic Gaussian Controller for a Simplified Car”, in *IEEE Int. Conf. Robot. Autom.*, Seattle: IEEE, 2015, pp. 448–453.

[41] D. E. Chang, K. S. Phogat, and J. Choi, “Model Predictive Tracking Control for Invariant Systems on Matrix Lie Groups via Stable Embedding into Euclidean Spaces”, *IEEE Trans. Automat. Contr.*, vol. 65, no. 7, pp. 3191–3198, 2020.

[42] U. V. Kalabić, R. Gupta, S. Di Cairano, A. M. Bloch, and I. V. Kolmanovsky, “MPC on Manifolds With an Application to the Control of Spacecraft Attitude on SO(3)”, *Automatica*, vol. 76, pp. 293–300, 2017. arXiv: `1509.08567`.

[43] D. Zlotnik, S. Di Cairano, and A. Weiss, “MPC for Coupled Station Keeping, Attitude Control, and Momentum Management of GEO Satellites Using On-off Electric Propulsion”, in *Am. Control Conf.*, Boston: American Automatic Control Council (AACC), 2016, pp. 1835–1840.

[44] C. K. Tan, M. J. Tippett, and J. Bao, “Model Predictive Control With Non-uniformly Spaced Optimization Horizon for Multi-timescale Processes”, *Comput. Chem. Eng.*, vol. 84, pp. 162–170, 2016.

[45] T. Brudigam, D. Prader, D. Wollherr, and M. Leibold, “Model Predictive Control with Models of Different Granularity and a Non-uniformly Spaced Prediction Horizon”, in *Am. Control Conf.*, New Orleans, 2021, pp. 3876–3881. arXiv: `2108.08014`.

[46] P. Scokaert and J. Rawlings, “Constrained Linear Quadratic Regulation”, *IEEE Trans. Automat. Contr.*, vol. 43, no. 8, pp. 1163–1169, 1998.

[47] A. Walsh, J. R. Forbes, S. A. Chee, and J. J. Ryan, “Kalman-filter-based Unconstrained and Constrained Extremum-seeking Guidance on SO(3)”, *J. Guid. Control. Dyn.*, vol. 40, no. 9, pp. 2260–2271, 2017.

[48] M. T. Heath, *Scientific Computing: An Introductory Survey*, 2nd. McGraw-Hill, 2002.

[49] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, “The Explicit Linear Quadratic Regulator for Constrained Systems”, *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[50] S. Di Cairano, A. Bemporad, I. V. Kolmanovsky, and D. Hrovat, “Model Predictive Control of Magnetically Actuated Mass Spring Dampers for Automotive Applications”, *Int. J. Control*, vol. 80, no. 11, pp. 1701–1716, 2007.

[51] F. Borrelli, A. Bemporad, M. Fodor, and D. Hrovat, “A Hybrid Approach to Traction Control”, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2034, pp. 162–174, 2001.

[52] J. Lee and H. J. Chang, “Explicit Model Predictive Control for Linear Time-variant Systems With Application to Double-lane-change Maneuver”, *PLoS One*, vol. 13, no. 12, pp. 1–17, 2018.

[53] T. Besselmann and M. Morari, “Autonomous Vehicle Steering Using Explicit LPV-MPC”, *2009 Eur. Control Conf. ECC 2009*, no. 1, pp. 2628–2633, 2014.

[54] J. Schlagenhauf, P. Hofmeier, T. Bronnenmeyer, R. Paelinck, and M. Diehl, “Cascaded Nonlinear MPC for Realtime Quadrotor Position Tracking”, *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 7026–7032, 2020.

[55] A. Ulbig, M. Arnold, S. Chatzivasileiadis, and G. Andersson, *Framework for Multiple Time-scale Cascaded MPC Application in Power Systems*. IFAC, 2011, vol. 44, pp. 10 472–10 480.

[56] S. A. Bortoff, P. Schwerdtner, C. Danielson, S. Di Cairano, and D. J. Burns, “H-infinity Loop-shaped Model Predictive Control With HVAC Application”, *IEEE Trans. Control Syst. Technol.*, pp. 1–16, 2022.