Nonlinear Incremental Optimal Control for Underactuated Mechanical Systems

Thesis report

by

Tingyu Zhang

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on October 30, 2023 at 09:30

Thesis committee:

Chair:	Dr.ir. R. De Breuker
Supervisors:	Dr. X. Wang
External examiner:	Dr.ir. E. Van Kampen
Place:	Faculty of Aerospace Engineering, Delft
Project Duration:	January, 2023 - October, 2023
Student number:	5478413

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Faculty of Aerospace Engineering · Delft University of Technology


TU Delft
Delft
University of
Technology

Preface

I am always interested in the optimal control theorem as I was told that scientists and engineers have actually implemented the optimal control into the rocket and sent it to the moon in the 20th century. The evolving mathematics behind optimal control is both fascinating and frightening. It is just math, but it is all about math.

Writing this thesis, I have fully delved into the theory behind the nonlinear model predictive control. The complexities of stability, feasibility, and optimality at times felt overwhelming. I still have not fully mastered the intricacies of robust stability, but I feel confident in my understanding of nominal stability scenarios. The implementation of the control algorithm made me realize that the soundness of a well-established control theorem does not necessarily guarantee ease of implementation.

I would like to thank my supervisor, Sherry Wang, for her introduction to these captivating subjects and for her steadfast support throughout my research. Our weekly discussions were a continuous source of inspiration, motivating me to delve deeper into my work. My appreciation also extends to Sihao Sun for his invaluable insights into control and simulations.

Tingyu Zhang

Oct. 2023

Contents

Acronyms

DOB Disturbance Observer. 38
DP Dynamic Programming. 29
INBS Incremental Nonlinear Backstepping. 2, 27
INDI Incremental Nonlinear Dynamic Inversion. 2, 27, 43, 47, 48
ISMC Incremental Sliding Mode Control. 2, 27, 38
LMI Linear Matrix Inequality. 24
LPV Linear Parameter Varing. 1, 24
LQR Linear Quadratic Regulator. 29
LTV Linear Time-varying. 24
MPC Model Predictive Control. 29
NBS Nonlinear Backstepping. 1, 25
NDI Nonlinear Dynamic Inversion. 1, 25
NMPC Nonlinear Model Predictive Control. 2, 3, 29–31, 33–38, 40, 41, 43, 47, 48, 50
PID Proportional Integral Derivative. 27
SMC Sliding Mode Control. 1, 26
SOTA State-Of-The-Art. 47
UMS Underactuated Mechanical Systems. 1–3, 19, 21, 25–28, 30, 43, 44

Nomenclature

ζ	position vector , m
v	velocity vector , m/s
η	Euler angle vector , rad
Ω	rate of change of the orientation angles in the inertial frame , rad/s
Ω_B	rate of change of the orientation angles in the body frame , rad/s
ω	rotor angualr speed , rad/s
θ	pitch angle , rad
ϕ	roll angle , rad
ψ	yaw angle , rad
q	unit quaternion
$\mathcal{F}_I$	inertial frame
$\mathcal{F}_B$	body frame
m	mass , kg
g	gravitational constant , m/s^2
T	collective thrust , N
I_v	inertia matrix
τ	torque , Nm
$\mathbb{X}$	state constraint set
$V_N(\cdot)$	cost function for N prediction horizon
$l(\cdot)$	stage cost
$V_f(\cdot)$	terminal cost
$\mathbb{U}$	input constraint set
$\mathbb{X}_f$	terminal constraint
$V_N^0(x)$	optimal cost
u^0	optimal control sequence
$\kappa_N(x)$	first control action from the optimal control sequence
X_N	domain of attraction

List of Figures

List of Tables

1

Introduction

1.1. Motivation

Underactuated Mechanical Systems (UMS) are a class of mechanical models with fewer control inputs than their controlled states. The causes for underactuation can be attributed to i) the inherent dynamics of the system, ii) specific design choices, iii) actuator failures, and iv) the use of artificial low-order nonlinear models to gain insights into the control of higher-order UMS [1]. While simpler UMS, such as the inverse pendulum and the beam-and-ball system, have been thoroughly investigated in literature [2], this research focuses on more complex, naturally occurring nonlinear UMS. Their profound impact and applications in fields like robotics and aerospace underscore their significance. Examples of these systems include aeroelastic systems, aircraft, cranes, underwater vehicles and unmanned air vehicles.

Controlling nonlinear UMS necessitates an in-depth understanding of various aspects such as their inherent nonlinearities, built-in underactuation, stability characteristics, as well as input and state saturations. Additionally, to enhance the robustness of control, it is crucial to reduce reliance on the precise system model. Though a plethora of traditional nonlinear control strategies for UMS has been developed, they often fall short of comprehensively addressing the facets emphasised above. For instance, the Nonlinear Backstepping (NBS) has been applied to the vertical take-off aircraft [3], and the Nonlinear Dynamic Inversion (NDI) was introduced as a control mechanism for UMS in [4]. As both NBS and NDI are categorised under feedback linearisation techniques, they require specific dynamic prerequisites for effective implementation. However, the existing feedback linearisation coordinates transformation approaches [5][1] tend to either complicate the controller design – by intertwining linear and nonlinear subsystems with a new control input – or are limited in their applications to UMS where the number of controlled inputs is at least as many as that of unactuated states. When one achieves the desired feedback linearisation form, the efficacy of either NBS or NDI becomes contingent upon the stability of the internal dynamics containing all the unobservable states. Consequently, supplementary controllers become indispensable to stabilise these internal dynamics. Beyond these structural and stability challenges brought by underactuation, both NBS and NDI exhibit a strong model dependency and lack provisions to handle input and state constraints.

In contrast, Sliding Mode Control (SMC) has earned acclaim due to its adeptness at handling minor and unstructured uncertainties, highlighted by its application in contexts like underactuated satellites [6] and helicopters [7]. However, it is noteworthy that in low model dependent SMC, higher gains are often preferred. These elevated gains might lead to the onset of input and state saturations and generate high-frequency oscillations known as 'chattering'.

While the aforementioned methods can adequately address nonlinearities, another approach to modelling nonlinear systems is through the Linear Parameter Varing (LPV) paradigm. LPV models, characterised by linear state-space representations with state and time-dependent matrices, render conventional linear control strategies viable. Notably, while effective against state-dependent nonlinearities, LPV models cannot capture hard nonlinearities such as saturation and nonlinearly coupled modes [8]. The stability criteria for controlling LPV models are still in the early stage, with the majority of stability and performance assessments being validated through empirical experiments and high-fidelity simulations. Moreover, the literature on modelling and control of UMS via the LPV approach remains sparse, with only a handful of studies [9] [10].

Recent innovations in control methodologies have sought to minimize the pronounced model dependencies seen in conventional techniques. The novel incremental approaches, Incremental Nonlinear Dynamic Inversion (INDI), Incremental Nonlinear Backstepping (INBS), and Incremental Sliding Mode Control (ISMC), are significant examples. They utilise additional measurements to supplant part of the model information. The successful applications of these methods demonstrate superior performance and robustness [11, 12, 13, 14, 15]. Notably, the ISMC method, as elaborated upon in [15], showcases diminished control gains. For those keen on stability evaluations, INDI has been discussed in [16], while INBS is covered in [17] and [14]. Despite being more robust and less model-dependent compared to traditional nonlinear control methods, controllers solely based on a single incremental control approach have limitations. Specifically, both INBS and INDI are intrinsically unable to stabilise the internal dynamics introduced by underactuation. Moreover, these methods offer limited capability in addressing states and input constraints.

Optimal control offers a distinct advantage in its versatility, capable of addressing both underactuation and nonlinearities without restriction by system type and structure. In cases where a longer tracking horizon is required, the concept of Nonlinear Model Predictive Control (NMPC) arises. A salient merit of NMPC lies in its intrinsic capacity to manage input and state constraints as encompassed within the NMPC problem formulation. Its stability conditions are detailed in [18][19][20][21]. However, being a model-based approach, NMPC exhibits strong model dependency and is sensitive to noise. Fortuitously, the advent of Robust NMPC with tightened constraints presents a promising avenue to bolster its robustness. Specifically, nominal NMPC with tightened constraints, as explored in [22], assures the robust satisfaction of state and input constraints. The efficacy of this robust NMPC has been validated in experimental settings [23]. However, this approach employs the upper disturbance bounds for constraint tightening, potentially resulting in a conservative or infeasible solution. To mitigate conservativeness, two different approaches [24] and [25] have been proposed. Their primary objectives are to lower uncertainties and relax the conservativeness of robust NMPC solutions. However, the sliding manifold in [24] heavily depends on model accuracy and a disturbance estimation error dynamics is required in [25]. Furthermore, only input-matched disturbances are compensated in these approaches, leaving the orthogonal complement disturbances untouched. Consequently, there is an urgent need for a less model-dependent approach that can effectively address both matched and unmatched uncertainties and disturbances.

Among the design aspects, namely nonlinearity, stability, underactuation, states and input constraints, robustness, and model dependency, the robust NMPC shows promising potential in addressing most of the problem but has limitations regarding heavy reliance on the model. On the other hand, the nonlinear incremental control offers a notable advantage of lower model dependency. Can these strengths be synergistically combined?

A recent study hinted at this possibility, presenting a loosely coupled NMPC with INDI [26]. However, its reliance on a nominal NMPC model lacks robustness considerations. This raises the question: Is it feasible to integrate incremental nonlinear control into the robust NMPC framework, paralleling techniques from [24] and [25], to encapsulate disturbances more effectively?

In light of these considerations, this research proposes a tightly coupled robust NMPC with an incremental nonlinear control approach. By utilising online sensor-based observations, we aim to curtail uncertainties and estimate the disturbance residues. These residues are then managed by the constraint-tightening mechanism to ensure robustness. We will validate the proposed control strategy's proficiency across diverse UMS under varying simulated conditions.

1.2. Aim and Research Questions

While many UMS have been investigated on a case-by-case basis, the input-affine UMS emerges as a pivotal model for stability and control research. This model is particularly relevant given its applicability to prevalent aerospace systems, including the aeroelastic wing and quadrotor. The primary objective of this research is to develop an incremental optimal nonlinear control for input-affine UMS. This control seeks to comprehensively address the system's inherent challenges, such as nonlinearity, underactuation, stability concerns, robustness, and constraints on states and inputs. The research goal is framed as:

> **"Development of nonlinear incremental optimal control for underactuated mechanical systems."**

Building on this research aim and the preceding discussions on control prerequisites, the primary research question and its related sub-questions are articulated as:

> **"How can robust NMPC and incremental nonlinear control be hybridised in a new control scheme that effectively controls input affine UMS, while ensuring Lyapunov stability, minimizing model dependency, adhering to states and input constraints, and reinforcing robustness."**

1. What types of disturbances should be considered and included?
2. How can a stable cost function be constructed for the robust NMPC?
3. What types of incremental nonlinear control are to be synchronised with robust NMPC?
4. How can the selected incremental nonlinear control be utilised to filter out disturbances and improve robustness?
5. How to formulate tightened constraints to ensure robustness?

1.3. Report Outline

The body of this report is organised into four parts. Part I commences with a scientific paper that presents the research methodologies, results and discussion of the main research phase of this thesis. Thereby, the scientific paper presents a deeply coupled robust NMPC with INDI for underactuated mechanical systems. The scientific paper is followed by part II, which includes a literature review and a preliminary analysis. Part II ends with a preliminary analysis of the proposed controller applied to a quadrotor. Then, part III follows with additional results on the proposed robust NMPC controller with different terminal conditions. Finally, part IV wraps up the report by revisiting and addressing the research questions delineated in section 1.2 and providing recommendations for further research.

Part I

Scientific Article

1

[1]A part of this article has been submitted to The 41st IEEE Conference on Robotics and Automation (ICRA 2024). A part of this article also presents initial results for The International Forum on Aeroelasticity and Structural Dynamics (IFASD) 2024

Robust Tightly-Coupled NMPC and INDI applied to Underactuated Mechanical Systems

Tingyu Zhang

***Abstract*—Underactuated mechanical systems (UMS) feature prominently in robotics and aerospace, with aircraft, unmanned air vehicles, and aeroelastic wings as prime examples. These systems present multifaceted control challenges, ranging from inherent underactuation and stability concerns to state and control saturation and an overarching need for robustness. Crucially, reducing model dependency is a key strategy to enhance control robustness. Nonlinear Model Predictive Control (NMPC) is valued for addressing underactuation and constraints within complex nonlinear dynamics while considering future stages for immediate decision-making. However, implementing NMPC in UMS can pose challenges due to model uncertainties and external disturbances. To enhance NMPC's robustness in UMS control, we introduce a disturbance rejection NMPC strategy, which is tightly coupled with the incremental nonlinear dynamic inversion (INDI), a sensor-based adaptive control approach. The INDI is expected to reject most of the disturbances. Any disturbance residues are managed within a robust NMPC framework through constraint tightening. The efficacy of our method is exemplified through its application to two distinct UMS models. The proposed controller is first customized for a nonlinear aeroelastic system. Compared to the nominal NMPC, the simulation studies demonstrate up to 37.60% and 40.00% error reductions in plunge and pitch motions. Subsequently, we adapt this controller for quadrotor trajectory tracking tasks and compare the results with a benchmark control strategy that loosely coupled the NMPC with INDI. Extensive simulation validations have been performed to track agile trajectories, showing up to a 79.58% reduction in position error and up to a 44.08% reduction in heading error.**

***Index Terms*—nonlinear model predictive control (NMPC), incremental nonlinear dynamic inversion (INDI), aeroelastic system, quadrotor, robust control.**

I. Introduction

Underactuated mechanical systems (UMS) are a class of mechanical models with fewer control inputs than their controlled states. Examples of inherent UMS include aeroelastic systems, aircraft, cranes, underwater vehicles and unmanned air vehicles. Conventional nonlinear control on UMS faces many challenges. These challenges include managing underactuation, handling coupled dynamics, mitigating nonlinearities, preventing constraint violation and control saturation, and establishing robustness against external disturbances and uncertainties.

Robust nonlinear model predictive control (NMPC) with tightened constraints has emerged as an effective remedy for these challenges. In contrast to the min-max NMPC approaches [1], [2], which first maximize the cost function with respect to the worst-case disturbance scenarios and then minimize the maximized cost in relation to control inputs—resulting in high computational complexity—the open loop nominal NMPC with tightened constraints [3] demands fewer computational resources. Under the bounded disturbances, by tightening state and input constraints, robustness in constraint satisfaction is achieved. The efficacy of robust NMPC has been demonstrated through experiments [4]. However, this approach employs the upper bound of disturbances for constraint tightening, potentially resulting in conservative or even infeasible solutions.

The issue of conservativeness was addressed by integrating an integral sliding mode-based disturbance observer into the robust NMPC framework [5]. Additionally, reference [6] introduced a unified disturbance rejection NMPC framework for input-affine UMS. While these innovations aim to reduce uncertainties and conservativeness, they come with their own set of limitations. The sliding manifold in [5] heavily relies on model accuracy, and an estimated disturbance error dynamics is required in [6]. Furthermore, only input-matched disturbances are compensated in these approaches, leaving the orthogonal complement disturbances untouched. This calls for an approach that is less anchored in model dependency and can effectively address both types of disturbances.

Unlike the prevailing model-based controllers and observers, a novel sensor-based approach has drawn attention to controller design. This approach, named Incremental Nonlinear Dynamic Inversion (INDI), can capture both matched and unmatched disturbances. Furthermore, INDI has found widespread applications in the control of quadrotors [7]–[9] and aircraft [11]. However, controller design solely based on INDI cannot address the stability of internal dynamics arising from underactuation, nor can it handle input and state constraints. Consequently, a new approach is desired to explore and integrate the merits of both robust NMPC and INDI.

Recently, a loosely coupled approach combining NMPC with INDI was introduced in [9] for quadrotor trajectory tracking tasks. Within this framework, the inner-loop INDI control improved tracking accuracy beyond the capabilities of the standalone NMPC. Nevertheless, the integrated scheme operates based on a nominal NMPC model, lacking robustness considerations. Notably, INDI is leveraged for only a subset of the dynamics, with no compensatory measures for disturbances in the remaining dynamics. Furthermore, while serving as a control mechanism, INDI does not leverage or disclose disturbances information to NMPC. As a consequence, the system lacks robustness provisions to ensure constraint satisfaction.

This paper has three major contributions. First, we propose a generalized new control approach that tightly couples NMPC with INDI to address disturbances. Such tight coupling exploits the online observation of the sensor-based INDI to

reduce uncertainties in dynamic models utilized by the NMPC. Second, the proposed method is tailored for the control of an aeroelastic system and a quadrotor. For the quadrotor control, to solve the undesired thrust vector tracking challenge, we propose a thrust-direction-aware INDI inner-loop attitude control to correct the thrust direction mismatch. Third, we leverage the uncertainty residues observed from INDI in a robust NMPC framework to tighten the input and state constraints, which effectively avoids violating the original constraints. We have performed extensive simulation validations to demonstrate the advantage of the proposed framework over the pure NMPC method in aeroelastic wing applications. Additionally, our approach outperforms a state-of-the-art controller that loosely coupled NMPC with INDI in quadrotor applications.

II. Preliminary

A. NMPC stability conditions

Most NMPC schemes use a Lyapunov terminal cost and constraints to establish stability [10]. However, constructing such terminal conditions can be challenging, particularly for complex nonlinear systems and time-varying reference trajectories. As a result, in practical applications, it is often preferred to remove the terminal conditions to simplify the controller design and enhance computational efficiency.

For a typical receding horizon optimal control problem without stabilizing terminal conditions, the setup is demonstrated as follows

$$x(k+1) = f(x(k), u(k)), \tag{1}$$

$$V_N(x,u) := \sum_{k=0}^{N-1} l(x(k), u(k)). \tag{2}$$

Equation (1) is the sampled system with sampling time T/N, and (2) is the cost function where $l(x(\cdot), u(\cdot))$ is the stage cost, $x(k) \in \mathbb{X}$ and $u(k) \in \mathbb{U}$ for all $k \geq 0$. Let $\mathbf{u}_N^0$ be the optimal control sequence and V_N^0 be the optimal cost. Only the first control action from the optimal control solution $\kappa_N(x) = \mathbf{u}_N^0(1)$ is applied to the system.

The stability results for the problem above rely on constructing a relaxed descent condition of the optimal cost,

$$V_N^0(x) \geq \alpha l(x, \kappa_N(x)) + V_N^0(f(x, \kappa_N(x))), \ \forall x \in \mathbb{X} \tag{3}$$

where $\alpha \in (0, 1]$. Moreover, the optimal cost and the stage cost are bounded by $\mathcal{K}_\infty$ functions, $\alpha_1(\cdot)$, $\alpha_2(\cdot)$, and $\alpha_3(\cdot)$ ensuring the following inequalities:

$$\alpha_1(|x|) \leq V_N^0(x) \leq \alpha_2(|x|) \tag{4}$$

$$l(x, \kappa_N(x)) \geq \alpha_3(|x|). \tag{5}$$

Consequently, the optimal cost serves as a Lyapunov function. According to the Lyapunov stability theorem, the origin is asymptotically stable for $x \in \mathbb{X}$ with respect to the system.

The main assumptions for stabilising NMPC without terminal conditions are summarized from [16].

Assumption 2.1 (Bounds on the stage cost and infinite horizon cost [16]):

1) There are functions $\alpha_1(\cdot), \alpha_2(\cdot) \in \mathcal{K}_\infty$ such that $l^*(x) := min_{u \in \mathbb{U}} l(x, u)$ satisfies

$$\alpha_1(|x|) \leq l^*(x) \leq \alpha_2(|x|) \tag{6}$$

2) There exists a function $\alpha_3 \in \mathsf{K}_\infty$ such that the optimal infinite horizon cost,

$$V_\infty(x) \leq \alpha_3(|x|), \tag{7}$$

holds for all $x \in \mathbb{X}$

Assumption 2.2 (Bounds on V_N^0 [16]): There exists $\gamma_k > 0$ such that the inequality

$$V_k^0(x) \leq \gamma_k l^*(x) \tag{8}$$

holds for $k = 2,\ 3, \ldots,\ N-1,\ N$, all $x \in \mathbb{X}$. l^* is defined in Assumption 2.1.

Theorem 2.1 (Stability without terminal conditions [16]): Consider the NMPC without terminal conditions satisfies Assumptions 2.1 and 2.2. Let the performance index α_N governed by

$$\alpha_N \geq 1 - \frac{(\gamma_N - 1)\Pi_{k=1}^N(\gamma_k - 1)}{\Pi_{k=2}^N \gamma_k - \Pi_{k=2}^N(\gamma_k - 1)}. \tag{9}$$

If $\alpha_N \in (0, 1]$. Then, the nominal NMPC closed-loop system with NMPC-feedback law $\kappa_N(x)$ is asymptotically stable in $\mathbb{X}$.

The first criterion in Assumption 2.1 is automatically satisfied when the stage cost has the form $x^T Q x + u^T P u$. The second criterion is typically presumed valid. This is based on the general consideration that there always exist input sequences resulting in a bounded cost. In simpler terms, we inherently assume the system to be controllable. Assumption 2.2 can be verified via simulations.

Under the conditions specified in Assumption 2.1 and 2.2, the existence of the performance index, as proposed in Theorem 2.1, guarantees the stability of the closed-loop system. For the proof of Theorem 2.1, see [16]. To interpret the meaning of α_N from Theorem 2.1, let us assume that, for a particular number of stages N, α_N from (9) equals 0.95. Since $1/0.95 \approx 1.053$, we can conclude that for the number of stages equals to N, the performance of the NMPC controller is about 5.3% worse than the performance of infinite horizon controller [16].

III. METHODOLOGIES

This section introduces the proposed generalized robust NMPC scheme with tightened constraints relaxed by the uncertainty reduction using the inner-loop INDI controllers. The diagram of the proposed control scheme is illustrated in Fig. 1, where the constraint-tightening block establishes the connection between NMPC and inner loop INDI controllers.

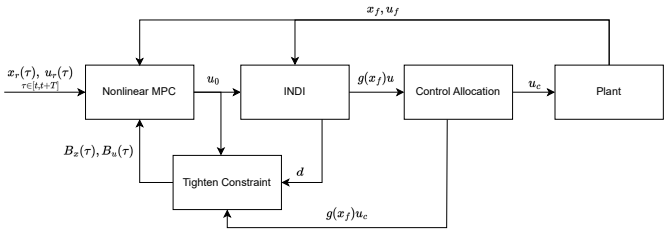


Fig. 1. The control diagram of the proposed robust NMPC with inner loop INDI controller.

A. Constraint-tightened robust NMPC for input-affine systems

We consider a class of nonlinear input affine systems formed by:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(x(t)) + d, \tag{10}$$

where d is an additive perturbation covering modelling and numerical errors, external disturbances, uncertain parameters, etc. We assume that both $f(\cdot)$ and $u(\cdot)$ are continuous. Therefore, the model measurement error can be expressed in the one additive perturbation and be included by d [13]. Furthermore, x and u are in some vector spaces $\mathcal{X}$ and $\mathcal{U}$ constraint by their subspace $\mathbb{X}$ and $\mathbb{U}$. The nominal model is given as:

$$\dot{\bar{x}}(t) = f(\bar{x}(t)) + g(\bar{x}(t))u(\bar{x}(t)). \tag{11}$$

For (11), we have assumed that the dynamics are Lipschitz continuous for $x \in \mathbb{X}$ and $u \in \mathbb{U}$.

Assumption 3.1 (Lipschitz continuous in x): Let the nominal dynamics be denoted as $\dot{\bar{x}} = F(\bar{x}, u)$. There exists a Lipschitz constant $L > 0$, such that for any $\bar{x}_1$, $\bar{x}_2 \in \mathbb{X}$ and any $u \in \mathbb{U}$, $|F(\bar{x}_1, u) - F(\bar{x}_2, u)|_2 \leq L|\bar{x}_1 - \bar{x}_2|_2$.

To perform numerical optimisation, the states and inputs are sampled in N equal time intervals spanning the time horizon T. The constrained optimization problem at any time step k, with state $x(k) = x_0$, is formulated in (12). The initial time index k is omitted for simplicity reasons.

$$\begin{aligned}
&\mathbf{u}_{\text{opt}}(x) = \underset{u}{\operatorname{argmin}} \sum_{i=0}^{N-1} (||\bar{x}_i - x_{i,r}||_{Q_x} + ||u_i - u_{i,r}||_{Q_u}) \\
&\text{subject to } \dot{\bar{x}}(t) = f(\bar{x}(t)) + g(\bar{x}(t))u(\bar{x}(t)), \\
&\bar{x}_i \in \mathbb{X} \sim \mathcal{B}_x(i),\ u_i \in \mathbb{U} \sim \mathcal{B}_u(i)
\end{aligned} \tag{12}$$

In the above optimal control problem, the symbol $\sim$ is the Pontryagin set difference, i.e., Given two sets $A \subseteq \mathbb{R}^n$ and $B \subseteq \mathbb{R}^n$, $A \sim B \triangleq \{z \in \mathbb{R}^n | z + y \in A, y \in B\}$. Sets $\mathcal{B}_x(i)$ and $\mathcal{B}_u(i)$ hold tightened constraint conditions.

Upon solving the optimal control problem defined in (12), the optimal control action, u_0, for the first time step is extracted from $\mathbf{u}_{\text{opt}}$. The state disturbance residue x_{res} is given by

$$x_{\text{res}}(s) = \|d(s) + g(x_f(s))u_c - g(x_f(s))u_0\|_2. \tag{13}$$

where x_f is the low-pass filtered state from sensor measurement. The command input u_c and the estimation of additive disturbance d will be introduced later in Section III-B. The state disturbance residue represents the norm of the difference between the optimal nominal state and the actual state. Furthermore, the input disturbance residue u_{res} is defined as

$$u_{res}(s) = \|u_c(s) - u_0(s)\|_2, \tag{14}$$

which is the norm of the difference between commanded and optimal inputs.

By the Lipschitz condition in Assumption 3.1, the difference between the real and the predicted optimal nominal states, $x(t)$ and $\bar{x}^0(t)$, under the initial conditions $x(t_0) = \bar{x}^0(t_0) = x_0$, can be formulated as

$$\|\bar{x}^0(t) - x(t)\|_2 \leq \int_{t_0}^{t} L\|\bar{x}^0(s) - x(s)\|_2 + x_{\text{res}}(s)ds. \tag{15}$$

The Grownwall-Bellman inequality gives the upper bound of the constraint state error

$$\|\bar{x}^0(t) - x(t)\|_2 \leq \frac{\bar{x}_{\text{res}}}{L}(e^{L(t-t_0)} - 1), \tag{16}$$

where $\bar{x}_{\text{res}} = \sup_{s \in [0,\infty]} x_{\text{res}}(s)$. Motivated by [4]–[6], the state and input constraints are tightened by sets

$$\mathcal{B}_x(k) = \{x \in \mathcal{X} | \|z\|_2 \leq \frac{\bar{x}_{\text{res}}}{L}(e^{L\frac{T}{N}k} - 1)\}, \tag{17}$$

$$\mathcal{B}_u = \{u \in \mathcal{U} | \|u\|_2 \leq \bar{u}_{res}\}, \tag{18}$$

where $\bar{u}_{\text{res}} = \sup_{s \in [0,\infty]} u_{\text{res}}(s)$. Therefore, the robust satisfaction of nominal constraints is guaranteed when the tightened constraints bound the predicted states and inputs. During implementation, the suprema of x_{res} and u_{res} are determined from the previous sampled data using a sliding window approach with a time interval smaller than the one over the cut-off frequency of the low-pass filter for the dynamics. This selection is valid as we assume that the disturbances change slower than the low-pass filtered dynamics.

B. Couple with INDI control

This section introduces INDI-based control to reduce uncertainties. The INDI controller functions as a disturbance observer in the proposed controller framework. The instantaneous additive disturbances can be estimated using the inverse of the dynamics and low-pass filtered states [7]:

$$d = \dot{x}_f - f(x_f) - g(x_f)u_f \tag{19}$$

This external disturbance expression holds true under the assumption that external disturbances are slow-changing relative to the low-pass filtered dynamics. With the optimal control action u_0, the desired state derivative $\dot{x}_d$ is given by

$$\dot{x}_d = f(x_f) + g(x_f)u_0. \tag{20}$$

Then, by substituting the estimated disturbances (19) back into the dynamic equations (10), and replacing $\dot{x}$ with $\dot{x}_d$, we obtain the following incremental expression for the command input:

$$\begin{aligned}
g(x_f)u_c &= \dot{x}_d - \dot{x}_f + g(x_f)u_f \\
&= g(x_f)u_0 - d.
\end{aligned} \tag{21}$$

The explicit solution for u_c from (21) may not exist as $g(x_f)$ is not always in full rank. Therefore, a control allocation strategy is required to obtain u_c. For different models, the control allocation strategies are uniquely defined. The overall goal is to minimize the difference between $g(x_f)u_c$ and $g(x_f)u_0 - d$. In addition, inequality $|g(x_f)u_c - g(x_f)u_0 + d| \leq |\bar{d}|$ should be satisfied such that the command input after INDI and control allocation results in a smaller state disturbance residue compared to the pure NMPC optimal input.

Remark: In Fig. 1, the NMPC is defined by (12), and the tightened constraints are derived in (17) and (18). The command inputs obtained from control allocation aim to minimize state disturbance residues. Because of the disturbance rejection INDI framework, the proposed method will be more robust and less conservative as compared to the robust NMPC in [5]. Additionally, as the state constraints are tightened, our method ensures robustness in constraint satisfaction, which is a challenge unaddressed by INDI.

IV. Application to an Aeroelastic System

A. Aeroelestic model

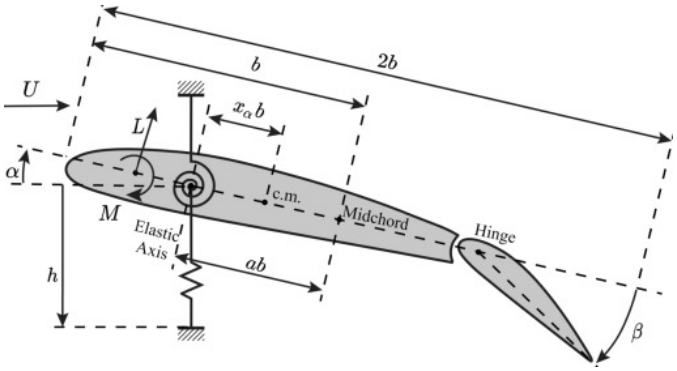


Fig. 2. An underactuated aeroelastic system with two states and one control surface [15].

Figure 2 is taken from [15] and shows a typical aeroelastic wing section with two degrees of freedom and one control input. Its dynamic equation is demonstrated in (22). The states are the plunge displacement h and the pitch angle θ. The control input is the flap angle β.

$$\begin{bmatrix} m_t & m_w x_\alpha b \\ m_w x_\alpha b & I_\alpha \end{bmatrix} \begin{bmatrix} \ddot{h} \\ \ddot{\alpha} \end{bmatrix} + \begin{bmatrix} c_h & 0 \\ 0 & c_\alpha \end{bmatrix} \begin{bmatrix} \dot{h} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} k_h(h) & 0 \\ 0 & k_\alpha(\alpha) \end{bmatrix} \begin{bmatrix} h \\ \alpha \end{bmatrix} = \begin{bmatrix} -L \\ M \end{bmatrix} + d, \tag{22}$$

The additive disturbance is denoted as d. The aerodynamic force L and moment M have the forms:

$$L = \rho U^2 b c_{L\alpha}(\alpha + \frac{\dot{h}}{U} + \bar{a}b\frac{\dot{\alpha}}{U}) + \rho U^2 b c_{L\beta}\beta \tag{23}$$

$$M = \rho U^2 b^2 c_{m\alpha}(\alpha + \frac{\dot{h}}{U} + \bar{a}b\frac{\dot{\alpha}}{U}) + \rho U^2 b^2 c_{m\beta}\beta. \tag{24}$$

From (22), m_t is the total mass of the wing and flap, and m_w is the mass of the wing alone. The wing semichord is denoted as b, and the nondimensional distance from the midchord to the elastic axis is represented by a. x_α is the nondimensional distance between the elastic axis and the centre of mass, and I_α is the moment of inertia. We use c_h, c_α, $k_h(h)$ and $k_\alpha(\alpha)$ to represent the damping and stiffness of plunge and pitch. Furthermore, in (23) and (24), ρ is the air density. $c_{L\alpha}$ and $c_{L\beta}$ are the lift coefficients, and $c_{m\alpha}$ and $c_{m\beta}$ are the moment coefficients. We define $\bar{a}$ as $0.5 - a$. The freestream velocity is denoted by U. For the detailed setup of the parameters, see [15]. The actuator dynamics is modelled as a first-order lag system given by

$$\tau_\beta \dot{\beta} + \beta = \beta_c \tag{25}$$

where $\tau_\beta = 30$ ms and β_c is the command input. The complete state vector has the form $\begin{bmatrix} h & \alpha & \dot{h} & \dot{\alpha} & \beta \end{bmatrix}^T$. We consider the surface deflection limitations. Hence, the constraints are applied to the command input.

B. Constraint-tightened NMPC for the aeroelastic wing

For the aeroelastic wing model, since the constraints only applied to the control input, the state-tightened procedure $x_k \in \mathbb{X} \sim \mathcal{B}_x(k)$ in (12) is dropped. The goal is to steel all states to zero. With state vector $x = \begin{bmatrix} h & \alpha & \dot{h} & \dot{\alpha} & \beta \end{bmatrix}^T$ and control input $u = \beta_c$, the constrained optimization problem started at arbitrary sampling time with initial state x_0 is formulated below

$$\begin{aligned} &\mathbf{u}_{\text{opt}}(x) = \underset{u}{\text{argmin}} \sum_{i=0}^{N-1} (||\bar{x}_i||_{Q_x} + ||u_i||_{Q_u}) \\ &\text{subject to } \bar{x}(i) = \bar{f}(\bar{x}_i, u_i) \qquad \bar{x}(0) = x_0 \\ &u_i \in \{u \in \mathbb{R}^1 | |u| \leq u_b\} \sim \mathcal{B}_u(i) \end{aligned} \tag{26}$$

In the above optimal control problem, the variable $\bar{x}$ is the state of the nominal system, $f(\bar{x}_i, u_i)$, which takes the form of (22) but excludes external disturbances d. The input disturbance residue u_{res} is defined precisely as (14), with the optimal control action u_0 solved from (26).

C. Inner loop INDI and Control Allocation

The instantaneous external disturbances d are estimated using the filtered sensor measured states and (22):

$$\begin{aligned} d(1) =& m_t \ddot{h}_f + m_w x_\alpha b \ddot{\alpha}_f + c_h \dot{h}_f + k_h(h_f) h_f \\ &+ \rho U^2 b c_{L\alpha}(\alpha_f + \frac{\dot{h}_f}{U} + \bar{a}b\frac{\dot{\alpha}_f}{U}) + \rho U^2 b c_{L\beta}\beta_f \end{aligned} \tag{27}$$

$$\begin{aligned} d(2) =& m_w x_\alpha b \ddot{h}_f + I_\alpha \ddot{\alpha}_f + c_\alpha \dot{\alpha}_f + k_\alpha(\alpha_f)\alpha_f \\ &- \rho U^2 b^2 c_{m\alpha}(\alpha_f + \frac{\dot{h}_f}{U} - \bar{a}b\frac{\dot{\alpha}_f}{U}) - \rho U^2 b^2 c_{m\beta}\beta_f. \end{aligned} \tag{28}$$

The desired linear and angular accelerations, $\ddot{h}_d$ and $\ddot{\alpha}_d$, are given by

$$\begin{bmatrix} \ddot{h}_d \\ \ddot{\alpha}_d \end{bmatrix} = \begin{bmatrix} m_t & m_w x_\alpha b \\ m_w x_\alpha b & I_\alpha \end{bmatrix}^{-1} \left(- \begin{bmatrix} c_h & 0 \\ 0 & c_\alpha \end{bmatrix} \begin{bmatrix} \dot{h}_f \\ \dot{\alpha}_f \end{bmatrix} - \begin{bmatrix} k_h(h_f) & 0 \\ 0 & k_\alpha(\alpha_f) \end{bmatrix} \begin{bmatrix} h_f \\ \alpha_f \end{bmatrix} + \begin{bmatrix} -L_0 \\ M_0 \end{bmatrix} \right), \tag{29}$$

where the optimal lift L_0 and moment M_0 are computed from (23) and (24) using the filtered state measurements and the optimal deflection β_0. This deflection angle is derived from the discretized actuator dynamics (30) where k is the current sampling step, and c_1 and c_2 are the coefficients after the discretization.

$$\beta_0 = c_1 \beta_{f,k} + c_2 u_{0,k} \tag{30}$$

According to (21), the desired deflection angle β_d for the aeroelastic wing has the incremental form of

$$\begin{bmatrix} -\rho U^2 b c_{L\beta} \\ \rho U^2 b^2 c_{m\beta} \end{bmatrix} \beta_d = \begin{bmatrix} \ddot{h}_d \\ \ddot{\alpha}_d \end{bmatrix} - \begin{bmatrix} \ddot{h}_f \\ \ddot{\alpha}_f \end{bmatrix} + \begin{bmatrix} -\rho U^2 b c_{L\beta} \\ \rho U^2 b^2 c_{m\beta} \end{bmatrix} \beta_f. \tag{31}$$

Note that directly solving (31) for β_d is not possible as it has only one variable and two equations. An alternative approach involves minimising the state disturbance residue as discussed in Section III-B. Consequently, a quadratic cost function is desiged to determine β_d.

$$\begin{aligned} F &= \begin{bmatrix} -\rho U^2 b c_{L\beta} \\ \rho U^2 b^2 c_{m\beta} \end{bmatrix} \beta - \begin{bmatrix} \ddot{h}_d \\ \ddot{\alpha}_d \end{bmatrix} + \begin{bmatrix} \ddot{h}_f \\ \ddot{\alpha}_f \end{bmatrix} - \begin{bmatrix} -\rho U^2 b c_{L\beta} \\ \rho U^2 b^2 c_{m\beta} \end{bmatrix} \beta_f \\ \beta_d &= \underset{\beta}{\operatorname{argmin}} \ \bar{F}^T k_F \bar{F} + k_\beta (\beta_0 - \beta)^2 \\ &\text{s.t. } |F| \leq |d|, \ |\frac{\beta - c_1 \beta_{f,k}}{c_2}| < |u_b|, \end{aligned} \tag{32}$$

In (32), F is the remaining disturbance. The normalized F by $\begin{bmatrix} -\rho U^2 b c_{L\beta} & \rho U^2 b^2 c_{m\beta} \end{bmatrix}^T$ is indicated as $\bar{F}$. The weight matrices are labelled by k_F and k_β. This control allocation strategy ensures that the remaining disturbance never exceeds the actual disturbance. The second inequality constraint guarantees the satisfaction of input constraint from (26), as the final input command is given by

$$\beta_{c,k} = \frac{\beta_d - c_1 \beta_{f,k}}{c_2}. \tag{33}$$

V. Numerical Validation – Aeroelastic Wing

A. Simulation setup

Simulations are conducted in MATLAB 2022b [17]. The receding horizon optimal control problem is solved using the embedded ACADOS toolbox [18], which employs a sequential quadratic programming (SQP) algorithm. The control allocation described in (32) is accomplished by the MATLAB built-in SQP solver. The NMPC algorithm operates at a frequency of 33.33 Hz with prediction horizon $T = 1$ s and prediction stage $N = 20$. Both the INDI and the control allocation algorithm sample at 100 Hz, which aligns with the sensor frequency. The system dynamics are propagated at a rate of 1000 Hz, and the total simulation time is 8 seconds. All state measurements are filtered by a second-order Butterworth low-pass filter with a 30 Hz cut-off frequency. We have set the sliding window length for identifying the maximum input disturbances residues to 0.03 s, which is marginally less than 1/30 s. The aeroelastic wing specifications are sourced from [15], and additional controller parameters are listed in Table I.

TABLE I
Control Parameters for the Aeroelastic Wing

Parameter	Value
Q_x in (26)	diag([1, 1, 1, 1,1])
Q_u in (26)	1
u_b in (26)	10 deg
k_F in (32)	diag([2, 2])
K_β in (32)	1

B. Nominal case tests

The system's initial state is given by $\begin{bmatrix} -0.02 \text{ m} & 10^\circ & 0 & 0 & 0 \end{bmatrix}^T$. For a freestream velocity of 12 m/s, the system exhibits limit cycle oscillation (LCO), depicted in Fig. 3. The outcomes of our proposed controller are also illustrated in the figure. Using our controller, both pitch and plunge motions stabilize to zero. To highlight the nominal stability, performance indexes for $N \leq 20$ have been computed and are presented in Table II. For $N \geq 18$, we find that $\alpha_N \geq 0$, thus assuring nominal stability as per Theorem 2.1. Furthermore, for $N = 20$, we have $\alpha_N = 0.583$. The closed-loop system is stable, and the performance of the proposed robust NMPC controller is about 151.3% worse than that of the infinite horizon controller.

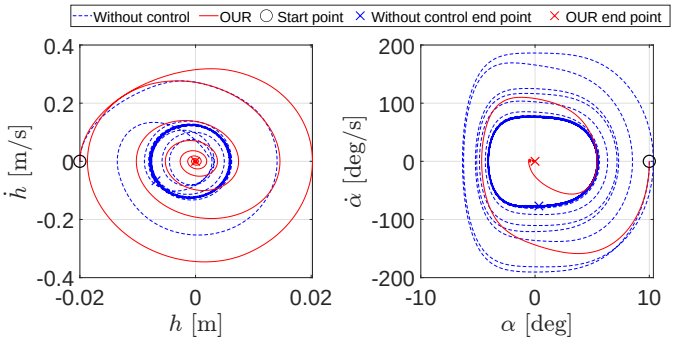


Fig. 3. Phase portrait from the nominal condition at $U = 12$ m/s.

TABLE II
Performance Index for $N \leq 20$

N	18	19	20
α_N	2.07e-1	3.17e-1	3.98e-1

C. Robustness tests

For the robustness test, a sinusoidal disturbance, represented by $A\cos(4\pi t + 0.3\pi)$, was introduced to the freestream velocity, external lift, and external moment separately. In this expression, A denotes the amplitude. Given these disturbances, we simulated the proposed controller's performances under freestream velocities at 12 m/s and 27 m/s. During these tests, the inequalities outlined in (26) related to disturbances, d, were relaxed. This adjustment was made because when one form of disturbance is excited, other forms of disturbance may not be present. In other words, either the lift or moment disturbances

could be small or even zero, rendering the inequality infeasible. To evaluate the effectiveness of the proposed controller, the pure NMPC method was employed as a reference. The root mean square errors for plunge and pitch states can be found in Table III. For most of the simulation cases, our method mitigates the disturbances, yielding reduced errors in both plunge and pitch displacements. Specifically, the error is reduced up to 37.60% for plunge motions and 40.00% for pitch motions.

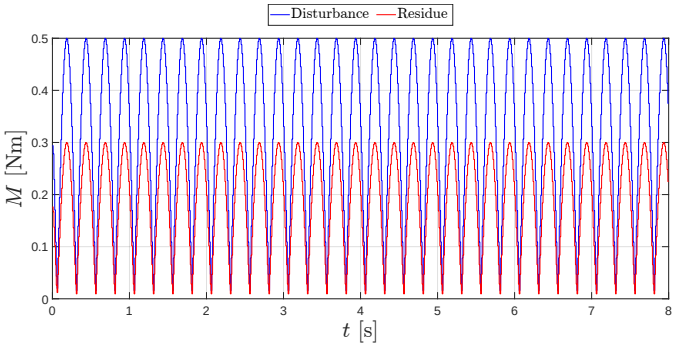


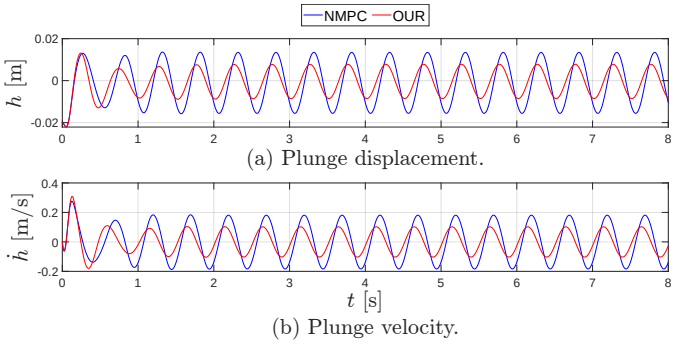
Fig. 4. Magnitudes of moment disturbances and disturbance residuals for $U = 27$ m/s subject to sinusoidal moment disturbances

Figure 4 demonstrates the external moment magnitude using (28) when the wing is subjected to sinusoidal moment disturbances at a speed of 27 m/s. Alongside this, the magnitude of the disturbance residue is shown, indicating a 40% disturbance rejection. The evolution of plunge and pitch trajectories are plotted in Figs. 5 and 6. Figure 7 presents the corresponding command input and real deflection of control surfaces. Owing to the INDI and control allocation strategies detailed in (31) and (32), a significant portion of the disturbances are rejected. In comparison to the results from the nominal NMPC, both the pitch and plunge-related states show smoother fluctuations. Our method has enhanced the accuracy of the plunge displacement and pitch angle by 37.60% and 40.00%. Errors in the rate changes of these states have also been improved by 37.44% and 36.56%. Furthermore, both the command input and the real deflection of the control surfaces remain within control limits and display smoother transitions compared to NMPC outcomes.

D. Limitations

When gust disturbances are introduced, the additive disturbances estimated using (27) and (28) at 12 m/s exhibit more intense fluctuations than those estimated at a speed of 27 m/s, as depicted in Fig. 8. The magnitude of the disturbance residues is illustrated in Fig. 9. Despite these substantial disturbances, the resultant command input manages to keep the disturbance residues relatively low. However, as evidenced in our robustness results under gust disturbances in Table III, a low disturbance residue does not always equate to superior performance.

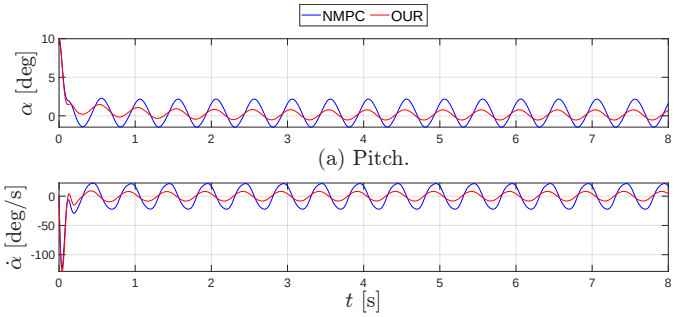
Compensation for aggressive disturbances leads to more volatile control actions, as seen in Fig. 10. Although such



(a) Plunge displacement.

(b) Plunge velocity.

Fig. 5. Evolution of plunge states for $U = 27$ m/s subject to sinusoidal moment disturbances.



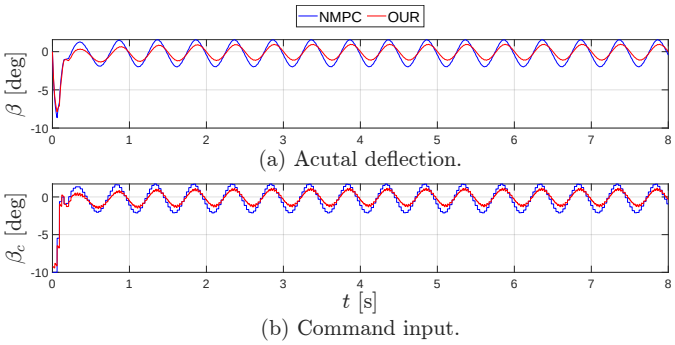
(a) Pitch.

(b) Pitch rate.

Fig. 6. Evolution of pitch states for $U = 27$ m/s subject to sinusoidal moment disturbances.

actions do not exacerbate the error in plunge displacement, they amplify the errors associated with plunge velocity and pitch states. Detailed insights from Fig. 11 demonstrate that our controller produces a larger plunge velocity. From Fig. 12, both the pitch angle and pitch rate exhibit a greater overshoot compared to the NMPC method. Therefore, It is crucial to consider the rate of change in the control inputs. Aggressive control commands that are able to reject disturbances might impair system performance. Addressing this nuanced balance will be vital for the next phases of research.

Overall, the proposed controller showcases robust perfor-



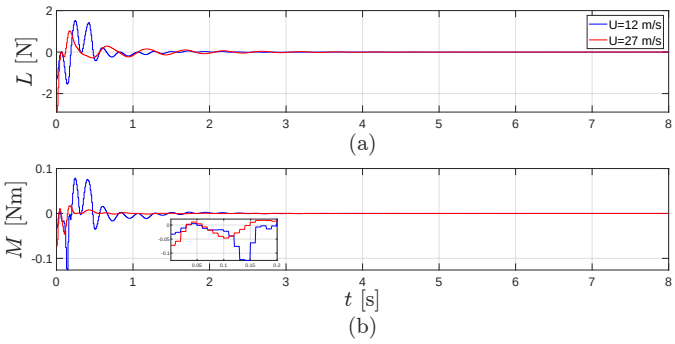
(a) Acutal deflection.

(b) Command input.

Fig. 7. Command and real deflection of the control surface for $U = 27$ m/s subject to sinusoidal moment disturbances.

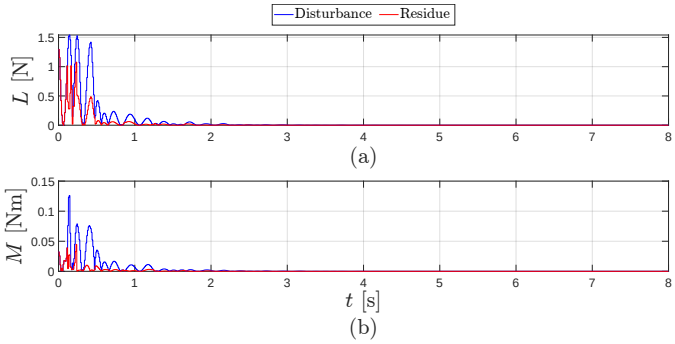
TABLE III
ROBUSTNESS TESTS ON THE AEROELASTIC WING

U [m/s]	Type of Disturbance	RMSE							
		h [m]		α [deg]		$\dot{h}$ [m/s]		$\dot{\alpha}$ [deg/s]	
		OUR	NMPC	OUR	NMPC	OUR	NMPC	OUR	NMPC
12	0.5 [m/s]	4.05e-3	4.05e-3	1.08e0	**9.96e-1**	1.89e-2	**1.74e-2**	2.21e1	**1.99e1**
	5 [N]	**4.87e-3**	4.91e-3	**9.62e-1**	9.87e-2	**7.21e-2**	7.27e-2	**1.94e1**	1.96e1
	0.5 [Nm]	**5.64e-3**	6.17e-3	**1.45e0**	2.01e0	**7.81e-2**	8.89e-2	**2.78e1**	2.82e1
27	0.5 [m/s]	**2.90e-3**	2.94e-3	**8.24e-1**	8.32e-1	**3.38e-2**	3.44e-2	9.95e0	**9.93e0**
	5 [N]	**3.94e-3**	4.00e-3	**7.93e-1**	8.35e-1	**4.82e-2**	4.84e-2	1.01e1	1.01e1
	0.5 [Nm]	**6.49e03**	1.04e-2	**8.94e-1**	1.49e0	**8.07e-2**	1.29e-1	**1.18e1**	1.86e1



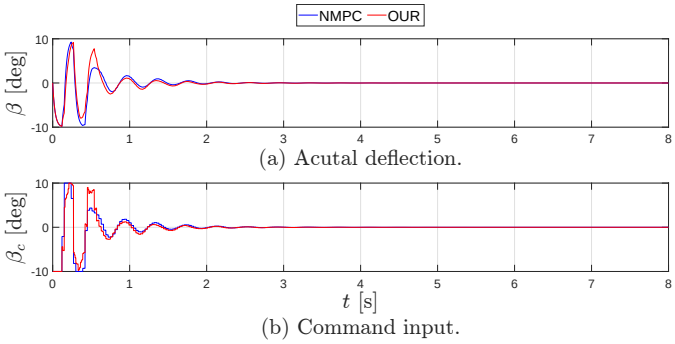
(a)
(b)

Fig. 8. Gust-induced disturbances under different freestream velocities. (a) contains lift disturbances, and (b) shows moment disturbances.



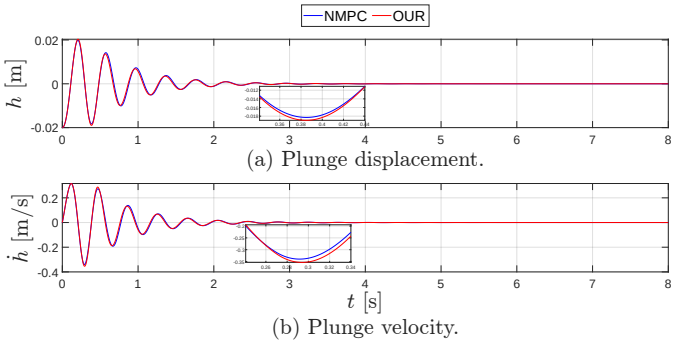
(a)
(b)

Fig. 9. Disturbances and residues at $U = 12$ m/s with 0.5 m/s sinusoidal wind gust. (a) Magnitudes of lift disturbance and residue. (b) Magnitudes of moment disturbance and residue.



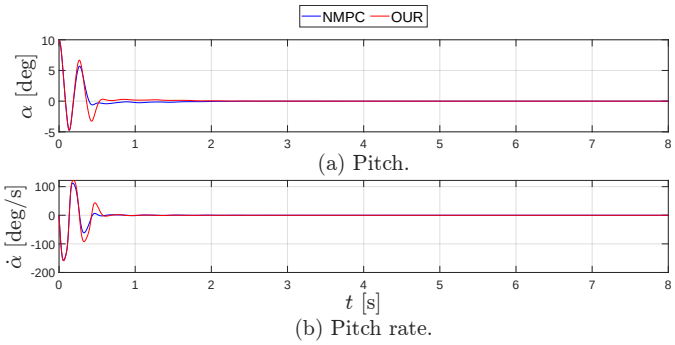
(a) Acutal deflection.
(b) Command input.

Fig. 10. Control command generated by two controllers at $U = 12$ m/s with 0.5 m/s sinusoidal wind gust.



(a) Plunge displacement.
(b) Plunge velocity.

Fig. 11. Evolution of the plunge states by two controllers at $U = 12$ m/s under 0.5 m/s sinusoidal wind gust.



(a) Pitch.
(b) Pitch rate.

Fig. 12. Evolution of the pitch states by two controllers at $U = 12$ m/s under 0.5 m/s sinusoidal wind gust.

mance against isolated force or moment disturbances as well as gust disturbances at higher speeds. While the results presented in this paper are promising, there are avenues for both practical and theoretical enhancements. In the demonstrated cases, the focus remained solely on input constraints. Yet, given the inherent stall angle and structural constraints of an aeroelastic wing, it is important to integrate these as state constraints for future research. Utilizing (17) will then serve to ensure robustness concerning these state constraints.

VI. APPLICATION TO QUADROTOR

A. Quadrotor models

To describe the quadrotor dynamics, two right-handed coordinate frames are utilized. The body frame $\mathcal{F}_B$ with axes x_B, y_B and z_B, has its x-axis pointing forward and the z-axis

aligning with the collective thrust direction. The inertial frame $\mathcal{F}_I$, with axes x_I, y_I and z_I, has its x-axis pointing towards North and the z-axis oppositing to the gravity. With these two frames, the quadrotor translational dynamics are defined as

$$\dot{\zeta} = v \tag{34}$$

$$\dot{v} = \frac{T z_B(q) + f_{\text{ext}}}{m} + g, \tag{35}$$

where ζ and v are position and velocity in the inertial frame, T is the magnitude of the collective thrust and $q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T \in S^3$ is the unit quaternion. The gravitational acceleration and quadrotor mass are denoted by g and m. The external force vector, f_{ext}, includes aerodynamic drags, possible wind gusts and other force disturbances. The rotational dynamics are given by

$$\dot{q} = \frac{1}{2} q \otimes \begin{bmatrix} 0 \\ \Omega_B \end{bmatrix}, \tag{36}$$

$$I_v \dot{\Omega}_B = -\Omega_B \times I_v \Omega_B + \tau + \tau_{ext}, \tag{37}$$

where $\otimes$ represents the quaternion multiplication and $\Omega_B = \begin{bmatrix} \Omega_x & \Omega_y & \Omega_z \end{bmatrix}^T$ is the body frame angular velocity. In equation (37), I_v is the quadrotor inertia matrix and $\tau = \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^T$ is the total torques generated from four rotors.

Lastly, the collective thrust and total torques can be represented by a function of rotor speed $\omega = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 & \omega_4 \end{bmatrix}^T$,

$$\begin{bmatrix} T \\ \tau \end{bmatrix} = G_1 u + G_2 \dot{\omega} + G_3(\Omega_B)\omega, \tag{38}$$

$$u = c_t \omega^{\circ 2}, \tag{39}$$

where u represents the thrust from each rotor, c_t is the thrust coefficient, and $^\circ$ is the Hadamard power. The last two terms from (38), $G_2\dot{\omega}$ and $G_3(\Omega_B)\omega$, account for the torque generated from rotor angular acceleration and gyroscopic effects [9].

B. Constraint-Tightened Robust NMPC Scheme

The state vector for a quadrotor is denoted as $x = \begin{bmatrix} \zeta & v & q & \Omega_B \end{bmatrix}^T$. To perform numerical optimisation, the states and inputs again are sampled in N equal time intervals spanning the time horizon T. The constrained optimization is formulated in (40).

$$\begin{aligned} \mathbf{u}_{\text{opt}}(x) = \underset{u}{\text{argmin}} \sum_{i=0}^{N-1} (||\bar{x}_i - x_{i,r}||_{Q_x} + ||u_i - u_{i,r}||_{Q_u}) \\ + \|\bar{x}_N - x_{N,r}\|_{Q_N} \\ \text{subject to } \bar{x}(i) = \dot{f}(\bar{x}_i, u_i) \qquad \bar{x}(0) = x_0 \\ \Omega_{B,i} \in \{\Omega \in \mathbb{R}^3 | -\Omega_{\max} \le \Omega \le \Omega_{\max}\} \sim \mathcal{B}_\Omega(i) \\ u_i \in \{u \in \mathbb{R}^4 | u_{\min} \le u \le u_{\max}\} \sim \mathcal{B}_u(i) \end{aligned} \tag{40}$$

In the above optimal control problem, $x_{(\cdot,r)}$ is the reference state. The variable $\bar{x}$ is the state of the nominal system, $f(\bar{x}_i, u_i)$, which includes (34) to (39) but excludes external disturbances f_{ext} and τ_{ext}. Furthermore, the nominal model only takes the thrust-generated torques into account, namely the G_1 related term in (38). The G_2 related torques will only be incorporated into the inner loop controller design, following a similar approach as in [9]. Torques related to G_3 are omitted due to their negligible effects as documented in [8]. For this nominal system, we have the Lipschitz-continuous assumption on the constraint states, as presented in Assumption 6.1 widely adopted in robust NMPC [3]–[6]. Sets $\mathcal{B}_\Omega(i)$ and $\mathcal{B}_u(i)$ hold tightened constraint conditions. The quaternion error is defined as $q - q_r = q \otimes q_r^{-1}$, representing the required rotation to transform from the current quaternion state q to the reference quaternion state q_r.

Assumption 6.1 (Lipschitz continuous in Ω): Let the nominal rotational dynamics be denoted as $\dot{\bar{\Omega}}_B = f_\Omega(\bar{\Omega}_B, u)$. There exists a Lipschitz constant $L > 0$, such that for any $\bar{\Omega}_{B,1}$, $\bar{\Omega}_{B,2} \in [-\Omega_{\max}, \Omega_{\max}]$ and any $u \in [u_{\min}, u_{\max}]$, $\|f_\Omega(\bar{\Omega}_{B,1}, u) - f_\Omega(\bar{\Omega}_{B,2}, u)\|_2 \le L\|\bar{\Omega}_{B,1} - \bar{\Omega}_{B,2}\|_2$.

The optimal control action, u_0, obtained from (40) will be used to derive the required thrust and torques, T_{NMPC} and τ_{NMPC}, following (38). The state disturbance residue τ_{res} is given by

$$\tau_{\text{res}}(s) = \|I_v^{-1}(\tau_{\text{ext}}(s) + \tau(\omega_c(s)) - \tau_{\text{NMPC}}(s))\|_2. \tag{41}$$

This represents the norm of the difference between the optimal nominal and the actual angular accelerations. The external torque disturbance is estimated using (53). Furthermore, the input disturbance residue u_{res} is defined as

$$u_{res}(s) = \|u_c(\omega_c(s)) - u_0(s)\|_2, \tag{42}$$

which is the norm of the difference between commanded and optimal rotor thrust inputs, $u_c(\omega_c)$ and u_0. The rotor speed, ω_c, will be derived at the end of Section VI-C in (56).

By the Lipschitz condition in Assumption 6.1, the difference between the real and the predicted optimal nominal angular velocity, $\Omega_B(t)$ and $\bar{\Omega}_B^0(t)$, under the initial conditions $\Omega_B(t_0) = \bar{\Omega}_B^0(t_0) = \Omega_{B,0}$, can be formulated as

$$\|\bar{\Omega}_B^0(t) - \Omega_B(t)\|_2 \le \int_{t_0}^{t} L\|\bar{\Omega}_B^0(s) - \Omega_B(s)\|_2 + \tau_{\text{res}}(s) ds. \tag{43}$$

The Grownwall-Bellman inequality gives the upper bound of the angular speed error

$$\|\bar{\Omega}_B^0(t) - \Omega_B(t)\|_2 \le \frac{\bar{\tau}_{\text{res}}}{L}(e^{L(t-t_0)} - 1), \tag{44}$$

where $\bar{\tau}_{\text{res}} = \sup_{s \in [0,\infty]} \tau_{\text{res}}(s)$. Following (17) and (17), the state and input constraints are tightened by sets

$$\mathcal{B}_\Omega(k) = \{\Omega \in \mathbb{R}^3 | \|\Omega\|_2 \le \frac{\bar{\tau}_{\text{res}}}{L}(e^{L\frac{T}{N}k} - 1)\}, \tag{45}$$

$$\mathcal{B}_u = \{\triangle u \in \mathbb{R}^4 | \|\triangle u\|_2 \le \bar{u}_{res}\}, \tag{46}$$

where $\bar{u}_{\text{res}} = \sup_{s \in [0,\infty]} u_{\text{res}}(s)$. Therefore, the robust satisfaction of nominal angular rate constraints and thrust limitation is guaranteed when the tightened constraints bound the predicted angular rate and optimal control action. The same sliding

window method is utilized to find the maximum of τ_{res} and u_{res}.

C. Inner-loop position and attitude control

This section introduces INDI-based position control and tilt-prioritized INDI attitude control. The former addresses force disturbances by generating a collective thrust command, while the latter focuses on tracking the commanded thrust vector and yaw angle.

The instantaneous force and torque disturbances can be estimated using the inverse of the dynamics and low-pass filtered flight states [7]:

$$T_{\text{ext}} = m(\dot{v}_f - g) - T_f z_B(q_f), \tag{47}$$

$$\tau_{\text{ext}} = I_v \dot{\Omega}_{B,f} - \tau_f + \Omega_{B,f} \times I_v \Omega_{B,f}, \tag{48}$$

where T_f and τ_f are the collective thrust and total torques derived from (49) using filtered rotor speed ω_f:

$$\begin{bmatrix} T_f \\ \tau_f \end{bmatrix} = G_1 \omega_f^{\circ 2} + G_2 \triangle t^{-1} (\omega_{f,k} - \omega_{f,k-1}). \tag{49}$$

The filtered linear and angular accelerations, $\dot{v}_f$ and $\dot{\Omega}_{B,f}$, come from the accelerometer and IMU. These external disturbance expressions hold true under the assumption that external disturbances are slow-changing relative to the low-pass filtered dynamics. The rotor angular acceleration is approximated using the finite difference method where $\triangle t$ is the INDI sampling time and subscript $k-1$ is the last sampled variable. With T_{NMPC} and τ_{NMPC} provided in the previous section, the desired linear and angular accelerations, $\dot{v}_d$ and $\dot{\Omega}_{B,d}$ are given by

$$\dot{v}_d = \frac{T_{\text{NMPC}} z_B(q_f)}{m} + g, \tag{50}$$

$$\dot{\Omega}_{B,d} = I_v^{-1} (\tau_{\text{NMPC}} - \Omega_{B,f} \times (I_v \Omega_{B,f})). \tag{51}$$

Then, by substituting the estimated disturbances (47) and (48) back into the dynamic equations (35) and (37), and replacing $\dot{v}$ by $\dot{v}_d$ from (50), we obtain the following expressions for the command thrust and torques:

$$T_c z_B(q_c) = m(\dot{v}_d - \dot{v}_f) + T_f z_B(q_f), \tag{52}$$

$$\tau_c = \tau_f + I_v(\dot{\Omega}_{B,c} - \dot{\Omega}_f), \tag{53}$$

where $\dot{\Omega}_{B,c}$ is the command angular acceleration and will be given in the latter part of this section. The thrust command T_c is then obtained using

$$T_c = \|T_c z_B(q_c)\|_2. \tag{54}$$

A tilt-prioritized attitude control strategy is employed to track the direction of the commanded thrust and the yaw angle [14]. The quaternion error between commanded and current orientations is defined as $q_e = q_c \otimes q_f^{-1}$, where the components of the quaternion error are $q_e = \begin{bmatrix} q_{e,0} & q_{e,1} & q_{e,2} & q_{e,3} \end{bmatrix}^T$. Subsequently, the reduced quaternion error $q_{e,\text{red}}$ and the yaw error $q_{e,\text{yaw}}$ are formed based on q_e, as detailed in [14, eq.(18), eq.(20)]. The reduced quaternion error corresponds to the shortest rotation to align the current thrust direction to the desired one, while the yaw error is the required rotation to align the current x and y-axis to the desired coordinate frame. Finally, the command angular acceleration is generated by the tilt-prioritized attitude control [14]:

$$\begin{aligned} \dot{\Omega}_{B,c} = {} & 2k_r \hat{q}_{e,\text{red}} \\ & + 2k_y n_y \text{sign}(q_{e,0}) \hat{q}_{e,\text{yaw}} + K_\Omega \Omega_{B,e} + \dot{\Omega}_{B,d} \end{aligned} \tag{55}$$

where k_r, k_y are positive constants, K_Ω is a diagonal positive definite matrix, and $\Omega_{B,e}$ is the angular velocity error expressed in the body frame. The feed-forward angular acceleration, $\dot{\Omega}_{B,d}$, is derived from (51). Vectors, denoted as $\hat{q}_{e,\text{red}}$ and $\hat{q}_{e,\text{yaw}}$, correspond to the last three elements of the quaternion errors $q_{e,\text{red}}$ and $q_{e,\text{yaw}}$.

From (49), (53), and (54), the rotor speed control is obtained by solving a quadratic cost function,

$$\begin{aligned} \omega_c = \underset{\omega_{c,k}}{\text{argmin}} \; & \| \begin{bmatrix} T_c \\ \tau_c \end{bmatrix} - G_1 \omega_{c,k} - G_2 \triangle t^{-1} (\omega_{c,k} - \omega_{c,k-1}) \|_W \\ \text{s.t. } & \omega_{c,k} \in \mathbb{W}, \end{aligned} \tag{56}$$

where $\mathbb{W}$ is the rotor speed constraint and W is a positive-definite diagonal weight matrix. Each diagonal entry denotes the weight for thrust, pitch, roll and yaw channels. The weight for the yaw channel is set to be smaller than the weights for the other channels. This strategy is employed to mitigate control saturation by sacrificing yaw performance, given that the yaw angle has a lesser impact on the quadrotor's stability. The control diagram customized to the quadrotor is plotted in Fig. 13

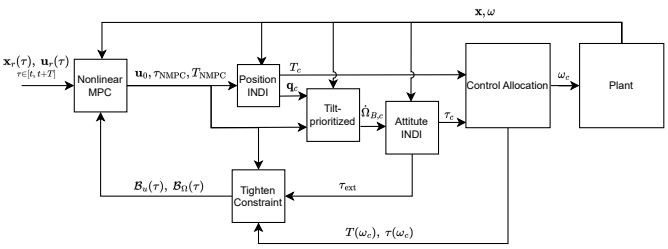


Fig. 13. Control diagram of the proposed controller tailored for a quadrotor. The title-prioritized control block aims to track both the desired thrust vector and desired yaw angle.

VII. Numerical Validation – Quadrotor

A. Simulation setup

Most of the setups are similar to the one explained in Section V-A. For the quadrotor simulations, the NMPC algorithm operates at a frequency of 100 Hz with $T = 1$ s and $N = 20$. The INDI algorithm samples at 250 Hz, the same as the sensor frequency. The system dynamics are propagated at a rate of 500 Hz. The simulation spans 20 seconds. We use the same aerodynamic drag model as in [9] and treat it as an unknown external disturbance. All state measurements are filtered by a second-order Butterworth low-pass filter with a 30 Hz cut-off

frequency [7]. The actuator dynamics is modelled as a first-order low pass filter with a 20 ms time constant. Quadrotor specifics are taken from [9], and other controller parameters are listed in Table IV.

TABLE IV
PARAMETER SETUP

Parameter	Value
Q_x in (40)	diag([200, 200, 500, 1,1,1,0,5,5,200,1,1,1])
Q_u in (40)	diag([6,6,6,6])
k_r, k_y in (55)	3, 2.2
K_Ω in (55)	diag([0.3, 0.3, 0.3])
W in (56)	diag([5,5,5,1])
$u_{\min}$, $u_{\max}$ [N] in (40)	0, 8.5
$\Omega_{\max}$ [rad/s] in (40)	15

The Lipschitz constant L is determined through the Monte Carlo simulation and has been determined to be 17. In these simulations, we randomly generate control inputs and states that satisfy the input and state constraints and compute the corresponding system dynamics. The Lipschitz constant is computed for each pair of inputs and states, and we utilize the maximum Lipschitz constant obtained from a total of 10,000 simulations.

B. Nominal stability

The nominal stability for the closed-loop system is verified through the performance index from Theorem 2.1. Table VII-B

TABLE V
PERFORMANCE INDEX UNDER DIFFERENT SAMPLING STEPS

N	11	12	13	14	15
α_N	6.22e-2	1.76e-1	1.46e-1	2.23e-1	3.43e-1
N	16	17	18	19	20
α_N	4.54e-1	5.09e-1	5.27e-1	5.45e-1	5.61e-1

presents values for α_N for the number of prediction stages up to $N = 20$. A crucial observation is that positive values of α_N, indicative of system stability, commence from $N = 11$ onwards. In our design choice, we have opted for $N = 20$, corresponding to an α_N of 0.561. With this performance index. Under the current system parameters and in nominal operating conditions, we can conclude that the proposed robust NMPC controller ensures stability. However, when compared to an ideal infinite horizon controller, our controller exhibits a 78.3% worsen in performance. This suggests that while the controller is stable and reliable for the chosen parameters, there exists room for optimization to bring its performance closer to that of an infinite horizon controller, such as increasing the prediction length and the number of prediction stages.

C. Tracking agile trajectories

Three sets of pre-planned race tracks (RACE1, RACE2 and RACE3) at speeds up to 20 m/s from [9], [12] were subject to simulation. The simulation results are presented in Table VI, including average position and average heading errors. Both the SOTA method and our approach successfully track all three tracks. Our method outperforms the SOTA in all cases,

TABLE VI
AVERAGE POSITION AND HEADING ERRORS FOR TRACKING AGILE TRAJECTORIES

	Average Position Error [m]		Average Heading Error [°]	
	OUR	SOTA	OUR	SOTA
RACE1	**0.071**	0.087	**5.57**	6.62
RACE2	**0.070**	0.10	**8.12**	8.16
RACE3	**0.074**	0.11	**10.24**	11.53

achieving superior position and angular tracking accuracy. Remarkably, the position-tracking error of our method did not increase as the agility of the race track escalated. The consistency of position tracking error and enhanced angular tracking accuracy highlight the effectiveness of our INDI position control and tilt-prioritized INDI attitude control.

Figure 14 demonstrates the simulation results for the most agile track, RACE3. On the left side of the figure, we

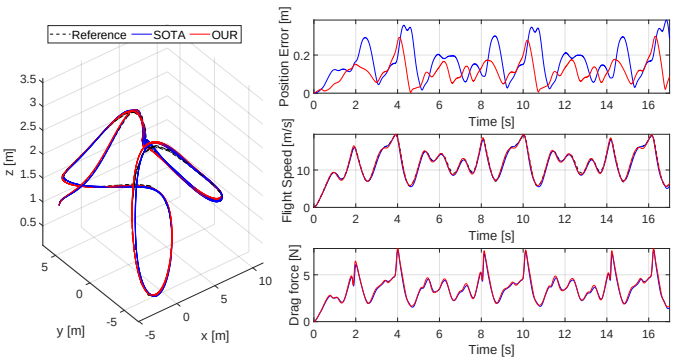


Fig. 14. Tracking performance of RACE3 with a maximum speed up to 20 m/s.

have plotted the simulated trajectories from both approaches alongside the reference trajectory. On the right side, we present the sequential plots for position error, flight speed, and aerodynamic drag forces. Regarding average drag forces, our method and the SOTA approach yield values of 3.62 N and 3.51 N. The maximum drag force reached 7.80 N for our method and 7.65 N for the SOTA. Despite the persistent perturbations caused by drag forces, our approach consistently demonstrates reduced positional error throughout the majority of the simulation duration compared to the SOTA results.

D. Tracking agile trajectory with an unknown Slung load

The race tracks were also employed to assess unknown slung load carrying tasks. The slung load dynamics was modelled as a unilateral mass-spring-damper system featuring a 3 N weight, 1000 N/m linear stiffness, and a 50 Ns/m damping coefficient. In this scenario, the unknown disturbances included drag forces and forces exerted by the slung load.

Table VII summarises the simulation results, where both approaches fail to track RACE3 in this context. Regarding RACE1 and RACE2, our method has improved position tracking performance by 81.25% and 60.00%, respectively. The heading tracking performance is comparable.

TABLE VII
AVERAGE POSITION AND HEADING ERRORS FOR TRACKING AGILE TRAJECTORIES WITH AN UNKNOWN SLUNG LOAD

	Average Position Error [m]		Average Heading Error [°]	
	OUR	SOTA	OUR	SOTA
RACE1	**0.045**	0.24	**7.14**	12.73
RACE2	**0.14**	0.35	15.32	**14.44**

Comparing results from Table VI and Table VII, in the least agile RACE1 case, our method exhibited minimal sensitivity to the slung load with unknown weight. On the contrary, the SOTA method has a significant drop in performance when the slung load is applied. The average heading error has increased by 92.30% for the SOTA approach, while it is only 28.19% for our approach. In the case of RACE2, after applying the slung load, the average position error for our method increased by 100.00%, while the SOTA method exhibited a 250.00% worsening in position tracking. While our method did result in a slightly larger average heading error (6.09%), its superiority in position tracking still makes it favourable for tracking agile trajectories with unknown disturbances.

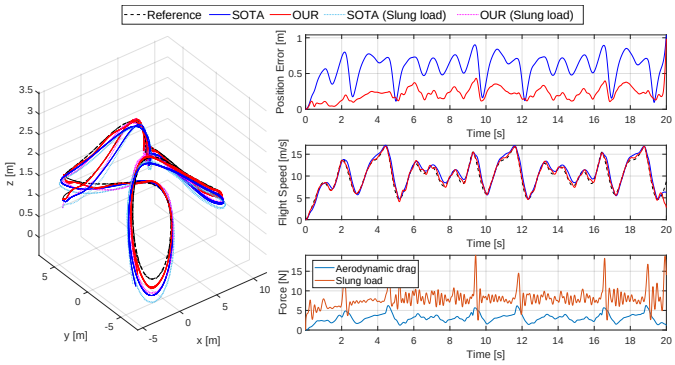


Fig. 15. Tracking performance for RACE2 with unknown slung load attached. The bottom right figure shows the external force disturbance from the aerodynamic drag and the slung load respectively.

Figure 15 displays the simulated trajectories for RACE2 on the left side. The solid lines and the dashed lines represent the trajectories of the quadrotor and the slung load. On the right side, we have plotted the position error, speed, and force disturbances. It's important to highlight that the maximum force disturbance generated by the slung load reached 20 N, yet our method maintained a low position tracking error throughout the entire simulation.

E. Robustness analysis

A total of 40 horizontal and 36 vertical feasible ellipsoidal reference trajectories, as adopted from [9], were utilized to evaluate the robustness of constraint satisfaction under external disturbances. These disturbances are characterized by the form $1 - \cos$, as required by aerospace certification criteria [19]. Distinct tests were conducted to assess robustness against force and torque disturbances. The external force is applied along the x and -z-axis in the inertia frame, while the external torque is exerted in the body frame's x and y directions. The constraint violation criterion triggers if angular velocities exceed the nominal thresholds.

Simulation results for the robustness tests are summarized in Table VIII. The maximum angular velocities observed in these robustness tests are 18.76 rad/s and 19.24 rad/s from our approach and the SOTA method, exceeding 25.07% and 28.33% of the nominal state constraints. Our method consistently showcases improved position tracking performance across all types and all levels of disturbances compared to the SOTA results. As the magnitudes of disturbances increase, both methods exhibit an increase in the average heading error. The SOTA results in smaller heading errors for 0.2 and 0.3 Nm external torques cases. The decline in heading angle tracking performances for our method can be attributed to the tilt-prioritized control strategy employed in the attitude control. Equation (57) are the result after substituting (53) and (55) into (37),

$$I_v\dot{\Omega}_B = -\Omega_B \times I_v\Omega_B + \tau_{\text{NMPC}} + \tau_{\text{tilt}}. \tag{57}$$

$$\tau_{\text{tilt}} = I_v(2k_r\hat{q}_{e,\text{red}} + 2k_y n_y \text{sign}(q_{e,0})\hat{q}_{e,\text{yaw}} + K_\Omega\Omega_{B,e}), \tag{58}$$

where τ_{tilt} consists of first three terms of (55). From (57), the proposed controller eliminates τ_{ext} but induces an artificial term τ_{tilt}. In scenarios with larger disturbances, a larger torque command may be generated from (58) to align the current and desired thrust. This larger torque command will further shrink the size of the tightened state constraint and compromise yaw angle tracking accuracy. In essence, this outcome suggests that the robust thrust control aspect of our method effectively ensures position tracking, albeit at the expense of yaw angle tracking accuracy when facing substantial disturbances.

In our approach, the number of constraint violation cases has decreased when external torques are applied, as compared to the SOTA results. However, when external forces are introduced, we have observed an increase in the instances of constraint violations. This phenomenon can be attributed to two potential factors.

The first factor is analogous to the decline in yaw angle tracking accuracy. The presence of external force disturbances, particularly perpendicular to the thrust direction, induces a larger direction mismatch between the commanded thrust direction and the actual thrust direction. Consequently, the control equation (55) generates a larger command angular acceleration, pushing the angular speed beyond its permissible bounds. Additionally, a larger command angular acceleration is equivalent to a larger $\|\tau_{\text{tilt}}\|_2$ from (58), and the control allocation strategy outlined in (56) may fail to yield an exact zero cost. These two conditions will lead to a larger state disturbance residue defined in (41) and a consequent reduction in the size of the tightened state constraints.

The second contributing factor is linked to the magnitude of the external force disturbance. When a larger force disturbance is presented, it gives rise to a larger input error set $\mathcal{B}_u$ defined in (46). This leads to a reduction in the size of the tightened input constraints. Consequently, the predictive capabilities of

TABLE VIII
ROBUSTNESS TEST RESULTS UNDER EXTERNAL TORQUES AND FORCES

	Position RMSE [m] (mean±std)		Heading Error [°] (mean±std)		Constraint Violation Cases	
	OUR	SOTA	OUR	SOTA	OUR	SOTA
Baseline	**0.14±0.076**	0.17±0.080	**1.90±2.26**	2.30±3.04	6	6
0.1 Nm	**0.14±0.076**	0.17±0.080	**2.84±2.24**	3.06±2.95	**6**	7
0.2 Nm	**0.14±0.076**	0.17±0.080	3.99±3.66	**3.66±3.39**	**6**	7
0.3 Nm	**0.14±0.076**	0.17±0.080	6.78±6.71	**5.67±5.72**	**8**	9
5 N	**0.16±0.063**	0.18±0.066	**2.90±3.17**	3.15±3.69	8	**8**
10 N	**0.19±0.050**	0.22±0.049	**3.71±4.24**	4.17±4.89	**11**	12
15 N	**0.40±1.37**	0.50±1.50	**5.54±7.57**	6.55±8.54	14	**12**

the control framework deteriorate when the tightened state and input constraints become excessively conservative.

To mitigate constraint violations arising from excessively conservative tightened constraints, a potential solution could involve employing the nominal input constraints. This approach can be effective since the control allocation in (56) also ensures satisfaction of input constraints.

VIII. CONCLUSIONS

This work proposed a general framework that deeply couples robust NMPC and INDI controllers. Simulations were conducted on distinct models to evaluate the performance of the proposed method, and the outcomes were compared to other methods from the literature. The effectiveness was tested in two primary scenarios: controlling an aeroelastic wing under diverse disturbances and manoeuvring a quadrotor to trace agile trajectories while subject with an unknown slung payload. The robust state constraint satisfaction is evaluated using extensive simulations. Results show that our method outperforms the selected SOTA and benchmark approaches. In the aeroelastic wing case, both plunge and pitch state errors are reduced. As for the quadrotor case, our approach achieves superior position-tracking performance while maintaining comparable heading angle tracking accuracy. Furthermore, our method exhibits strong robustness when faced with external disturbances and uncertainties.

REFERENCES

[1] H. Chen, C. W. Scherer and F. Allgower, "A game theoretic approach to nonlinear robust receding horizon control of constrained systems," Proceedings of the 1997 American Control Conference (Cat. No.97CH36041), Albuquerque, NM, USA, 1997, pp. 3073-3077 vol.5, doi: 10.1109/ACC.1997.612023.

[2] Magni, L., De Nicolao, G., Scattolini, R. and Allgöwer, F. (2003), Robust model predictive control for nonlinear discrete-time systems. Int. J. Robust Nonlinear Control, 13: 229-246.

[3] D. L. Marruedo, T. Alamo and E. F. Camacho, "Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties," Proceedings of the 41st IEEE Conference on Decision and Control, 2002., Las Vegas, NV, USA, 2002, pp. 4619-4624 vol.4, doi: 10.1109/CDC.2002.1185106.

[4] K. Zhang, Y. Shi and H. Sheng, "Robust Nonlinear Model Predictive Control Based Visual Servoing of Quadrotor UAVs," in IEEE/ASME Transactions on Mechatronics, vol. 26, no. 2, pp. 700-708, April 2021, doi: 10.1109/TMECH.2021.3053267.

[5] M. Rubagotti, D. M. Raimondo, A. Ferrara and L. Magni, "Robust Model Predictive Control With Integral Sliding Mode in Continuous-Time Sampled-Data Nonlinear Systems," in IEEE Transactions on Automatic Control, vol. 56, no. 3, pp. 556-570, March 2011, doi: 10.1109/TAC.2010.2074590.

[6] H. Xie, L. Dai, Y. Lu and Y. Xia, "Disturbance Rejection MPC Framework for Input-Affine Nonlinear Systems," in IEEE Transactions on Automatic Control, vol. 67, no. 12, pp. 6595-6610, Dec. 2022, doi: 10.1109/TAC.2021.3133376.

[7] E. Tal and S. Karaman, "Accurate Tracking of Aggressive Quadrotor Trajectories Using Incremental Nonlinear Dynamic Inversion and Differential Flatness," in IEEE Transactions on Control Systems Technology, vol. 29, no. 3, pp. 1203-1218, May 2021, doi: 10.1109/TCST.2020.3001117.

[8] Smeur E J J, Chu Q, Croon G C H E D, "Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles," Journal of Guidance Control and Dynamics, 39(3):1-12, 2016.

[9] S. Sun, A. Romero, P. Foehn, E. Kaufmann and D. Scaramuzza, "A Comparative Study of NMPC and Differential-Flatness-Based Control for Quadrotor Agile Flight," in IEEE Transactions on Robotics, vol. 38, no. 6, pp. 3357-3373, Dec. 2022, doi: 10.1109/TRO.2022.3177279.

[10] D.Q. Mayne and J.B. Rawlings and C.V. Rao and P.O.M. Scokaert, "Constrained model predictive control: Stability and optimality". In: Automatica 36.6 (2000), pp. 789–814.

[11] Sieberling S, Chu Q P, Mulder J A, "Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction," Journal of Guidance, Control, and Dynamics, 33(6):1732-1742, 2010.

[12] Foehn P., Kaufmann E., Romero A., Penicka R., Sun S., Bauersfeld L., Laengle T., Cioffi G., Song Y., Loquercio A. and Scaramuzza D., "Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight," Science Robotics, 7(67), p.eabl6259, 2022.

[13] Grüne, L., Pannek, J. (2011). Feasibility and Robustness. In: Nonlinear Model Predictive Control. Communications and Control Engineering. Springer, London.

[14] D. Brescianini and R. D'Andrea, "Tilt-Prioritized Quadrocopter Attitude Control," in IEEE Transactions on Control Systems Technology, vol. 28, no. 2, pp. 376-387, March 2020, doi: 10.1109/TCST.2018.2873224.

[15] Bo Sun and Xuerui Wang and Erik-Jan van Kampen, "Event-triggered intelligent critic control with input constraints applied to a nonlinear aeroelastic system," in Aerospace Science and Technology, vol. 120(2022), Article 107279, doi: 10.1016/j.ast.2021.107279.

[16] Grüne, L., Pannek, J. (2011). Stability and Suboptimality Without Stabilizing Constraints. In: Nonlinear Model Predictive Control. Communications and Control Engineering. Springer, London.

[17] The MathWorks, Inc. (2022). MATLAB version: 9.13.0 (R2022b). Accessed: January 01, 2023. Available: https://www.mathworks.com.

[18] Robin Verschueren et al. "acados – a modular open-source framework for fast em- bedded optimal control". In: Mathematical Programming Computation (Oct. 2021). ISSN: 1867-2957. DOI: 10.1007/s12532-021-00208-8.

[19] EASA, "Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes (CS-25)". Amendment 27. November 24, 2021.

Part II

Preliminary Analysis

*This part has been assessed for the course AE4020 Literature Study.

3

Literature Review

3.1. Overview on UMS Control

UMS are systems that have fewer independent control actuators than degrees of freedom to be controlled. For linear UMS, the principles of observability and controllability are aptly encapsulated within their respective matrices. As articulated by Kalman [27], if a linear system is completely controllable for any initial state, there exists a control signal that steers the system to any desired final state. Alternatively, if the system is completely observable, the initial state can be constructed from observing the output. While research on linear UMS is extensive, studies on nonlinear UMS often require a more meticulous, case-by-case analysis, owed largely to their inherent intricacy and variability. An overview of the UMS controller design is listed in Table 3.1.

Table 3.1: Summary of the UMS control

Source	Key Findings	Relation to this study
Olfati-Saber [1]	The author presents an explicit coordinate transformation technique for UMS and proposes a control method for the transformed UMS.	The application of the method is limited because it linearizes the dynamics of the unactuated states only when the number of controlled inputs is greater than or equal to the number of unactuated states.
Hauser, Sastry, and Kokotovic [2]	In their study, an approximate input-output linearization-based control law is proposed for the beam-and-ball system to achieve approximate output tracking. The traditional input-output linearization fails because the beam-and-ball system does not possess a well-defined relative degree.	The approximate nonlinear system does not capture all nonlinearities of the original system.
Olfati-Saber [3]	The author develops a controller for a VTOL aircraft, a UMS with strong input coupling. To address this, the author adopts a coordinate transformation proposed in [1] and applies it to the original system. Backstepping is utilized for the controller design.	The coordinate transformation method is limited to certain types of UMS. Meanwhile, backstepping does not consider state constraints.
Ashrafiuon and Erwin [6]	The paper proposes a stable attitude control method for an underactuated satellite using its appendages.	The controller is specifically designed for the model of a satellite system. It is not applicable to other UMS.
Ríos et al. [7]	Sliding mode techniques are employed for the control of an underactuated helicopter.	The proposed sliding mode control generates high-frequency input signals and causes chattering.
Kalman [27]	The paper introduces the fundamental concepts of controllability and observability in linear systems.	Linear control theory cannot address high nonlinearity problems.

3.2. Modelling and Control of a Quadrotor

The quadrotor stands as a quintessential representation of input-affine UMS. Its versatility is showcased in a myriad of aerial applications, ranging from aerial photography and videography to more critical roles in search and rescue operations and object tracking. Given our intention to first implement the proposed controller on this platform, a profound understanding of quadrotor dynamics and pertinent topics becomes imperative.

A typical quadrotor model based on Euler angles has twelve states: position vector (ζ), velocity vector (v), Euler angle vector (η) and rate of change of the orientation angles in the inertial frame (Ω) where $\zeta,\ v,\ \eta,\ \Omega \in \mathbb{R}^3$. For this model, the control input is the rotor angualr speed (ω), where $\omega \in \mathbb{R}^4$. This model was developed through successive work in papers [28, 29].

However, when using a Yaw-Pitch-Roll Euler sequence for controller design, singularities can arise when the pitch angle (θ) is at 90° or 270°, leading to infinite solutions for roll angle (ϕ) and yaw angle (ψ). Moreover, the associated trigonometric calculations can be computationally intensive. To overcome these issues, Fresk and Nikolakopoulos [30] proposed a full quaternion-based quadrotor attitude control. In their approach, Euler angles are replaced with unit quaternion (q), $q \in \mathbb{R}^4$, which results in thirteen states and four inputs. Simulations and experimental results demonstrated that quaternion-based controllers avoid singularity problems and improve the control scheme's efficiency during tracking tasks. The high efficiency in quaternion computations prompted the proposal of various stable quaternion-based attitude controllers in papers [31, 32, 33], all of which are computationally lightweight. As a result, the quaternion-based quadrotor model is selected for simulations in this study.

The quaternion-based quadrotor model, as synthesized from the literature [30, 31, 32, 33], employs two right-hand coordinate frames: the inertial frame ($\mathcal{F}_I$) and body frame ($\mathcal{F}_B$). Within the inertial frame, the positive z_I axis points opposite of gravity. In the body frame, x_B represents the forward direction of the quadrotor, while z_B is the direction of the collective thrust. Given these frames, the quadrotor translational dynamics are stated as follows,

$$\dot{\zeta} = v, \tag{3.1}$$

$$\dot{v} = \frac{T z_B(q)}{m} + g, \tag{3.2}$$

where m represents the quadrotor's mass, g denotes the gravitational constant, and T signifies the collective thrust.

The rotational kinematic in the quaternion expression is

$$\dot{q} = \frac{1}{2} q \otimes \begin{bmatrix} 0 \\ \Omega_B, \end{bmatrix} \tag{3.3}$$

where Ω_B is the angular velocity expressed in the body frame, and $\otimes$ is the quaternion multiplication operator. Lastly, the rotational dynamic equation is

$$I_v \dot{\Omega}_B = -\Omega_B \times I_v \Omega_B + \tau, \tag{3.4}$$

where I_v is the inertia matrix and τ is the resultant torque from four rotors. The collective thrust and torque generated by rotors are functions of rotor angualr speed (ω) and accelerations ($\dot{\omega}$), and they are shown in Eq. (3.5).

$$\begin{bmatrix} T \\ \tau \end{bmatrix} = G_1 u + G_2 \dot{\omega} + G_3(\Omega_B)\omega \tag{3.5}$$

In Eq. (3.5), $u = c_t \omega^{\circ 2}$ depicts the thurst from each rotor, and $^\circ$ is the Hadamard power. G_1, G_2 and G_3 are four by four matrices. The terms $G_2\dot{\omega}$ and $G_3(\Omega_B)\omega$ account for the torque generated from rotor angular acceleration and gyroscopic effects. Notably, these two terms are omitted in the nominal model as the angular acceleration-related torque will be compensated by the controller design, and the torques due to gyroscopic effects are often negligible.

Given a disturbance-free environment and a second-order continuous reference trajectory function

denoted as $\zeta = \mathcal{F}_{\text{ref}}(t)$, alongside a specified yaw angle $\psi_{\text{ref}}(t)$, one can compute the nominal control as:

$$R_i^b \begin{bmatrix} a_x \\ a_y \\ a_z + g \end{bmatrix} = \frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}, \tag{3.6}$$

$$R_i^b = R_x R_y R_z = \begin{bmatrix} C_\theta C_\psi & S_\psi C_\theta & -S_\theta \\ S_\phi S_\theta C_\psi - C_\phi S_\psi & C_\phi C_\psi + S_\phi S_\theta S_\psi & S_\phi C_\theta \\ S_\phi S_\psi + C_\phi S_\theta C_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi & C_\phi C_\theta \end{bmatrix}, \tag{3.7}$$

where R_i^b is the rotational matrix defined in Eq. (3.7) and $\begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T$ is the desired linear acceleration calculated from the second derivative of the reference trajectory. Simplifications $C.$ and $S.$ denote cos$(\cdot)$ and sin$(\cdot)$ respectively. Following algebraic manipulations, the closed-form solution for pitch (θ) and roll (ϕ) angles are

$$\theta_d = \text{atan2}\left(\frac{a_x \text{cos}(\psi_{\text{ref}}) + a_y \,\text{sin}(\psi_{\text{ref}})}{a_z + g}\right), \tag{3.8}$$

$$\phi_d = \text{atan2}\left(\frac{\text{cos}(\theta_d)(a_x \,\text{sin}(\psi_{\text{ref}}) - a_y \,\text{cos}(\psi_{\text{ref}}))}{a_z + g}\right). \tag{3.9}$$

Should the first and second-order derivatives of the Euler angles be accessible, we can directly compute desired angular velocities and accelerations. As a result, desired collective thrust (T) and torque (τ) emerge from the amalgamation of desired linear accelerations, angular velocities, and angular accelerations.

To evaluate the robustness of the developed controller, simulations often introduce external disturbances. These disturbances may encompass aerodynamic drag forces induced by air resistance, as well as additive forces and moments resulting from wind gusts. As detailed in [26], an effective linear aerodynamic drag model is employed. In alignment with aerospace certification standards [34], these additive disturbances are typically characterized using the $1 - \cos$ formulation. In addition to external disturbances, model uncertainties are also considered during robustness testing. These uncertainties typically arise in the G_1, G_2 and I_v matrices from Eq. (3.4) and Eq. (3.5). Model uncertainties can also include measurement errors in the quadrotor's mass.

This subsection delves into an in-depth review of quadrotor models, both based on Euler angles and quaternions. Notably, quaternion-based controllers are deemed superior in terms of computational efficiency and are exempt from the singularity problems inherent to Euler angles [30, 31, 32, 33]. Given these advantages, our research will adopt the quaternion-based quadrotor model. Additionally, we discuss disturbance models with the intention of incorporating them into our simulations, ensuring a thorough evaluation of controller robustness. These disturbances account for factors such as aerodynamic drag forces caused by air resistance, model uncertainties, and external forces and moments due to wind gusts. The papers discussed herein are summarized in Table 3.2 for easy reference and a better understanding of the related literature.

Table 3.2: Summary of quadrotor models and control methods

Source	Key Findings	Relation to this study
Sun et al. [26]	The authors conduct a performance comparison between two quadrotor control schemes. The disturbances considered in the study encompass constant external forces and moments, model uncertainties, and aerodynamic drags. The INDI inner loop attitude control enhances the robustness.	The robustness in state satisfaction is not considered. The robustness in position tracking is not addressed as the INDI is only implemented for attitude control.
Bouabdallah, Murrieri, and Siegwart [28]	A quadrotor model based on Euler angles is developed in the reviewed paper. Stabilizable attitude control based on the control Lyapunov function is proposed, and a stable height control is established using the linearization method.	Euler angle has a singularity problem. The linearization method for height control does not capture the model's nonlinearities.
Bouabdallah and Siegwart [29]	In the reviewed paper, the quadrotor model introduced in [28] is extended to incorporate rotor dynamics, including the propeller and gearbox models. To control the proposed quadrotor model effectively, backstepping and sliding-mode techniques are applied.	The sliding mode technique introduces high frequency, low amplitude vibrations and leads to sensor drift. The backstepping control does not address the zero dynamic problem. Singularity due to the use of Euler angle is not avoided.
Fresk and Nikolakopoulos [30]	The translational and Rotational dynamics in quaternion forms are derived from the Euler Newton equations. To achieve attitude control, A quaternion-based nonlinear proportional squared control is proposed.	Quaternion-based quadrotor model and attitude control avoid singularity and require less computational power compared to using the Euler angles. However, this paper only designed an attitude controller, which can not address the coupled dynamics of the quadrotor.
Brescianini, Hehn, and D'Andrea [31]	The paper presents a unit quaternion-based attitude controller and proposes a method to prioritize the tilt control.	The stability of the proposed controller is not derived.
Mueller [32]	The author provides a comprehensive review of the current state-of-the-art quadrotor attitude controllers and proposes a high-performance quaternion-based attitude control law.	One attitude control does not address the coupled dynamics between rotation and translation. Integrating a position control is urgent.
Brescianini and D'Andrea [33]	The authors extend the quaternion-based tilt priority controller proposed in [31] by incorporating a control allocation strategy.	To achieve both position and attitude control, it requires an integration with the state-of-the-art position control method.

3.3. Linear Control

In this section, traditional linear control methods will be reviewed, as they are widely favoured in the aerospace industry due to the comprehensive linear system theory. The first step in controller design is often to linearize the nonlinear system around an operating point, followed by applying linear controller design approaches. However, this approach is only practical for a limited operating range, and linear controller design based on local approximations may be invalid for a wide range of operating conditions. More critically, neglecting nonlinear terms in the system's dynamics after linearization fails to ensure global stability.

One advanced linear control method that addresses part of the model's nonlinearity is the linear parameter varying (LPV) model. LPV models are linear state-space models with state and time-dependent matrices, facilitating the utilization of linear control strategies. These models incorporate an unknown state and time-dependent parameter known as a scheduling signal, which can be measured or designed based on physical insight or engineering experience. By properly selecting scheduling variables, the performance of linear controllers can be improved. While LPV models handle state-dependent nonlinearity adeptly, they struggle with hard nonlinearities such as stiction, hysteresis, saturation and nonlinearly coupled modes [8]. The stability of LPV relies on the stability theory of the Linear Time-varying (LTV) system [35]. Most stability and performance analyses were proven through experiments and high-fidelity simulations. Only a few papers have implemented LPV to control UMS, such as an underactuated ship [9] and a robot uARM [10].

In conclusion, the LPV approach tackles state-oriented nonlinearities but fails to address state and input constraints. While the advantages inherent to LPV are palpable, its stability conditions require the existence of specific Lyapunov functions [36], which can be computationally complex. Furthermore, its capability to solve underactuation challenges remains nascent. Table 3.3 summarizes the papers discussed in this subsection.

Table 3.3: Summary of the linear control

Source	Key Findings	Relation to this study
Hoffmann and Werner [8]	Tha paper is a survey that highlights the application results of LPV control. LPV control is favoured when there is known physical insight or engineering experience.	In the context of our study, the LPV model is not suitable because it cannot accurately capture or handle state and input constraints.
Liu, Guo, and Yuan [9]	The authors propose an integrated LPV control tailored for an underactuated ship.	LPV is more of a modelling technique and cannot capture all nonlinearities.
Siqueira and Terra [10]	The authors propose a robust control scheme for an underactuated wheeled mobile robot based on the quasi-LPV representation and game theory.	The stability conditions for this quasi-LPV representation using the H infinity approach are not addressed in the paper.
Zhang, Tsiotras, and Knospe [35]	The authors propose computationally tractable stability criteria designed for LPV time-delayed systems.	Stability tests are cast as solvable LMI convex optimization problems. However, the computational complexity of solving the LMIs is a challenge.
Shamma [36]	The author provides a comprehensive review of various concepts and analytical approaches for LPV systems.	The stability of an arbitrary time variation LPV system has high computational complexity. The theoretical stability analysis of LPV control can be challenging and elusive.

3.4. Nonlinear Control

Relative to linear control, nonlinear control has natural advantages in addressing nonlinearity. However, it also comes with its own drawbacks. Conventional nonlinear control methods often fall short in solving global or semi-global stabilization challenges stemming from underactuation, known as zero-dynamics. The zero-dynamics is a concept associated with a feedback linearization technique to nonlinear controller design. This technique can transform nonlinear systems with well-defined relative degrees into controllable canonical forms. However, after the transformation, a part of the dynamics loses its relationship with the input and becomes unobservable. These unobserved states are called internal states, and the zero-dynamics is defined as the system's internal dynamics when the system outputs are kept at zero. The efficacy of controller designs utilizing nonlinear dynamics inversion (NDI) or nonlinear backstepping (NBS) hinges on the internal dynamics' stability. Ensuring global stability for the internal dynamics is challenging, and in many cases, only local stability is guaranteed, even if the zero-dynamics is globally exponentially stable. Few attempts have been made to stabilize the internal dynamics. Getz [4] designed an internal equilibrium controller under the nonlinear dynamic inversion framework, and Bencsik, Kovács, and Zelei [37] proposed using periodic servo-constraints to overcome stability problems and enhance dynamic behaviour.

Applying traditional nonlinear control strategies to UMS requires obtaining specific dynamic structures. Two coordinate transformation methods, known as Collocated partial feedback linearization [5] and Noncollocated partial feedback linearization [1], can transform the UMS into cascade normal forms. However, the first method highly complicated the controller design by coupling linear and nonlinear subsystems with a new control input. The second method linearizes the dynamics of the unactuated states only when the number of the controlled inputs is greater or equal to the number of unactuated states. Additionally, a notable limitation of traditional nonlinear controls, such as NDI and NBS, is their high model dependency, making them susceptible to model inaccuracies and external perturbations. A summary of the mentioned nonlinear control techniques is provided in Table 3.4.

Table 3.4: Summary of the nonlinear control

Source	Key Findings	Relation to this study
Getz [4]	In this paper, the author proposes an internal equilibrium controller that maintains internal stability. The proposed controller approximately tracks the reference trajectory and keeps the internal dynamic stable.	Only approximate tracking is achieved, and error does not converge to zero asymptotically.
Spong [5]	The author introduces the concept of collocated partial feedback linearization.	The proposed method may highly complicate the controller design by coupling linear and nonlinear subsystems with a new control input.
Bencsik, Kovács, and Zelei [37]	Periodic servo constraints are presented to stabilize the internal dynamics.	The proposed method is highly model-based. The robustness of the proposed controller needs to be validated.

3.5. Sliding Mode Control

Control techniques for UMS remain a frontier of exploration, especially in unpredictable and intricate environments. As industries demand high resilience against unpredictable factors such as noise and disturbances, the race to devise superior control tools intensifies. In this landscape, sliding mode control (SMC) emerges as a promising contender, lauded for its resilience against minor, unstructured uncertainties and its unwavering performance amidst parameter variations and external disturbances. SMC has varied applications, from manoeuvring underactuated satellites [6] and helicopters [7] to the precision control of tower cranes [38]. Yet, no solution is without its drawbacks. Determining the most effective sliding mode control surface can be difficult. Moreover, SMC induces high-frequency oscillations around the state trajectories, known as ”chattering” [39]. To address this issue, innovations like reaching law-based SMC [40] are emerging as solutions. There's also a suite of alternative strategies, such as quasi-sliding-mode control strategies [41, 42], higher order SMC [43], and fuzzy SMC [44, 45], all designed to attenuate chattering. But innovation comes with its challenges. These new solutions add additional complexity to controller designs and are limited by the control objectives. For example, the higher-order SMC is not suitable in the context of finite-time convergence [46]. In essence, while the designed control surface ensures stability and robustness, perfecting its implementation, especially in minimising chattering, increases the complexity of SMC design. An overview of the referenced papers is provided in Table 3.5.

Table 3.5: Summary of the sliding mode control

Source	Key Findings	Relation to this study
Van Trieu et al. [38] and Utkin [39]	A fault-tolerant SMC is proposed and implemented on an underactuated tower crane system [38] . The paper explores the applications of SMC in a manipulator control system [39]. SMC design offers a simpler design approach and is less reliant on detailed system knowledge	The major problem for SMC is the high-frequency oscillation.
Mija and Thomas [40], Bartoszewicz [41], and Adamiak [42],	A reaching law approach for SMC is proposed to reduce chattering [40]. Researchers introduce Quasi-Sliding Mode Control (SMC) strategies as a means to mitigate chattering [41, 42].	Most chattering reduction techniques do not consider state constraints.
Kachroo [43]	The authors propose the incorporation of a boundary layer and a continuous control scheme within the boundary layer into the Sliding Mode Control (SMC) framework. The SMC with a boundary layer results in chattering-free systems.	This is a different approach using a boundary layer in SMC for chattering reduction. Still, other controller design considerations, such as state constraints, are not considered.
Wai [44] and Ha et al. [45]	Both papers develop fuzzy SMC systems, in which a fuzzy hitting control law is embedded into the conventional SMC system to remove the control chattering.	The fuzzy SMC may require larger control gains, causing control saturation.
Utkin [46]	The author compares higher order SMC with conventional SMC with respect to finite time convergence, chattering suppression, relative degree, and other fields. The conventional SMC is simpler and has similar performances as the higher-order SMC in finite-time convergence.	Higher order SMC can be used for chattering reduction but has higher design complexity.

3.6. Incremental Nonlinear Control

Incremental nonlinear control approaches are potential solutions to the high model dependency prevalent in traditional control methods. State-of-the-art methods include Incremental nonlinear dynamic inversion (INDI), incremental nonlinear backstepping (INBS) and incremental sliding mode control (ISMC). All utilize additional measurements to replace part of the model information.

Incremental nonlinear control involves taking the first-order Taylor expansion of a nonlinear dynamic system. Under the time-scale separation assumption, which assumes the state-related first-order term is negligible, the physical control input can be computed using the previous input and the incremental input. INDI, for instance, has demonstrated its robustness and stability in control for aircraft, as supported by simulations and in-depth theories outlined in [16]. Its efficacy is further illustrated in quadrotor control scenarios. Smeur, Croon, and Chu [11] employed INDI controllers for both attitude and position control on a quadrotor, achieving remarkable improvements over traditional Proportional Integral Derivative (PID) controllers. Moreover, Tal and Karaman [12] explored the integration of INDI controllers with differential flatness techniques for quadrotor aggressive trajectory tracking. Unlike INDI, the studies of closed-loop effect of uncertainty for INBS have been confined to simulations and experiments [13, 14]. Nonetheless, recent work by Jeon et al. [17] offers theoretical interpretations on the impacts of the one degree of freedom uncertainty for INBS. The study by Wang et al. [15] is notable for introducing an ISMC framework optimized for fault-tolerant flight control, marking an improvement in robustness and reduced model dependence.

Yet, these incremental controllers are not without challenges. Both INDI and INBS grapple with stability concerns tied to the zero-dynamics, an issue brought by underactuation. Specifically, the matrix inversion required by INDI may not have full rank due to underactuation. INBS often mandates a system transformation into a chain structure -— a process not always feasible, especially for UMS like flexible aircraft. Also, a common limitation is the disregard of state constraints by these incremental controls.

In summary, while incremental nonlinear control is promising in curtailing model dependency and bolstering robustness, it doesn’t comprehensively address the challenges of underactuated systems and state constraints. The cited papers are summarized in Table 3.6.

Table 3.6: Summary of the incremental nonlinear control

Source	Key Findings	Relation to this study
Smeur, Croon, and Chu [11]	The INDI controllers are proposed for both attitude and position control of a micro air vehicle.	The cascade INDI framework proposed in the paper demonstrates a systematic approach to incorporating INDI into the control of UMS. However, the instability can be triggered by control saturation.
Tal and Karaman [12]	The authors proposed a control law that combines INDI with the differential flatness technique to achieve quadrotor trajectory tracking.	The incremental forms used in position and attitude control do not address the coupled dynamics.
Acquatella, Kampen, Chu, et al. [13]	The authors propose an incremental backstepping strategy for flight control.	The incremental nonlinear control approach exhibits high robustness and reduced model dependencies, making it an attractive choice for our study, especially under the assumption of low sensor error.
Wang and Van Kampen [14]	A synchronization of Incremental Backstepping and Sliding Mode Control is proposed for nonlinear strict-feedback systems	This study demonstrates the application of INBS on UMS due to actuator failure. However, it requires the system to have a strict-feedback form and does not consider the state constraints.
Wang et al. [15]	An incremental sliding-mode control framework is introduced for fault-tolerant control.	This study demonstrates the application of ISMC on UMS. There is still a research gap on the robustness of constraint satisfaction.
Wang et al. [16]	The stability and robustness criteria of INDI control are presented in the literature. The closed loop system under INDI control is ultimately bounded by a $\mathcal{K}_\infty$ function of perturbation bounds. Furthermore, INDI has better robustness compared to NDI.	INDI has less model dependency and higher robustness, but it cannot stabilize the zero dynamics.
Jeon et al. [17]	A closed-loop analysis using incremental backstepping is given in this paper.	Only the impacts of one degree of freedom uncertainty are analyzed.

3.7. Optimal Control

Recent research spotlighting optimal control for nonlinear UMS stems from its advanced prediction capabilities and adeptness at handling constraints. At its core, the optimal control takes a control system (plant) and creates a cost function with each possible behaviour. The control law/action is determined by minimising the designed cost function, wherein the underactuation problem, which involves assigning control of many states to a few controllers, is automatically addressed during the minimisation procedure. Dynamic Programming (DP) is a common approach to solving the optimal control problem. However, DP often turns out to be computationally expensive and, for nonlinear systems, might not always yield explicit solutions. Nonetheless, DP solutions provide valuable insights into the control of complex systems, including the formulation of stability and convergence criteria.

One special exemplar in optimal control is the Linear Quadratic Regulator (LQR), which is characterized by linear systems and a quadratic cost function. The optimal feedback control law for LQR, derived through iterative DP, can be discerned either from the Riccati differential equation (for finite-horizon LQR) or its algebraic counterpart (for infinite-horizon LQR). Moreover, under the assumption that the system is observable, the DP solutions for infinite-horizon LQR asymptotically stabilize the origin for the closed-loop system [47]. Infinite-horizon LQR demonstrates excellent stability margins when input weight is a scalar multiplying the identity matrix [48]. Despite its merits, the real-time application of infinite-horizon LQR hits computational barriers due to its inherent infinite steps. From a practical standpoint, finite-horizon LQR is often preferred for implementation. The finite-horizon LQR solution for a controllable system is stable and converges to the infinite-horizon LQR solution as the horizon time limits to infinity [49]. In scenarios demanding an extended tracking horizon, optimizing the actions for the entire time horizon becomes heavy, and thus, Model Predictive Control (MPC) comes into play. MPC predicts the future states within a fixed horizon, which is smaller than the horizon that needs to be optimized. Only the input from the first step is applied to the real plant. This process is repeated at each time step, allowing the total horizon under consideration to move forward in time. Linear MPC is a generalization of LQR, and its DP solution is equivalent to the solution obtained from the finite-horizon LQR version when disturbances are absent.

Methods such as finite horizon LQR and linear MPC only stabilize the local trajectory of the nonlinear UMS. This is where Nonlinear MPC (NMPC) stands out, particularly in managing nonlinearity. Although applying DP directly to the NMPC problems can be computationally complex, researchers have derived sufficient stability conditions for NMPC using the principle of DP. NMPC can be categorized into four distinct models: NMPC with terminal cost and constraints, NMPC with terminal equality constraints, NMPC with only terminal cost, and NMPC without stabilizing terminal conditions. Before delving into the specifics of each model, it is important to address the concept of the domain of attraction, which refers to the set of states that can reach the terminal region with steps equal to or less than the predicted horizon.

The first model confines the domain of attraction within the feasible set by using a predefined terminal cost and constraint set. Its stability has been reviewed by Mayne et al. [18] through two approaches from the literature. The first approach, known as the direct approach, employs the cost function as a Lyapunov function, while the second approach utilizes the monotonicity property of the cost function. Gilbert and Tan [50] have designed an algorithm to approximate a maximal invariant control admissible set used for the terminal constraint set through the linearized model. The NMPC with equality terminal constraint can be viewed as a special case of the first model, and its stability conditions have been analysed in [19]. Although predefined terminal cost and constraints confine the domain of attraction to the feasible set and guarantee recursive feasibility, ensuring that the online optimization problem has a feasible solution, in real-time implementation, terminal constraints are often omitted for ease in design purposes.

Constructing the Lyapunov function as a terminal cost is challenging for higher-order nonlinear systems. Limon et al. [20] demonstrated that for NMPC with terminal cost but without terminal constraint, a sufficiently large weight of the terminal cost could enlarge the unconstrained region of attraction to the same size as the terminal constraint NMPC. Furthermore, they presented a practical procedure for calculating the stabilizing weighting factor of the terminal cost.

The last NMPC model simplifies controller design and is readily applicable without precomputation, such as constructing terminal costs or constraints. Its sufficient stability conditions can be verified after simulations and are detailed in [21]. However, this model may require a larger horizon to ensure asymptotic stability. When comparing all four NMPC models regarding design difficulty and stability, the NMPC model without stabilization terminal conditions emerges as the preferred choice, largely due to its diminished

terminal costs and constraints. The remaining three models can serve as backup plans when feasibility and stability are not guaranteed.

To encapsulate, the use of optimal control methods, particularly NMPC, provides promising avenues for addressing the complexities of nonlinear UMS and offers stability and control insights that are valuable in practical implementations. The drawbacks are its higher computational costs and model dependency. The literature mentioned in this section is summarized in Table 3.7.

Table 3.7: Summary of the optimal control

Source	Key Findings	Relation to this study
Mayne et al. [18]	This paper serves as a comprehensive survey on model predictive control, with a particular focus on analyzing stability and optimality aspects.	The positive invariant terminal constraints are difficult to construct and require a high computational cost. The nominal stability does not ensure robust stability.
Magni and Sepulchre [19]	Under specific assumptions, optimal control laws derived from the nonlinear receding horizon optimal control problem are optimal with respect to a modified infinite horizon optimal control problem.	The inverse optimal property demonstrates that certain forms of receding horizon control possess stability margins. Computing such margins is still an open question.
Limon et al. [20]	The paper defines stability criteria for model predictive control in the absence of terminal constraints. The domain of attraction of the NMPC can be enlarged using a stabilizing terminal weight factor.	The design of NMPC without terminal constraints significantly reduces the overall complexity of the controller. The computation of terminal cost weight requires knowledge of the terminal constraints.
L. Grüne and J. Pannek [21]	Stability criteria for MPC without terminal conditions are proposed.	The MPC without terminal conditions further reduce the complexity of the controller design because no prior knowledge is needed. However, the stability is not guaranteed during the runtime and can only be verified after.
Liberzon [47]	The author discusses the topics of finite-horizon and infinite-horizon linear quadratic regulators. The finite and infinite linear quadratic regulator problems can be solved through the Riccati differential equation and the algebraic Riccati equation.	Although linear optimal control offers closed-form solutions, it falls short in addressing nonlinearity and exhibits a high model dependency.
Ci [48]	The author assesses the stability margin of the linear quadratic regulator when subjected to various types of perturbations.	The stability margin for NMPC has not been well studied yet.
Tedrake [49]	Topics about underactuated nonlinear dynamics and optimal controls are included in this book.	Linear quadratic regulators provide a local approximation of the optimal control solution for the nonlinear system. However, not all nonlinearities can be captured using LQR approximation.
Gilbert and Tan [50]	The authors introduce a method to obtain the maximal output admissible sets for control systems.	Computational effort for the proposed algorithm is reasonable for a linear system but is still heavy for a nonlinear system.

3.8. Stability Conditions for NMPC

Here is an overview of the prevailing stability proofs for both NMPC and robust NMPC. To set the foundation, we will begin by introducing the definitions of comparison functions.

Definition 3.8.1 ($\mathcal{K} - Function$**)** $\mathcal{K} := \{\alpha(\cdot) : \mathbb{R}_0^+ \to \mathbb{R}_0^+ |\ \alpha(\cdot)$ *is continuous* & *strictly increasing with* $\alpha(0) = 0\}$

Definition 3.8.2 ($\mathcal{K}_\infty - Function$**)** . $\mathcal{K}_\infty := \{\alpha(\cdot) : \mathbb{R}_0^+ \to \mathbb{R}_0^+ |\ \alpha(\cdot) \in \mathcal{K}$ & *with* $\alpha(\cdot)$ *is unbounded*$\}$

Definition 3.8.3 ($\mathcal{L} - Function$**)** $\mathcal{L} := \{\delta(\cdot) : \mathbb{R}_0^+ \to \mathbb{R}_0^+ |\ \delta(\cdot)$ *is continuous* & *strictly decreasing with* $\lim_{t\to\infty} \delta(t) = 0$

Definition 3.8.4 ($\mathcal{KL} - Function$**)** $\mathcal{KL} := \{\beta(\cdot) : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+ |\ \beta(\cdot)$ *is continuous,* $\beta(\cdot, t) \in \mathcal{K},\ \beta(r, \cdot) \in \mathcal{L}\}$

Consider a continuous autonomous nonlinear system represented by:

$$\dot{x} = F(x). \tag{3.10}$$

This system has a discrete counterpart, characterized by a sampling frequency h, as:

$$x(k+1) = f(x(k)), \tag{3.11}$$

where x lies with the state constraint set ($\mathbb{X}$), and it includes the origin. Assuming that an asymptotically stabilizing feedback control law for the sampled system (3.11) remains uniformly bounded over the sampling period $1/h$ when applied to the continuous model (3.10), we can conclude that the controller derived from the sampled system remains asymptotically stable in the continuous time system as well [21]. This assumption is crucial in ensuring that the stability properties observed in the sampled system also hold true to its continuous-time counterpart.

Utilizing comparison functions and the given sampled system (3.11), we can proceed to define ”asymptotic stability” and ”Lyapunov stability”.

Definition 3.8.5 (Asymptotically stable (AS) and Global asymptotically stable (GAS)) *Suppose* $\mathbb{X}$ *is positive invariant for the sampling system in* (3.11)*. The origin is asymptotically stable in* $\mathbb{X}$ *if there exists a* $\mathcal{KL}$ *function* $\beta(\cdot)$ *such that, for each* $x \in \mathbb{X}$

$$\Phi(k; x) \leq \beta(|x|, k)\ \forall k \geq 0. \tag{3.12}$$

If $\mathbb{X} = \mathbb{R}^n$*, the origin is GAS for* $x^+ = f(x)$

Definition 3.8.6 (Lyapunove function) *Suppose that* $\mathbb{X}$ *is positive invariant for the system in Eq.* (3.11)*. A function* $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ *is said to be a Lyapunov function in* $\mathbb{X}$ *for for the system if there exist functions* α_1*,* $\alpha_2 \in \mathcal{K}_\infty$ *and* $\alpha_3 \in \mathcal{K}$ *such that for any* $x \in \mathbb{X}$

$$V(x) \geq \alpha_1(|x|) \tag{3.13}$$

$$V(x) \leq \alpha_2(|x|) \tag{3.14}$$

$$V(F(x)) - V(x) \leq -\alpha_3(|x|) \tag{3.15}$$

Theorem 3.8.1 (Lyapunov stability theorem) *Suppose* $\mathbb{X} \in \mathbb{R}^n$ *is positive invariant for the system in Eq.* (3.11)*. If a Lyapunov function exists in* $\mathbb{X}$ *for such a system, then the origin is asymptotically stable in* $\mathbb{X}$ *for the system. If* $x = \mathbb{R}^n$*, the origin is globally asymptotically stable.*

3.8.1. Recursive Feasibility and Stability Conditions for NMPC with Terminal Conditions

For a typical receding horizon optimal control problem using stabilizing terminal conditions, the cost function for N prediction horizon ($V_N(\cdot)$) is defined below

$$x(k+1) = f(x(k), u(k)), \tag{3.16}$$

$$V_N(x,u) := \sum_{k=0}^{N-1} l(x(k), u(k)) + V_f(x(N)). \tag{3.17}$$

Equation (3.16) is the sampled system with sampling frequency h. The cost function is formed by stage cost ($l(\cdot)$) and a terminal cost ($V_f(\cdot)$). $x(k) \in \mathbb{X}$, $u(k) \in \mathbb{U}$ for all $k \geq 0$. $x(N)$ is contained in the terminal constraint ($\mathbb{X}_f$) where $\mathbb{X}_f \subseteq \mathbb{X}$. $u \in \mathcal{U}_N(x)$. The control constraint set $\mathcal{U}_N(x)$ is the set of control sequences

$$u := (u_0(x), u_x(x), \ldots, u_{N-1}(x)). \tag{3.18}$$

Let u^0 be the optimal control sequence and $V_N^0(x)$ be the optimal cost. Only the first control action from the optimal control solution $\kappa_N(x) = u_0^0(x)$ is applied to the system, and the optimal control problem is solved again in the next sampling step. Moreover, the domain of attraction (X_N) is defined as $X_N = \{x \in \mathbb{X} | u \neq \emptyset\}$. The detailed definition can be found in Definition 3.8.7. The existence of the solution to the above optimal control problem relies on the following two assumptions from [51].

Assumption 3.8.1 (Continuity of system and cost[51]) *The functions $f : \mathbb{Z} \to \mathbb{R}^n$, $l : \mathbb{Z} \to \mathbb{R}_{\geq 0}$ and $V_f : \mathbb{X}_f \to \mathbb{R}_{\geq 0}$ are continuous, $f(0,0) = 0$, $l(0,0) = 0$ and $V_f(0) = 0$.*

Assumption 3.8.2 (Properties of constraint sets [51]) *The set $\mathbb{Z}$ is closed. If there are control constraints, the set $\mathbb{U}(x)$ is compact and is uniformly bounded in $\mathbb{X}$ (i.e., $|f(x,u)| \leq C$, for $u \in U$, for all x). The set $\mathbb{X}_f \subseteq \mathbb{X}$ is compact (closed and bounded). Each set contains the origin. If there are no control constraints, the function $u \to V_N(x,u)$ is coercive, i.e., $V_N(x,u) \to \infty$ as $|u| \to \infty$ for all $x \in \mathbb{X}$*

A basic stability assumption relies on the specific properties of the terminal cost ($V_f(\cdot)$), terminal constraint ($\mathbb{X}_f$) and stage cost ($l(\cdot)$),

Assumption 3.8.3 (Stability assumption for NMPC with terminal conditions [51])

1. *For all $x \in \mathbb{X}_f$, $\mathbb{X}_f = \{x \in \mathbb{X} : V_f(x) \leq \alpha,\ \alpha > 0\}$, there exists a u, such that $x \in \mathbb{X}$, $u \in \mathbb{U}$, satisfying*

$$f(x,u) \in \mathbb{X}_f \tag{3.19}$$

$$V_f(f(x,u)) - V_f(x) \leq -l(x,u) \tag{3.20}$$

2. *There exist $\mathcal{K}_\infty$ functions $\alpha_1(\cdot)$ and $\alpha_f(\cdot)$ satisfying*

$$l(x,u) \geq \alpha_1(|x|),\ \forall x \in X_N,\ (x,u) \in \mathcal{Z} \tag{3.21}$$

$$V_f(x) \leq \alpha_f(|x|),\ \forall x \in \mathbb{X}_f \tag{3.22}$$

Under the Assumption 3.8.1, 3.8.2 and 3.8.3, the optimal cost ($V_N^0(x)$) satisfies Definition 3.8.6 and is a Lyapunov function. The detailed proof can be found in [51]. In short, the lower bound for $V_N^0(\cdot)$ is satisfied by the design of

$$l(x,u) = \frac{1}{2}(x^T Q x + u^T R u).$$

Thus, $V_N^0(x) \geq l(x, \kappa_N(x)) \geq \alpha_1(|x|)$, where $\alpha_1 \in \mathcal{K}_\infty$. Moreover, the upper bound for $V_N^0(\cdot)$ is bounded by

$$V_N^0(x) \leq V_{N-1}^0(x) \leq, \ldots, \leq V_0^0(x) \leq V_f(x) \leq \alpha_f(|x|),\ \forall x \in \mathbb{X}_f,$$

from (3.22). These inequalities are known as the monotonicity of the value function. The upper bound in set $\mathbb{X}_f$ implies a similar upper bound $\alpha_2(|x|) \in \mathcal{K}_\infty$ in the set X_N. Therefore, $\alpha_1(|x|) \leq V_N^0(x) \leq \alpha_2(|x|)$.

A weak controllability assumption is proposed for the terminal constraint that only contains the origin.

Assumption 3.8.4 (Weak controllability[51]) *There exists a $\mathcal{K}_\infty$ function $\alpha(\cdot)$ such that*

$$V_N^0(x) \leq \alpha(|x|) \; \forall x \in X_N \tag{3.23}$$

The descent property, $V_N^0(f(x, \kappa_N(x)) - V_N^0(x) \leq -l(x, \kappa_N(x)), \; \forall x \in X_N$, can be established from the monotonicity property and (3.20).

Theorem 3.8.2 (Asymptotic stability of the origin for NMPC with terminal conditions[51]) *Suppose Assumptions 3.8.2, 3.8.1, 3.8.3 and 3.8.4 are satisfied. Then*

1. *There exists $\mathcal{K}_\infty$ functions $\alpha_1(\cdot)$ and $\alpha_2(\cdot)$ such that*

$$\begin{aligned} \alpha_1(|x|) &\leq V_N^0(x) \leq \alpha_2(|x|) \\ V_N^0(f(x, \kappa_N(x))) - V_N^0(x) &\leq -\alpha_1(|x|) \end{aligned}$$

   for all $x \in X_N$.

2. *The origin is asymptotically stable in X_N for $x(k+1) = f(x(k), \kappa_N(x(k)))$.*

For the proof of Theorem 3.8.2, see [51].

Definition 3.8.7 (viability kernel and recursive feasibility [52]) *Let a constraint set $\mathbb{X}$ and an optimization horizon $N \in \mathbb{N}_\infty$ for the NMPC algorithm given in Equations 3.16 and 3.17.*

1. *A point $x \in \mathbb{X}$ is called feasible for $\mathbb{X}$ and N if $\mathcal{U}_N(x) \neq \emptyset$.*
2. *The domain of attraction (feasible set) for $\mathbb{X}$ and N is defined as*

$$\mathbb{X}_N := \{x \in \mathbb{X} | x \text{ is feasible for } \mathbb{X} \text{ and } N\}.$$

   The set $\mathbb{X}_\infty$ is called the viability kernel.

3. *A set $A \subseteq \mathbb{X}$ is called recursively feasible for optimization horizon $N \in \mathbb{N}$ if $A \subseteq \mathbb{X}_N$ and it is forward invariant for the NMPC feedback law $\kappa_N(x)$, i.e., if $f(x, \kappa_N(x)) \in A$ holds for all $x \in A$.*

From Definition 3.8.7, recursive feasibility ensures that the optimal control problem can always be solved if the initial optimal control problem is solvable and the viability kernel is the largest possible recursively feasible subset inside $\mathbb{X}$. Theorem 3.8.2 concludes the stability conditions for NMPC with terminal cost and constraint. The recursive feasibility from Definition 3.8.7 is also guaranteed through tailored terminal conditions. However, constructing the terminal cost as a Lyapunov function can be challenging, particularly for complex nonlinear systems and time-varying reference trajectories. As a result, in practical applications, it is often preferred to remove the terminal constraint to simplify the controller design. The computational burden is reduced by removing the terminal constraint, making the implementation more feasible in industrial settings.

3.8.2. Recursive Feasibility and Stability Conditions for NMPC Without Terminal Conditions

Removing the terminal constraint may come at a cost. The stability and feasibility of the control system could be lost, and the domain of attraction may be reduced. A study by Parisini and Zoppoli [53] explores the stability of NMPC with terminal cost but without terminal constraint, where the terminal cost function is defined as $V_f(x) = ax^T Px$, with $a > 0$ and P being a positive definite matrix. It is proven that for any stabilizable initial state, a triplet (a, P, and N) exists in which N is the prediction horizon, ensuring the stability of the controlled system. However, there is no indication of how to determine the appropriate triplet. The authors suggest trial-and-error procedures, which may not be practical for all systems.

To reestablish the stability and feasibility when the terminal constraint is removed, Limon et al. [20] proposed a weighted terminal scheme based on the work of Parisini and Zoppoli [53]. The main idea is to increase the terminal cost weight, which penalizes the system for not steering to the origin after N steps. The optical solution is forced to approach the terminal point more aggressively by introducing a larger penalty weight, thus potentially enlarging the domain of attraction. However, this comes at the expense of tracking ability, as the controller may prioritize reaching the terminal point over accurately tracking the reference trajectory. The study by Limon et al. [20] provides an approach to determine a suitable terminal cost weight that strikes an appropriate balance between stability and tracking performance and is summarized below. Starting with the assumption on stage cost ($l(\cdot)$):

Assumption 3.8.5 (lower bound for $l(x,u)$ [20]) *Let d be a positive constant such that*

$$l(x,u) > d,\ \forall x \notin \mathbb{X}_f,\ \forall u \in \mathbb{U}$$

.

Let $P_N(x,\mathbb{X})$ be the optimal control problem in (3.17) without terminal constraint ($\mathbb{X}_f$). With Assumptions 3.8.3 and 3.8.5, there exists a set $\Gamma_N \subseteq \mathbb{X}$, such that for any $x \in \Gamma_N$, the solutions from the optimal control problem $P_N(x,\mathbb{X})$ asymptotically stables the system as described by Theorem 3.8.3.

Theorem 3.8.3 (asymptotically stability [20]) *Consider $V_f(x)$ and $\mathbb{X}_f = \{x \in \mathbb{X} : V_f(x) \leq \alpha\}$ such that satisfy Assumption 3.8.3; let d be a constant such that Assumption 3.8.5 holds, then the optimal NMPC controller with $N \geq 1$ derived from $P_N(x,\mathbb{X})$ stabilizes asymptotically the system for any initial state in*

$$\Gamma_N = \{x \in \mathbb{R}^n : V_N^0(x) \leq l(x, K_N(x)) + (N-1)d + \alpha\} \tag{3.24}$$

From the above theorem, one can easily be shown that the set

$$\gamma_N = \{x \in \mathbb{R}^n : V_N^0(x) \leq Nd + \alpha\} \tag{3.25}$$

is also a domain of attraction of the system controlled by the proposed controller, and it is contained in Γ_N.

Now, adding the weight $\lambda \geq 1$ to the cost and define

$$\Gamma_N(\lambda) = \{x \in \mathbb{R}^n : V_N^0(x) \leq l(x, K_N(x)) + (N-1)d + \lambda\alpha\} \tag{3.26}$$

$$\gamma_N(\lambda) = \{x \in \mathbb{R}^n : V_N^0(x) \leq Nd + \lambda\alpha\}. \tag{3.27}$$

The enlargement of the domain of attraction using weighting factors is established in Theorem 3.8.4.

Theorem 3.8.4 (Enlarge the domain of attraction using Weighting factors [20]) *Consider $V_f(x)$ and $\mathbb{X}_f = \{x \in \mathbb{X} : V_f(x) \leq \alpha\}$ such that satisfies assumption 3.8.3. Consider the NMPC controller with $N \geq 1$ derived from $P_N(x,\mathbb{X})$ using a weighted termianl cost $V_f(\lambda, x)$, with $\lambda \geq 1$. Then for all $\lambda_0 \leq \lambda_1$, $\gamma_N(\lambda_0) \subseteq \gamma_N(\lambda_1)$*

The stability of the system using a weighted factor is stated as follows:

Theorem 3.8.5 (Stabilizing using weighted factor [20]) *Let $\tilde{\mathbb{X}}_f$ denote the interior of $\mathbb{X}_f$. For any $x_0 \in X_N(\tilde{\mathbb{X}}_f)$,*

$$X_N(\tilde{\mathbb{X}}_f) = \{x_0 \in \mathbb{X},\ x_N \in \tilde{\mathbb{X}}_f | u \neq \emptyset\},$$

there exists a finite constant λ such that $x_0 \in \gamma_N(\lambda)$, and hence, the system can be stabilized using the NMPC without terminal constraint.

For the proof of Theorem 3.8.3, 3.8.4 and 3.8.5, see [20]. Furthermore, the weighted factor can be calculated using

$$\lambda = \frac{N(D-d)}{(1-\rho)\alpha} \tag{3.28}$$

where $\rho \in (0, 1]$ is a constant such that $x_0 \in X_N(\mathbb{X}_{f,\rho})$, where $\mathbb{X}_{f,\rho} = \{x | V_f(x) \leq \rho\alpha\}$. D is defined as a constant such that for all $x \in \mathbb{X}$ and $u \in \mathbb{U}$, $l(x, u) \leq D$.

The NMPC scheme without terminal constraints shows promise. However, the stability results presented in Theorem 3.8.5 rely on the Assumption 3.8.3, which assumes that the region for the local Lyapunov terminal cost function, denoted as $V_f(x) \leq \alpha$ with $\alpha > 0$, is known. Determining such a region can be challenging and introduces additional design complexity. Moreover, the constant d from Assumption 3.8.5 also requires prior computations and must be found using Monte Carlo simulations before implementation.

The dependency on these assumptions and prior information may limit the practicality and applicability of the NMPC scheme without terminal constraints [20]. However, there are exceptions to this limitation. For instance, if the reference trajectory is well-defined and C^∞ continuous, a closed-form continuous control solution exists. In this scenario, when the initial condition aligns with the reference trajectory, the optimal cost V_N^0 is zero, and the optimal control solution is equivalent to the sequence of the closed-form control solution. Since $V_N^0 = 0$, this initial condition belongs to Γ_N from Eq. (3.25), guaranteeing stability.

However, for cases where the initial condition does not align with the reference trajectory, prior knowledge of the terminal set and constant d from Assumption 3.8.5 is required. To alleviate these prerequisites and reduce design complexity, we introduce the NMPC scheme without terminal conditions next. This approach is a ready-to-use method with minimal design complexity, and the stability results rely on constructing a relaxed descent condition of the optimal cost,

$$V_N^0(x) \geq \alpha l(x, \kappa_N(x)) + V_N^0(f(x, \kappa_N(x))), \ \forall x \in \mathbb{X} \tag{3.29}$$

where $\alpha \in (0, 1]$. Furthermore, the optimal cost and the stage cost should be bounded by $\mathcal{K}_\infty$ functions, $\alpha_1(\cdot)$, $\alpha_2(\cdot)$, and $\alpha_3(\cdot)$ such that the following inequalities hold.

$$\alpha_1(|x|) \leq V_N^0(x) \leq \alpha_2(|x|) \tag{3.30}$$

and

$$l(x, \kappa_N(x)) \geq \alpha_3(|x|) \tag{3.31}$$

The main assumptions for stabilising NMPC without terminal conditions are summarized from [21].

Assumption 3.8.6 (Bounds on the stage cost and infinite horizon cost [21])

1. *There are functions $\alpha_1(\cdot), \alpha_2(\cdot) \in \mathcal{K}_\infty$ such that $l^*(x) := min_{u \in \mathbb{U}} l(x, u)$ satisfies*

$$\alpha_1(|x|) \leq l^*(x) \leq \alpha_2(|x|) \tag{3.32}$$

2. *There exist a function $\alpha_3 \in \mathsf{K}_\infty$ such that the optimal infinite horizon cost,*

$$V_\infty(x) \leq \alpha_3(|x|), \tag{3.33}$$

holds for all $x \in \mathbb{X}$

Assumption 3.8.7 (Bounds on V_N^0 [21]) *There exists $\gamma_k > 0$ such that the inequality*

$$V_k^0(x) \leq \gamma_k l^*(x) \tag{3.34}$$

or

$$V_k^0(x) \leq B_k(l^*(x)), B_k \in \mathcal{K}_\infty \tag{3.35}$$

holds for all $k \geq 2$, all $x \in \mathbb{X}$ and l^* is defined in assumption 3.8.6.

Theorem 3.8.6 (Stability without terminal conditions [21]) *Consider the NMPC without terminal con-*

ditions satisfies Assumptions 3.8.6 and 3.8.7. Let the performance index α_N governed by

$$\alpha_N \geq 1 - \frac{(\gamma_N - 1)\Pi_{k=1}^{N}(\gamma_k - 1)}{\Pi_{k=2}^{N}\gamma_k - \Pi_{k=2}^{N}(\gamma_k - 1)}. \tag{3.36}$$

If $\alpha_N \in (0, 1]$, then the nominal NMPC closed-loop system with NMPC-feedback law $\kappa_N(x)$ is asymptotically stable in $\mathbb{X}$.

Under the assumptions stated in Assumption 3.8.6 and 3.8.7, the existence of the performance index, as proposed in Theorem 3.8.6, guarantees the stability of the closed-loop system. For the proof of Theorem 3.8.6, see [21]. Under the same assumptions, another interesting result regarding the prediction horizon, N, can also be concluded. The result suggests that the optimal NMPC closed-loop system, operating without terminal conditions, achieves asymptotic stability in $\mathbb{X}$ provided N is sufficiently large.

The recursive feasibility of NMPC without terminal conditions can be established under the optimality and stability properties of the solutions, along with additional assumptions on the viability kernel and a sufficient large prediction horizon. For the stabilizing condition, Assumption 3.8.8 is proposed to ensure Assumption 3.8.7 utilizing the class of $\mathcal{KL}_0$-functions.

Definition 3.8.8 ($\mathcal{KL}_0$-functions [21]) $\mathcal{KL}_0 := \{\beta(\cdot) : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+ | \beta(\cdot)$ *is continuous,* $\beta(\cdot, t) \in \mathcal{K}_\infty$ *or* $\beta(\cdot, t) \equiv 0,\ \beta(r, \cdot) \in \mathcal{L}\}$

Assumption 3.8.8 (Asymptotic controllability wrt. l [21]) *Consider the optimal control problem. We assume that the system is asymptotically controllable with respect to l with rate $\beta \in \mathcal{KL}_0$, i.e., for each $x \in \mathbb{X}$ and each $N \in \mathbb{N}$ there exist an admissible control sequence $u \in \mathcal{U}_N(x)$ satisfying*

$$l(x_u(k, x), u(k)) \leq \beta(l^*(x), k)$$

for all $k \in \{0, \ldots, N-1\}$.

If the $\mathcal{KL}_0$ function is linear to its first argument, $\beta(r, k) = c_k r$, with $c_k \geq 0$ and $c_k = 0$ for all $k \geq k_0$, then, the $\mathcal{K}_\infty$ function from Assumption 3.8.7 has the from

$$B_k(r) = C_k r \text{ with } C_k = \sum_{j=1}^{\min\{k_0, k-1\}} c_j.$$

Assumptions 3.8.7 or 3.8.8 holds true for all $x \in \mathbb{X}$ if the state constraint set $\mathbb{X}$ is viable, i.e., all the subsequent states are inside $\mathbb{X}$. Assumption 3.8.9 generalizes Assumption 3.8.8 for initial values inside the viability kernel, $\mathbb{X}_\infty$.

Assumption 3.8.9 (Asymptotic controllability [52]) *Consider the optimal control problem with a not necessarily viable state constraint. We assume that on the viability kernel $\mathbb{X}_\infty$ the system is asymptotically controllable with respect to l with rate $\beta \in \mathcal{KL}_0$, i.e., for each $x \in \mathbb{X}_\infty$ and each $N \in \mathbb{N}$ there exist an admissible control sequence $u \in \mathcal{U}_N(x)$ satisfying $x_u(k, x), u(k)) \in \mathbb{X}_\infty$ for all $k = 1, \ldots, N$ and*

$$l(x_u(k, x), u(k)) \leq \beta(l^*(x), k)$$

for all $k \in \{0, \ldots, N-1\}$.

Assumption 3.8.10 ([52]) *There exist α_1, α_2, α_3, $\alpha_4 \in \mathcal{K}_\infty$ and $N_0 \geq 2$ such that the inequalities*

$$\alpha_1(|x|) \leq V_N^0(x) \leq \alpha_2(|x|)$$

and

$$\alpha_3(|x|) \leq l^*(x) \leq \alpha_4(|x|)$$

holds for all $N \geq N_0$.

Assumption 3.8.10 helps apply the stability results of the NMPC scheme without terminal conditions. The last assumption ensures the existence of feedback solution around the origin under the given state constraints $\mathbb{X}$.

Assumption 3.8.11 ([52]) *There exists a ball $\mathcal{B}_\delta$ such that $\mathcal{B}_\delta \cap \mathbb{X} \subseteq \mathbb{X}_\infty$.*

With three Assumptions listed, the feasibility theorem for the NMPC without terminal conditions can finally be formulated.

Theorem 3.8.7 (Recursive feasibility for NMPC without terminal conditions [52]) . *Let Assumptions 3.8.9-3.8.11 hold, let $N \geq 2$ and assume that the performance index α_N from theorem 3.8.6 with β, linear to its first argument, from Assumption 3.8.9 satisfies $\alpha_N \in (0, 1]$. Then the set*

$$A = \{x \in \mathbb{X}_\infty | V_N^0(x) < N\alpha_3 \circ \alpha_2^{-1} \circ \alpha_3(\delta)\}$$

with α_2, $\alpha_3 \in \mathcal{K}_\infty$ from Assumption 3.8.10 is recursively feasible for the NMPC feedback $\kappa_N(x)$ from the NMPC scheme without terminal conditions. Furthermore, the NMPC closed loop is asymptotically stable on A.

In summary, the NMPC scheme with terminal conditions offers the advantage of guaranteed stability and recursive feasibility. However, the complexity of designing the terminal cost can be a drawback for practical implementation. On the other hand, the NMPC scheme without terminal conditions is simpler to implement, but ensuring recursive feasibility may require additional assumptions and computational efforts, particularly when dealing with non-viable state constraint sets.

3.9. Robust Nonlinear Model Predictive Control

The effectiveness of NMPC relies on plant accuracy, making it susceptible to disturbances that can significantly impact system performance. Two types of disturbances commonly considered in NMPC designs are stochastic and bounded disturbances. To address the impact of stochastic disturbances, stochastic NMPC is designed to optimize the expected value of NMPC's output. On the other hand, robust NMPC is developed to mitigate the effects of bounded disturbances. For the purpose of this study, because we consider the state constraints as an important controller design factor, the focus will be solely on bounded disturbances. As in stochastic cases, the disturbance is not necessarily assumed to be bounded and possibly results in constraint violations.

Achieving nominal robustness for state-constrained NMPC is a challenging task, as even a small disturbance can significantly alter the plant's behaviour. This limitation has been highlighted by Teel [54], who pointed out that a globally asymptotically stable state constraint NMPC has no robustness. With state constraints, arbitrary small perturbations can push states near the boundary to go out of bounds, thus destroying the recursive feasibility of a nominally recursively feasible set [52].

One approach to address the robustness of NMPC is rooted in game theory. Chen, Scherer, and Allgower [55] proposed a min-max NMPC formulation that first maximizes the cost function concerning worst-case disturbance realizations and then minimises the maximized cost function with respect to the control input. However, this formulation may lead to overly conservative solutions due to the open-loop nature of the approach. To tackle the conservativeness brought by the open-loop solution, a closed-loop min-max NMPC has been proposed in [56]. This closed-loop min-max NMPC accounts for the effect of uncertainty in predictions, thus mitigating conservativeness. Nevertheless, including the control law as a decision variable in the optimization problem increases the computational complexity, posing challenges in solving the optimization problem. The primary disadvantages of both open-loop and closed-loop min-max NMPC lie in the existence of feasible solutions and the high computational complexity they entail.

An alternative approach to achieve nominal robustness in state-constrained NMPC is proposed in [22]. This approach, known as open-loop nominal NMPC with tightened constraints, utilizes an upper bound on the disturbance to tighten the state and input constraints. The tightened constraints ensure that the predicted states remain within a region containing the desired equilibrium point. Using the nominal system offers the advantage of reducing computational costs during optimization. However, this method may lead to a conservative solution due to considering the worst effect of the disturbance when computing the tightened constraints.

Building upon the work of Marruedo, Alamo, and Camacho [22], Rubagotti et al. [24] introduced the use of ISMC to enhance the robustness of the NMPC algorithm. In the paper, ISMC generates an auxiliary control action that minimises the difference between the nominal predicted dynamics and the actual dynamics, thereby reducing the uncertainty. The reduced uncertainty is then used for constraint tightening in the robust NMPC algorithm, resulting in improved robustness. The final controller design combines both ISMC and NMPC control actions, as the former mitigates the effects of disturbance while the latter achieves desirable performance. Similarly, Xie et al. [25] proposed a Disturbance Observer (DOB) based NMPC algorithm. The DOB estimates the disturbance based on a known disturbance model and generates a control action to compensate for the matched disturbance. The effects of residue disturbance are addressed by the robust NMPC through constraint tightening. The ISMC and DOB-based robust NMPC algorithms ensure robustness by alleviating the effect of the disturbance. However, it is essential to note that both approaches heavily rely on the accuracy of the model. The DOB-based algorithm even requires additional prior knowledge of the disturbance model.

In conclusion, existing robust NMPC algorithms ensure stability, account for nonlinearity, have high robustness and deal with underactuation, but their performance highly relies on nominal model accuracy. Efforts to reduce the model dependency of robust NMPC remain important and are still under research. Please refer to Table 3.8 for a summarized list of the papers mentioned in this section.

Table 3.8: Summary of the robust NMPC

Source	Key Findings	Relation to this study
Teel [54]	Robustness of asymptotic stability in discrete receding horizon optimal control is discussed. Additionally, The author proposes a robust receding horizon optimal control scheme to address and enhance the system's robustness.	Robustness is not always guaranteed in the nominal NMPC solution. Additional state constraints or assumptions must be incorporated into the NMPC design to ensure robustness.
Grüne and Pannek [52]	Recursive feasibility and robustness of the closed-loop system subjected to additive disturbance and measurement errors are investigated.	The listed definitions of recursive feasibility and robustness can serve as valuable tools to establish and demonstrate our controller's recursive feasibility and robustness.
Chen, Scherer, and Allgower [55]	The proposed robust NMPC combines NMPC with H_∞ control and includes a specific terminal penalty.	Min-max NMPC considers all possible realizations of disturbances and requires heavy computing.
Magni et al. [56]	A closed-loop min-max NMPC is proposed. The future control policies are involved in the closed-loop strategy, and two different horizons are used. A longer horizon is used to optimize system performance, and a shorter horizon optimizes control policies.	min-max NMPC requires heavy computing. Adding a closed-loop policy would further increase runtime
Marruedo, Alamo, and Camacho [22]	A robust NMPC for bounded additive disturbance is proposed. The state constraints and terminal region are reformulated to get recursive feasibility.	This method utilizes a nominal plant to make predictions, requires less computational power and establishes a foundation for robust NMPC using tightened constraints. However, as the upper bound of disturbances is directly used, the tightened constraints could result in a null space.
Rubagotti et al. [24]	Extending the robust NMPC scheme from [22], the authors use integral Sliding Mode to reduce the impact of bounded uncertainty in state and input variables. The concept of input-to-state practical stability in continuous time is introduced.	The chattering phenomenon is not addressed, and only the input aligned disturbances can be compensated.
Xie et al. [25]	A disturbance rejection NMPC framework is proposed. This framework is particularly designed for the input-affine system.	This method can only compensate for input-aligned disturbances.

3.9.1. Recursive Feasibility and Robust Stability Conditions for Robust NMPC

To ensure robustness in state-constrained NMPC, two approaches are summarized from papers [25] and [52]. Both approaches utilize time-dependent state constraint sets. The following assumptions and definitions are presented to demonstrate the results from [25].

Consider a nonlinear input affine system

$$\dot{x}(t) = \mathcal{F}(x(t), u(t)) + B_w(x)w(t), \tag{3.37}$$

where the input-affine component $\mathcal{F}(x(t), u(t)) = f(x) + B(x)u(t)$, and $x \in \mathbb{X}$, $u \in \mathbb{U}$.

Assumption 3.9.1 (Bounded disturbance) *The disturbance and its derivative are assumed to be bounded by*

$$w(t) \in \mathbb{W} \triangleq \{w(t) | \|w(t)\| \leq \eta_1, \|\dot{w}(t)\| \leq \eta_2\} \tag{3.38}$$

where $\eta_1,\ \eta_2 > 0$ are known constants. The disturbance input matrix $B_w(x)$ is bounded by $\|B_w(x)\| \leq \bar{B}_w$ for all $x \in \mathbb{X}$. We denote the solution to system 3.37 as $\phi(t, x_0, t_0, w)$ where t is the sampling instance, x_0 is the initial state, t_0 is the initial time and w is the disturbance sequence.

Definition 3.9.1 (Robust Positive Invariant set) *A set $\Phi \subseteq \mathbb{R}^n$ is called a robust positively invariant for system 3.37 if, for all $x_0 \in \Phi$, it holds that $\phi(t, x_0, t_0, w) \in \Phi$, for all $w(\tau) \in \mathbb{W}$, where $\tau \in [t_0, t]$.*

Definition 3.9.2 (Regional ISpS in Φ) *Given a closed set $\Phi \in \mathbb{R}^n$, including the origin as an interior point, system 3.37 is said to be ISpS (Input-to-State practical Stable) in Φ with respect to w if Φ is robust positively invariant for system 3.37 and if there exist a $\mathcal{KL}$ function $\beta(\cdot)$, a $\mathcal{K}$ function $\gamma(\cdot)$ and a constant $c \geq 0$ such that*

$$\|\phi(t, x_0, t_0, w)\| \leq \beta(\|x_0\|, t - t_0) + \gamma(\|w\|) + c \tag{3.39}$$

for all $x_0 \in \Phi$ and $t \geq t_0$. When $c = 0$, system 3.37 is said to be ISS (Input-to-State Stable) in Φ with respect to w. For discrete sampling, definitions for ISS and ISpS are defined similarly to the continuous case.

Definition 3.9.3 (ISpS Lyapunov function in Φ) *. A function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is called an ISpS-Lyapunov function in Θ_w for system 3.37 with respect to w if:*

1. *Φ is a closed robust positively invariant set including the origin as an interior point*
2. *There exist $\mathcal{K}_\infty$ functions $\alpha_1(\cdot)$, $\alpha_2(\cdot)$ and $\alpha_3(\cdot)$, a $\mathcal{K}$ function $\gamma(\cdot)$, constants $c_1,\ c_2 \geq 0$ and a compact set $\Omega \subseteq \Phi,\ \{0\} \subset \Omega$ such that, for any $x(t) \in \Phi$ and almost all t*

$$\begin{aligned} V(x(t)) &\geq \alpha_1(\|x(t)\|) && (3.40)\\ V(x(t)) &\leq \alpha_2(\|x(t)\|) + c_1 && (3.41)\\ \dot{V}(x(t)) &\leq -\alpha_3(\|x(t)\|) + \gamma(\|w\|) + c_2 && (3.42) \end{aligned}$$

   For any point t_d of $V(x(t_d))$, if the inequality for $\dot{V}(x(t_d))$ does not hold, it is replaced by

$$V(x(t_d^+)) \leq V(x(t_d^-)) \tag{3.43}$$

   where t_d^- and t_d^+ are the left and right limits of time instant t_d.

3. *There exist suitable $\mathcal{K}_\infty$ functions $\alpha_c(\cdot)$ and $\gamma_c(\cdot)$ with $(Id - \gamma_c)(\cdot)$ being a $\mathcal{K}_\infty$ function and a compact set*

$$\Theta_w \triangleq \{x(t) | V(x(t)) \leq \bar{c}, \forall t \geq t_0\} \tag{3.44}$$

   where $Id(\cdot)$ is the identity function, $\bar{c} \triangleq b(\gamma(\eta_1) + c_3)$, $b(\cdot) \triangleq \alpha_4^{-1} \circ (Id - \gamma_c)^{-1}(\cdot)$, $\alpha_4(\cdot) \triangleq \underline{\alpha}_3 \circ \bar{\alpha}_2^{-1}(\cdot)$,$\underline{\alpha}_3 \triangleq min\{\alpha_3(s/2), \alpha_c(s/2)\}$ and $\bar{\alpha}_2 \triangleq \alpha_2(\cdot) + Id(\cdot)$, $c_3 = c_2 + \alpha_c(c_1)$.

A function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$is called an ISS-Lyapunov function in $\in \Phi$ if it is an ISpS Lyapunov function in Φ with $c_1 = c_2 = 0$.

Other papers have also explored the same definitions, as observed in Rubagotti et al. [24] and Raimondo et al. [57], where they have applied these concepts to nonlinear control affine systems. It is important

to mention that while Raimondo et al. [57] concentrates on the discrete-time case, Rubagotti et al. [24] studies continuous-time systems. Consequently, the equivalent discrete-time ISpS related definitions can be found in [57]. A sufficient condition to establish the regional ISpS is the existence of an ISpS Lyapunov function.

Theorem 3.9.1 *If system (3.37) admits an ISpS Lyapunov function associated with set Θ_w and Φ, the system is input-to-state practical stable in Φ and $lim_{t\to\infty}|\phi(t, x_0, t_0, w|_{\Theta_w} = 0$.*

The robust NMPC framework proposed by Xie et al. [25] adopts the NMPC with terminal conditions approach. To reduce the impact of disturbances, an auxiliary control is designed based on a disturbance observer. The remaining disturbances are handled by time-dependent state constraint sets. The weight matrices in the stage cost and terminal cost are carefully selected to ensure recursive feasibility. The proposed cost function also satisfies the second criterion of ISpS Lyapunov function (Definition 3.9.3).

However, for the first and third criteria in Definition 3.9.3, which pertain to the existence of a robust positively invariant set Φ and a compact set $\Theta_w \in \Phi$, these conditions were assumed to be satisfied due to computational difficulties.

Next, the second approach from [52] will be presented. The state constraint sets are denoted by $\mathbb{X}^k,\ k = N,\quad N-1,\quad \ldots,\quad 0$ with $\mathbb{X} = \mathbb{X}^N$. The admissible control sequence is modified as follows.

Definition 3.9.4 *For $K \in \mathbb{X}$ and $N \in \mathbb{N}$ with $K \leq N$ and an initial value $x_0 \in \mathbb{X}^K$ we call a control sequence $u \in \mathbb{U}^k$ and the corresponding trajectory $x_u(k, x_0)$ admissible for x_0 and K, if*

$$u(k) \in \mathbb{U}(x_u(k, x_0)) \text{ and } x_u(k+1, x_0) \in \mathbb{X}^{K-(k+1)}$$

holds for all $k = 0, \ldots, K-1$. The respect set of admissible control sequences for x_0 is denoted by $\mathbb{U}^K(x_0)$.

Recall that the NMPC without terminal conditions has the form

$$x(k+1) = f(x(k), u(k)),$$

$$\text{minimise } V_N(x, u) := \sum_{k=0}^{N-1} l(x(k), u(k)).$$

The perturbed closed-loop model is

$$\tilde{x}^+ = f(\tilde{x}, \kappa_N(\tilde{x} + e)) + d$$

where d is an additive perturbation and e is the measurement error. The robust stability property is defined in Definition 3.9.5.

Definition 3.9.5 (Practical stability with respect to perturbations [52]) *Given a set $A \in \mathbb{X}$ such that the optimal control problem defining κ_N is feasible for all $x_0 \in A$, we say that origin is semiglobally practically asymptotically stable on A with respect to the perturbations d and e if there exists $\beta \in \mathcal{KL}$ such that the following property holds:*
for each $\delta > 0$ and $\triangle > \delta$ there exists $\bar{d},\ \bar{e} > 0$, such that each solutions $x^{\tilde{}}_{\kappa_N}(\cdot, x_0) \in S_{(\bar{d},\bar{e})}$, where

$$S_{(\bar{d},\bar{e})}(x_0) := \{\tilde{x}_{\kappa_N}(\cdot, x_0) | \|d(k)\| \leq \bar{d}, \|e(k)\| \leq \bar{e} \text{ for all } k \in \mathbb{N}_0\},$$

with $x_0 \in A$ and $|x_0| \leq \triangle$ satisfies $\tilde{x}_{\kappa_N}(k, x_0) \in A$ and

$$|\tilde{x}_{\kappa_N}(k, x_0)| \leq max\{\beta(|x_0|, k), \delta\}$$

for all $k \in \mathbb{N}_0$, provided the initial measurement error $e(0)$ satisfies $x_0 + e(0) \in A$.

The next definition implies that the feasible set $\mathbb{X}_N$ is recursively feasible for the closed loop NMPC.

Definition 3.9.6 (Robust-optimal feasibility [52]) *For given optimization horizon $N \in \mathbb{N}$, the NMPC without terminal condition is said to be robust-optimal feasible, if for each closed ball $\bar{\mathcal{B}}_\rho$ around the origin there exists $\eta > 0$ such that the following holds:*
For each $x \in \bar{\mathcal{B}}_\rho \cap \mathbb{X}_N$, each $z \in \bar{\mathcal{B}}_\eta(f(x, \kappa_N(x)))$ and the optimal control u^* with $x_0 = x$ we have

$$x_{u^*(+1)}(k, z) \in \mathbb{X}_{N-k} \text{ for all } k \in \{0, \ldots, N-1\}.$$

A suitable controllability condition under the disturbance is proposed. This time, the recursive feasibility does not rely on the viability kernel but on the finite feasible set $\mathbb{X}_{N-k}$.

Assumption 3.9.2 ([52]) *Consider the optimal control problem with a not necessarily viable state constraint set $\mathbb{X}$. We assume that on the feasible sets $\mathbb{X}_{N-k}$, $k = 0, \ldots, N-1$ the system is asymptotically controllable with respect to l with rate $\beta \in \mathcal{KL}_0$, i.e., for each $x \in \mathbb{X}_{N-k}$ there exists an admissible control sequence $u \in \mathbb{U}^{N-k}(X)$ satisfying*

$$l(x_u(k, x), u(k)) \leq \beta(l^*(x), k)$$

for all $k \in \{0, \ldots, N-k-1\}$.

The practical stability for the perturbed closed-loop system can be established using the definition of robust-optimal feasibility with additional assumptions on the system's dynamics and Assumption 3.9.2.

The designs of robust NMPC indeed involve theoretical developments. However, it becomes challenging in practical applications to fully verify all the assumptions proposed in these algorithms, especially when dealing with complex systems. For example, in the case of Xie et al. [25], specifying the exact domain of attraction and determining the recursive feasible sets can be difficult. Similarly, in the context of [52], constructing the constraint sets $\mathcal{X}^k$ to satisfy the robust-optimal feasibility may not be straightforward. In practice, verifying robustness through theoretical proof alone can be a daunting task. Due to the complexity and uncertainties present in many real-world systems, robustness tests through simulations and experiments are often preferred.

3.10. Literature Synthesis

The design aspects previously mentioned, such as nonlinearity, stability, underactuation, robustness, state and input constraints, and model dependency, present multifaceted challenges in control UMS. While robust NMPC displays potential in addressing many of these issues, its major drawback lies in its substantial model reliance. In contrast, nonlinear incremental control stands out for its minimal model dependency, with the INDI technique showcasing sophisticated stability studies grounded in the Lyapunov stability theorem. The possibility of merging these two methods is of significant interest. Such a hybridization could potentially leverage the strengths of both approaches.

Recently, a loosely coupled approach combining NMPC with INDI was introduced in [26]. Within this framework, the inner loop control using INDI for part of the states has shown a marked enhancement in performance over standalone NMPC. Nevertheless, the integrated scheme operates based on a nominal NMPC model, lacking robustness considerations. Furthermore, no measurements have been utilized to compensate for other states. Hence, the coupled nonlinearities and robustness of these undressed states are not fully resolved. While serving as a control mechanism, INDI does not leverage or disclose disturbances information to NMPC. Overall, this loosely coupled approach does not guarantee the robustness of constraint satisfaction and is sensitive to disturbances that are not compensated by the INDI. To address limitations from [26] and to enhance robustness, a potential alternative is to use robust NMPC and incorporate information from INDI in a feedback loop. This would allow for a more comprehensive handling of disturbances.

4

Preliminary Work

This thesis is centred around the creation of a nonlinear incremental optimal control for input-affine Underactuated Mechanical Systems (UMS). Primarily, the designed control strategy will be implemented on a quadrotor to assess its performance during trajectory tasks. Additionally, we intend to broaden its application by testing it on other UMS systems, such as the aeroelastic systems.

Optimal control inherently demands the execution of computationally intensive routines. To facilitate this, we rely on two prominent simulation platforms: MATLAB [58] and the MATLAB embedded ACADOS toolbox [59]. MATLAB is a high-level programming language and provides several ready-to-use NMPC optimizers. The NMPC solver in MATLAB allows customization of cost functions and state constraints, making it ideal for implementing tightened constraints. On the other hand, the ACADOS toolbox offers faster optimal control solvers and a wider range of numerical analysis options. Notably, the default optimal control solvers in both platforms employ the sequential quadratic programming method. As for the system dynamics' numerical propagation, we've selected the fifth-order Runge-Kutta method.

Our assessment criteria for the controller's stability and robustness encompass both theoretical analysis and comprehensive simulations. For the preliminary test, the proposed controller will be deployed for quadrotor trajectory tracking. To this end, a spectrum of reference trajectories will be devised, and a set of distinct disturbances will be introduced. Simulations may not capture the entire dynamics and the full spectrum of potential disturbances. Therefore, future work will involve experimental tests.

4.1. Quadrotor Dynamics

To describe the quadrotor dynamics, two right-handed coordinate frames are utilized. The body frame $\mathcal{F}_B$ with axis x_B, y_B and z_B, has its x-axis points forward and z-axis aligns with the collective thrust direction. The inertial frame $\mathcal{F}_I$, with axis x_I, y_I and z_I, has its x-axis points towards North and z-axis opposites to the gravity. With these two frames, the quadrotor translational dynamics are defined as

$$\dot{\zeta} = v \tag{4.1}$$

$$\dot{v} = \frac{T z_B(q) + f_{\text{ext}}}{m} + g, \tag{4.2}$$

where ζ and v are position and velocity in the inertial frame, T is the magnitude of the collective thrust and $q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T \in S^3$ is the unit quaternion. The gravitational acceleration and quadrotor mass are denoted by g and m. The z-axis in the body frame z_B can be derived from the quaternion,

$$z_B = \begin{bmatrix} 2(q_1 q_3 + q_0 q_2) \\ 2(q_2 q_3 - q_0 q_1) \\ 1 - 2(q_1^2 + q_2^2) \end{bmatrix}.$$

The external force vector, f_{ext}, includes aerodynamic drags, possible wind gusts and other force disturbances. The rotational dynamics are given by

$$\dot{q} = \frac{1}{2} q \otimes \begin{bmatrix} 0 \\ \Omega_B \end{bmatrix}, \tag{4.3}$$

$$I_v \dot{\Omega}_B = -\Omega_B \times I_v \Omega_B + \tau + \tau_{ext}, \tag{4.4}$$

where $\otimes$ represents the quaternion multiplication and $\Omega_B = \begin{bmatrix} \Omega_x & \Omega_y & \Omega_z \end{bmatrix}^T$ is the body frame angular velocity. In Eq. (4.4), I_v is the quadrotor inertia matrix and $\tau = \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^T$ is the total torques generated from four rotors.

Lastly, the collective thrust and total torques can be represented by a function of rotor speed $\omega = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 & \omega_4 \end{bmatrix}^T$,

$$\begin{bmatrix} T \\ \tau \end{bmatrix} = G_1 u + G_2 \dot{\omega} + G_3(\Omega_B)\omega, \tag{4.5}$$

$$u = c_t \omega^{\circ 2}, \tag{4.6}$$

where u represents the thrust from each rotor, c_t is the thrust coefficient and $\circ$ is the Hadamard power. The last two terms from Eq. (4.5), $G_2\dot{\omega}$ and $G_3(\Omega_B)\omega$, account for the torque generated from rotor angular acceleration and gyroscopic effects [26]. G_1, G_2 and G_3 are given by

$$G_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ l\sin\beta & -l\sin\beta & -l\sin\beta & l\sin\beta \\ -l\cos\beta & -l\cos\beta & l\cos\beta & l\cos\beta \\ c_q/c_t & -c_q/c_t & c_q/c_t & -c_q/c_t \end{bmatrix} \tag{4.7}$$

$$G_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ I_p & -I_p & I_p & -I_p \end{bmatrix} \tag{4.8}$$

$$G_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ I_p\Omega_y & -I_p\Omega_y & I_p\Omega_y & -I_p\Omega_y \\ -I_p\Omega_x & I_p\Omega_x & -I_p\Omega_x & I_p\Omega_x \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.9}$$

where l is the length of quadrotor arm (from the centre to the tip), β is the angle between x_B and the quadrotor arm measured counterclockwise from the top view. c_q is the torque coefficient. I_p is the inertia of a single rotor around z_B and is estimated from

$$I_p = N_b \frac{1}{3} m_b l_b^2$$

where N_b is the number of blades on one rotor, m_b is the blade mass and l_b is the blade length. All parameters are summarized in Table 4.1 taken from [26].

Table 4.1: Quadrotor configurations [26].

Parameter	Value
m [kg]	0.75
l [m]	0.14
β [deg]	56
I_v [gm^2]	diag(2.5, 2.1, 4.3)
c_q [Nm^2]	$2.37e{-}8$
c_t [N^2]	$1.51e{-}6$
I_p [gm^2]	0.0107
$k_{d,x}, k_{d,y}, k_{d,z}$ [kg/m]	(0.26 0.28, 0.42)
k_h [kg/m]	0.01

4.2. State-of-the-art (SOTA) Approach

The State-Of-The-Art (SOTA) approach employed in [26] serves as a comparative basis for our controller, as the SOTA couples NMPC with INDI and has demonstrated its effectiveness in both simulations and experiments. The control diagram is shown in Fig. 4.1. This control strategy operates on a nominal NMPC model with nominal constraints. The external torque disturbances are reduced by an inner loop INDI attitude controller. Nevertheless, force disturbances are not explicitly handled, and the desired optimal thrust from the nominal NMPC model, denoted as T_{NMPC}, is directly used as the command thrust. In ideal cases, the INDI attitude controller filters out all external torque disturbances. However, due to actuator dynamics and control allocations, discrepancies between commanded and actual rotor-generated thrust and torques can exist, potentially leading to state constraint violations. Overall, the SOTA approach lacks robust satisfaction of state constraints and does not guarantee robustness in position control.

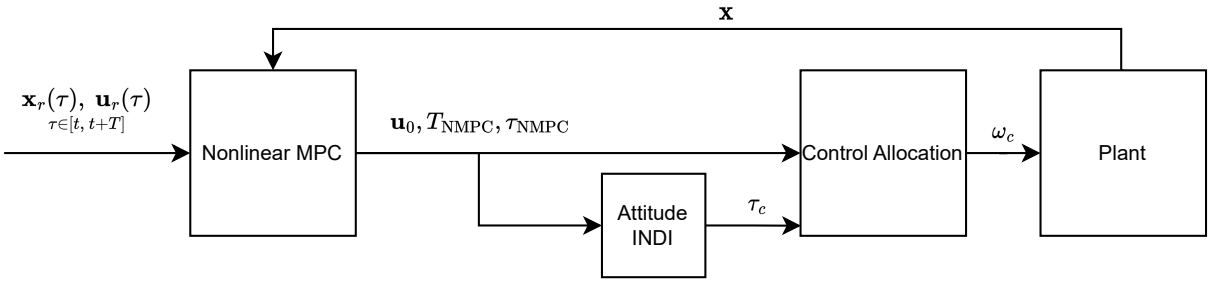


Figure 4.1: The control diagram of SOTA [26]. The attitude INDI follows Eq. (4.16), and the control allocation is solved by Eq. (4.23).

4.3. Methodology

This section proposes a robust NMPC scheme with tightened constraints relaxed by uncertainty reduction using the inner-loop INDI controllers. Furthermore, a tilt-prioritized INDI attitude controller is specifically tailored for the quadrotor model to reject the external torque and external force residues, mainly the thrust direction mismatch. The diagram of the proposed control scheme is illustrated in Fig. 4.2, where the constraint-tightening block establishes the connection between NMPC and inner loop INDI controllers.

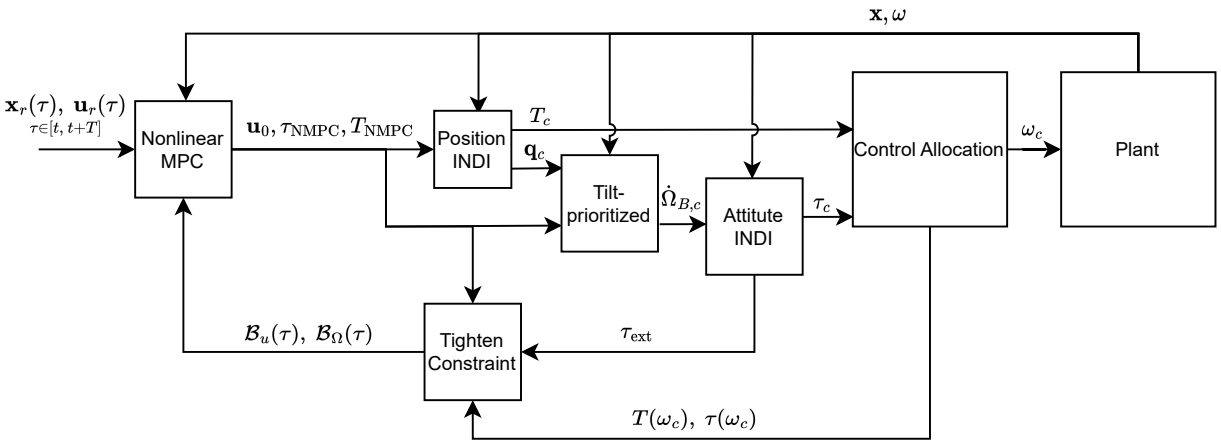


Figure 4.2: The control diagram of the proposed robust NMPC with inner loop INDI controllers tailored for quadrotor control.

4.3.1. Inner-Loop Position and Attitude Control

This section introduces INDI-based position control and tilt-prioritized INDI attitude control. The former addresses force disturbances by generating a collective thrust command, while the latter focuses on tracking the commanded thrust vector and yaw angle.

The INDI controller functions as a disturbance observer in the proposed controller framework. The instantaneous external disturbances can be estimated using the inverse of the dynamics and low-pass filtered flight states [12]:

$$T_{\text{ext}} = m(\dot{v}_f - g) - T_f z_B(q_f), \tag{4.10}$$

$$\tau_{\text{ext}} = I_v \dot{\Omega}_{B,f} - \tau_f + \Omega_{B,f} \times I_v \Omega_{B,f}, \tag{4.11}$$

where T_f and τ_f are the collective thrust and total torques derived from Eq. (4.12) using filtered rotor speed ω_f:

$$\begin{bmatrix} T_f \\ \tau_f \end{bmatrix} = G_1 \omega_f^{\circ 2} + G_2 \triangle t^{-1} (\omega_{f,k} - \omega_{f,k-1}). \tag{4.12}$$

The filtered linear and angular accelerations, $\dot{v}_f$ and $\dot{\Omega}_{B,f}$, come from the accelerometer and IMU. These external disturbance expressions hold true under the assumption that external disturbances are slow-changing relative to the low-pass filtered dynamics. The rotor angular acceleration is approximated using the finite difference method where $\triangle t$ is the INDI sampling time and subscript $k-1$ is the last sampled variable. With T_{NMPC} and τ_{NMPC} provided by the scheme from the next section, the desired linear and angular accelerations, $\dot{v}_d$ and $\dot{\Omega}_{B,d}$ are given by

$$\dot{v}_d = \frac{T_{\text{NMPC}} z_B(q_f)}{m} + g, \tag{4.13}$$

$$\dot{\Omega}_{B,d} = I_v^{-1}(\tau_{\text{NMPC}} - \Omega_{B,f} \times (I_v \Omega_{B,f})). \tag{4.14}$$

Then, by substituting the estimated disturbances Eq. (4.10) and Eq. (4.11) back into the dynamic equations Eq. (4.2) and Eq. (4.4), and replacing $\dot{v}$ by $\dot{v}_d$ from Eq. (4.13), we obtain the following expressions for the command thrust and torques:

$$T_c z_B(q_c) = m(\dot{v}_d - \dot{v}_f) + T_f z_B(q_f), \tag{4.15}$$

$$\tau_c = \tau_f + I_v(\dot{\Omega}_{B,c} - \dot{\Omega}_f), \tag{4.16}$$

where $\dot{\Omega}_{B,c}$ is the command angular acceleration and will be given in the latter part of this section. The thrust command T_c is then obtained using

$$T_c = \|T_c z_B(q_c)\|_2. \tag{4.17}$$

In Equation 4.15, the commanded thrust vector is coupled with the commanded quaternion q_c, which is different from the reference quaternion. To ensure the direction of the collective thrust, the quadrotor is required to rotate from its current attitude q_f to the commanded attitude q_c. Since the yaw angle has a lesser effect on the stability of the quadrotor, a tilt-prioritized attitude control approach [33] is utilized to achieve this rotation. This approach consists of two steps, and the first step is to align the z-axis in the body frame with the commanded thrust vector. The second step involves performing a yaw rotation to achieve the desired yaw angle.

The attitude error between q_c and q_f is defined as

$$q_e = q_c \times q_f^{-1}, \tag{4.18}$$

where the components of the quaternion error are $q_e = \begin{bmatrix} q_{e,0} & q_{e,1} & q_{e,2} & q_{e,3} \end{bmatrix}^T$. To achieve the first rotation, a reduced quaternion error is introduced

$$q_{e,red} = \frac{1}{\sqrt{q_{e,0}^2 + q_{e,3}^2}} \begin{bmatrix} q_{e,0}^2 + q_{e,3}^2 \\ q_{e,0}q_{e,1} - q_{e,2}q_{e,3} \\ q_{e,0}q_{e,2} + q_{e,1}q_{e,3} \\ 0 \end{bmatrix}, \tag{4.19}$$

where $q_{e,red}$ is defined as the shortest rotation to align the current thrust direction to the desired one. When $q_{e,0}^2 + q_{e,3}^2 = 0$, the reduced quaternion error is not well defined because there are infinite rotational options to rotate z-axis $180°$. Under this circumstance, an arbitrary rotational direction is picked. The yaw error for the second rotation is defined as

$$q_{e,yaw} = \frac{1}{\sqrt{q_{e,0}^2 + q_{e,3}^2}} \begin{bmatrix} q_{e,0} \\ 0 \\ 0 \\ q_{e,3} \end{bmatrix} \tag{4.20}$$

which is the required rotation to align the current x and y-axis to the desired coordinate frame. The $q_{e,yaw}$ is further relaxed to achieve improved tilt tracking. This modification is achieved through a relaxation procedure introduced in [31], which involves the use of a tunable parameter $\rho \in (0, 1]$. The expression of $q_{e,yaw}$ by vector rotation using the rotational axis $n_y = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ and rotational angle α_y is expressed in Eq. (4.21).

$$q_{e,yaw} = \frac{1}{\sqrt{q_{e,0}^2 + q_{e,3}^2}} \begin{bmatrix} q_{e,0} \\ 0 \\ 0 \\ q_{e,3} \end{bmatrix} = \begin{bmatrix} \cos(\rho\alpha_y/2) \\ 0 \\ 0 \\ \sin(\rho\alpha_y/2) \end{bmatrix}, \text{ for } \rho = 1. \tag{4.21}$$

By making $\rho \leq 1$, the yaw error is only corrected by ρ-fraction. The tilt-prioritized control law for the commanded angular acceleration is adopted from [32] but with extra tuning parameter $\rho \in (0, 1]$ to relax the yaw rotation,

$$\dot{\Omega}_{B,c} = 2k_r \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} q_{e,red} + 2k_y n_y sin(\frac{\rho\alpha_y}{2}) + K_\Omega \Omega_{B,e} + \dot{\Omega}_{B,d} \tag{4.22}$$

where k_r, k_y are positive constants, K_Ω is a diagonal positive definite matrix, and $\Omega_{B,e}$ is the angular velocity error expressed in the body frame. The desired angular acceleration $\dot{\Omega}_{B,d}$ is from Eq. (4.14). When $\rho = 1$, the controller above is equivalent to the one proposed in [33] and the one reviewed in [32]. The theoretical stability was also given in [33]. This tilt-prioritized control law gives an almost globally asymptotically convergence in the reduced attitude error. The yaw error almost globally converges to zero. In addition, this attitude control law has stable equilibriums. However, experimental results have shown that setting $\rho = 1$ does not lead to favourable tracking performance [31]. Instead, by adjusting $\rho \leq 1$, the controller can achieve better attitude tracking.

From Eq. (4.12), Eq. (4.16), and Eq. (4.17), the rotor speed control is obtained by solving a quadratic cost function,

$$\begin{aligned} \omega_c = \underset{\omega_{c,k}}{\text{argmin}} \; & \| \begin{bmatrix} T_c \\ \tau_c \end{bmatrix} - G_1 \omega_{c,k} - G_2 \triangle t^{-1} (\omega_{c,k} - \omega_{c,k-1}) \|_W \\ \text{s.t. } & \omega_{c,k} \in \mathbb{W}, \end{aligned} \tag{4.23}$$

where $\mathbb{W}$ is the rotor speed constraint and W is a positive-definite diagonal weight matrix. Each diagonal entry denotes the weight for thrust, pitch, roll and yaw channels. The weight for the yaw channel is set to be smaller than the weights for the other channels. This strategy is employed to mitigate control saturation by sacrificing yaw performance, given that the yaw angle has a lesser impact on the quadrotor's stability.

4.3.2. Constraint-Tightened Robust NMPC Scheme

The constraint-tightened robust NMPC ensures that future states and inputs remain within the nominal constraints even when subjected to bounded disturbances, thereby ensuring robustness. Concurrently, the reduced uncertainty relaxes the tightened constraint and alleviates the conservatives associated with this robust NMPC approach.

The state vector for a quadrotor is denoted as $x = \begin{bmatrix} \zeta & v & q & \Omega_B \end{bmatrix}^T$. To perform numerical optimization, the states and inputs are sampled in N equal time intervals spanning the time horizon T. The constrained optimization problem at any time step k, with state $x(k) = x_0$, is formulated in Eq. (4.24). The initial time index k is omitted for simplicity.

$$\begin{aligned} \mathbf{u}_{\text{opt}}(x) = \underset{u}{\text{argmin}} \sum_{i=0}^{N-1} & (||\bar{x}_i - x_{i,r}||_{Q_x} + ||u_i - u_{i,r}||_{Q_u}) \\ & + \|\bar{x}_N - x_{N,r}\|_{Q_N} \\ \text{subject to } & \bar{x}(i) = \dot{f}(\bar{x}_i, u_i) \qquad \bar{x}(0) = x_0 \\ & \Omega_{B,i} \in \{\Omega \in \mathbb{R}^3 | -\Omega_{\text{max}} \leq \Omega \leq \Omega_{\text{max}}\} \sim \mathcal{B}_\Omega(i) \\ & u_i \in \{u \in \mathbb{R}^4 | u_{\text{min}} \leq u \leq u_{\text{max}}\} \sim \mathcal{B}_u(i) \end{aligned} \tag{4.24}$$

In the above optimal control problem, $x_{(\cdot,r)}$ is the reference state. The variable $\bar{x}$ is the state of the nominal system, $f(\bar{x}_i, u_i)$, which includes Eq. (4.1) to Eq. (4.6) but excludes external disturbances f_{ext} and τ_{ext}. Furthermore, the nominal model only takes the thrust-generated torques into account, namely the G_1 related term in Eq. (4.5). The G_2 related torques will only be incorporated into the inner loop controller design, following a similar approach as in [26]. Torques related to G_3 are omitted due to their negligible effects. The symbol $\sim$ is the Pontryagin set difference, i.e., Given two sets $A \subseteq \mathbb{R}^n$ and $B \subseteq \mathbb{R}^n$, $A \sim B \triangleq \{z \in \mathbb{R}^n | z + y \in A, y \in B\}$. The sets $\mathcal{B}_\Omega(i)$ and $\mathcal{B}_u(i)$ hold tightened constraint conditions. The quaternion error is defined through the multiplicative inverse as $q - q_r = q \otimes q_r^{-1}$, representing the required rotation to transform from the current quaternion state q to the reference quaternion state q_r.

For this nominal system, we have the Lipschitz-continuous assumption on the constraint states, as presented in Assumption 4.3.1 widely adopted in robust NMPC [22, 23, 24, 25].

Assumption 4.3.1 (Lipschitz continuous in Ω) *Let the nominal rotational dynamics be denoted as $\dot{\bar{\Omega}}_B = f_\Omega(\bar{\Omega}_B, u)$. There exist a Lipschitz constant $L > 0$, such that for any $\bar{\Omega}_{B,1}$, $\bar{\Omega}_{B,2} \in [-\Omega_{max}, \Omega_{max}]$ and any $u \in [u_{min}, u_{max}]$, $\|f_\Omega(\bar{\Omega}_{B,1}, u) - f_\Omega(\bar{\Omega}_{B,2}, u)\|_2 \leq L\|\bar{\Omega}_{B,1} - \bar{\Omega}_{B,2}\|_2$.*

Upon solving the optimal control problem defined in Eq. (4.24), the optimal control action, u_0, for the first time step is extracted from $\mathbf{u}_{\text{opt}}$. The required thrust and torques, T_{NMPC} and τ_{NMPC}, are then derived using the nominal form of Eq. (4.5). The state disturbance residue τ_{res} is given by

$$\tau_{\text{res}}(s) = \|I_v^{-1}(\tau_{\text{ext}}(s) + \tau(\omega_c(s)) - \tau_{\text{NMPC}}(s))\|_2. \tag{4.25}$$

This represents the norm of the difference between the optimal nominal and the actual angular accelerations. The external torque disturbance is estimated using Eq. (4.16). Furthermore, the input disturbance residue u_{res} is defined as

$$u_{\text{res}}(s) = \|u_c(\omega_c(s)) - u_0(s)\|_2, \tag{4.26}$$

which is the norm of the difference between commanded and optimal rotor thrust inputs, $u_c(\omega_c)$ and u_0. The command rotor speed ω_c is derived at the end of Section 4.3.1 in Eq. (4.23).

By the Lipschitz condition in Assumption 4.3.1, the difference between the real and the predicted optimal nominal angular velocity, $\Omega_B(t)$ and $\bar{\Omega}_B^0(t)$, under the initial conditions $\Omega_B(t_0) = \bar{\Omega}_B^0(t_0) = \Omega_{B,0}$, can be formulated as

$$\|\bar{\Omega}_B^0(t) - \Omega_B(t)\|_2 \leq \int_{t_0}^{t} L\|\bar{\Omega}_B^0(s) - \Omega_B(s)\|_2 + \tau_{\text{res}}(s)ds. \tag{4.27}$$

The Grownwall-Bellman inequality gives the upper bound of the angular speed error

$$\|\bar{\Omega}_B^0(t) - \Omega_B(t)\|_2 \leq \frac{\bar{\tau}_{\text{res}}}{L}(e^{L(t-t_0)} - 1), \tag{4.28}$$

where $\bar{\tau}_{\text{res}} = \sup_{s\in[0,\infty]} \tau_{\text{res}}(s)$. Motivated by [23, 24, 25], the state and input constraints are tightened by sets

$$\mathcal{B}_\Omega(k) = \{\Omega \in \mathbb{R}^3 | \|\Omega\|_2 \leq \frac{\bar{\tau}_{\text{res}}}{L}(e^{L\frac{T}{N}k} - 1)\}, \tag{4.29}$$

$$\mathcal{B}_u = \{\triangle u \in \mathbb{R}^4 | \|\triangle u\|_2 \leq \bar{u}_{res}\}, \tag{4.30}$$

where $\bar{u}_{\text{res}} = \sup_{s\in[0,\infty]} u_{\text{res}}(s)$. Therefore, the robust satisfaction of nominal constraints is guaranteed when the tightened constraints bound the predicted states and inputs. During implementation, the suprema of τ_{res} and u_{res} are determined from the previous sampled data using a sliding window approach with a time interval smaller than the one over the cut-off frequency of the low-pass filter for the dynamics. This selection is valid as we assume that the disturbances change slower than the low-pass filtered dynamics.

4.4. Numerical Validation

4.4.1. Simulation setup

The simulations are conducted in MATLAB 2022b [58]. The receding horizon optimal control problem is solved by the embedded ACADOS toolbox [59], employing a sequential quadratic programming (SQP) algorithm. The rotor speed control in Eq. (4.23) is accomplished by the MATLAB built-in SQP solver. The NMPC algorithm operates at a frequency of 100 Hz with prediction horizon $T = 1$ s and $N = 20$. The INDI algorithm samples at 250 Hz, matching the sensor frequency. The system dynamics are propagated at a rate of 500 Hz. The simulation spans 20 seconds. We use the same aerodynamic drag model as in [26] and treat it as an unknown external disturbance. All state measurements are filtered by a second-order Butterworth low-pass filter with a 30 Hz cut-off frequency [12]. The actuator dynamics is modelled as a first-order low pass filter with a 20 ms time constant. All controller parameters are listed in Table 4.2.

As the low-pass filtered dynamics have a cut-off frequency 30 Hz, the sliding window for finding the supremum of state and input disturbance residues is set as 0.03 s, slightly smaller than 1/30 s. The Lipschitz constant L is determined through the Monte Carlo simulations and has been determined to be 17. In these simulations, we randomly generate control inputs and states that satisfy the input and state constraints and compute the corresponding system dynamics. The Lipschitz constant is computed for each pair of inputs and states, and we utilize the maximum Lipschitz constant obtained from a total of 10,000 simulations.

Table 4.2: Parameter setup

Parameter	Value
Q_x = Q_N in Eq. (4.24)	diag([200, 200, 500, 1,1,1,0,5,5,200,1,1,1])
Q_u in Eq. (4.24)	diag([6,6,6,6])
k_r, k_y, ρ in Eq. (4.22)	3, 2.2, 1
K_Ω in Eq. (4.22)	diag([0.3, 0.3, 0.3])
W in Eq. (4.23)	diag([5,5,5,1])
u_{min}, u_{max} [N] in Eq. (4.24)	0, 8.5
Ω_{max} [rad/s] in Eq. (4.24)	15

Notably, the state-tightened constraint set is defined as

$$\mathcal{B}_\Omega(k) = \frac{\tau_{res}}{L}(e^{L\frac{T}{N}k} - 1).$$

Equation Eq. (4.31) are the result after substituting Eq. (4.16) and Eq. (4.22) into Eq. (4.4),

$$I_v\dot{\Omega}_B = -\Omega_B \times I_v\Omega_B + \tau_{\text{NMPC}} + \tau_{\text{tilt}}. \tag{4.31}$$

$$\tau_{\text{tilt}} = I_v(2k_r \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} q_{e,red} + 2k_y n_y \sin\Big(\frac{\rho\alpha_y}{2}\Big) + K_\Omega\Omega_{B,e}) \tag{4.32}$$

where τ_{tilt} consists of first three tunable terms of Eq. (4.22).

To ensure the feasibility of the tightened angular velocity constraint set, it is important to keep τ_{res} a small value. If control allocation is executed perfectly, no compromise in yaw performance occurs. In such a scenario, the disturbance residue τ_{res} becomes exactly zero due to the inherent nature of the INDI formulation. On the contrary, with the implementation of the tilt-priority controller, τ_{res} roughly equates to $\|I_v^{-1}\tau_{\text{tilt}}\|$. Consequently, the feasibility of the constraint set hinges primarily on the design of gains in the tilt-priority controller. This prompts questions concerning the necessity of using tilt-prioritized attitude control. In this context, opting for smaller gains in the tilt-prioritized control is preferred to ensure appropriate feasibility. For this simulation, the control gains in the tilt-prioritized controller are normalized by $\frac{|\tau_{\text{tilt}}|_2}{|\tau_{\text{ext}}|_2}$ at every sampling step if the original gains lead to $|\tau_{\text{tilt}}|_2 \geq |\tau_{\text{ext}}|_2$. As a result, with normalized gains, the norm of actual τ_{tilt} will always be smaller than the norm of τ_{ext}.

To mitigate the potential issue of overly aggressive manoeuvres caused by the tilt-prioritized controller, a trigger is employed. For instance, when the quadrotor is at the top of a vertical trajectory, the commanded thrust vector might point in the opposite direction of the current position. Attempting to execute a 180-degree flip within the brief INDI sampling interval of 1/250 seconds would be excessively aggressive. Therefore, a trigger is added to deactivate the tilt-prioritized controller when the reduced quaternion error in Equation 4.19 has an angle of more than 90 degrees. In such case, the commanded thrust will be directly taken from the NMPC control action T_{NMPC}.

4.4.2. Reference Trajectories

The study presented in [26] used two types of reference trajectories which are in the form of horizontal and vertical ellipsoids to assess tracking performance. Evaluating the performance on these simpler trajectories can provide insights into the controller's behaviour on complex trajectory shapes. The trajectories, given by Eq. (4.33) and Eq. (4.34), were directly adopted from [26].

$$\zeta_r(t) = \begin{bmatrix} r_{\text{max}}\sin(kt) & r_{\text{min}}\cos(kt) & 5 \end{bmatrix}^T \tag{4.33}$$

$$\zeta_{r,v}(t) = \begin{bmatrix} r_{\text{max}}\sin(kt) & 0 & 5 + r_{\text{min}}\cos(kt) \end{bmatrix}^T \tag{4.34}$$

where

$$r_{\max} = \frac{V_{\max}}{a} \tag{4.35}$$

$$r_{\min} = \frac{r_{\max}}{n} \tag{4.36}$$

$$k = \frac{a_{\max}}{V_{\max}} \tag{4.37}$$

$V_{\max}$ and $a_{\max}$ are maximum velocity and acceleration, and n is the ellipticity.

For the horizontal trajectory Eq. (4.33), the desired yaw angle ψ_r is given by $-\pi/2 - kt$. The following derivations provide the reference states and commands for the horizontal trajectory. The linear velocity and acceleration are derived first:

$$\dot{\zeta}_r = \begin{bmatrix} r_{\max}\cos(kt)k & -r_{\min}\sin(kt)k & 0 \end{bmatrix}^T \tag{4.38}$$

$$\ddot{\zeta}_r = \begin{bmatrix} -r_{\max}\sin(kt)k^2 & -r_{\min}\cos(kt)k^2 & 0 \end{bmatrix}^T = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T. \tag{4.39}$$

Next, the desired Euler angles are calculated, and the reference quaternion can be computed from these Euler angles:

$$\theta_r = \text{atan2}(\frac{(a_x\cos(\psi_r) + a_y\sin(\psi_r))}{(a_z+g)}), \tag{4.40}$$

$$\phi_r = \text{atan2}(\frac{\cos(\theta_r)(a_x\sin(\psi_r) - a_y\cos(\psi_r))}{a_z+g}). \tag{4.41}$$

The time derivatives of yaw, pitch and roll angles are listed below.

$$\dot{\psi}_r = -k \tag{4.42}$$

$$\dot{\theta}_r = \frac{a_z+g}{(a_x\cos(\psi_r) + a_y\sin(\psi_r))^2 + (a_z+g)^2}(\dot{a}_x\cos(\psi_r) - a_x\sin(\psi_r)\dot{\psi}_r + a_y\cos(\psi_r)\dot{\psi}_r + \dot{a}_y\sin(\psi_r)) \tag{4.43}$$

$$\dot{\phi}_r = \frac{(a_z+g)}{(\cos(\theta_r)(a_x\sin(\psi_r) - a_y\cos(\psi_r)))^2 + (a_z+g)^2}(-\sin(\theta_r)\dot{\theta}_r(a_x\sin(\psi_r) - a_y\cos(\psi_r)) + \cos(\theta_r)(\dot{a_x}\sin(\psi_r) + a_x\cos(\psi_r)\dot{\psi}_r - \dot{a_y}\cos(\psi_r) + a_y\sin(\psi_r)\dot{\psi}_r)) \tag{4.44}$$

Finally, the reference Body-axis rates $\Omega_{B,r}$ are given by Eq. (4.45). Obtaining the explicit form of reference body-axis accelerations $\dot{\Omega}_{B,r}$ is computationally complex. Therefore, the finite difference method is employed to approximate the angular accelerations.

$$\Omega_{B,r} = \begin{bmatrix} 1 & 0 & -\sin(\theta_r) \\ 0 & \cos(\phi_r) & \sin(\phi_r)\cos(\theta_r) \\ 0 & -\sin(\phi_r) & \cos(\phi_r)\cos(\theta_r) \end{bmatrix} \begin{bmatrix} \dot{\phi}_r \\ \dot{\theta}_r \\ \dot{\psi}_r \end{bmatrix} \tag{4.45}$$

Recall that the nominal quadrotor dynamics are described by:

$$\ddot{\zeta} = \frac{Tz_B(q)}{m} + g,$$

and

$$I_v\dot{\Omega}_B = -\Omega_B \times I_v\Omega_B + \tau.$$

By inverse these dynamics, the reference input u_r can be expressed as:

$$u_r = G_1^{-1}\begin{bmatrix} a_x^2 + a_y^2 + (a_z+g)^2 \\ I_v\dot{\Omega}_{B,r} + \Omega_{B,r} \times I_v\Omega_{B,r} \end{bmatrix}. \tag{4.46}$$

Table 4.3: Horizontaol trajectories [26].

a_{max}[m/s^2]$	n=1				n=2				n=5			
	$V_{max}\ [m/s]$				$V_{max}\ [m/s]$				$V_{max}\ [m/s]$			
	5	10	15	20	5	10	15	20	5	10	15	20
10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
20	✓	✓	✓	✓	✓	✓	✓	✓	**x**	✓	✓	✓
30	✓	✓	✓	✓	**x**	✓	✓	✓	**x**	**x**	✓	✓
40	✓	✓	✓	✓	**x**	✓	✓	✓	**x**	**x**	**x**	✓

The tested trajectories are outlined in Tables 4.3 and 4.4 with **x** marking the infeasible trajectories due to the maximum acceleration limits. A total of 40 horizontal trajectories and 36 vertical trajectories are subjected to testing.

Table 4.4: Vertical trajectories [26].

a_{max}[m/s^2]	n=1				n=2				n=5			
	$V_{max}\ [m/s]$				$V_{max}\ [m/s]$				$V_{max}\ [m/s]$			
	5	10	15	20	5	10	15	20	5	10	15	20
10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
30	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

4.4.3. Testing Cases

Reference trajectories in the previous section are subjected to both disturbance-free and disturbances cases. These disturbances are characterized by the form $1 - \cos$, as required by aerospace certification criteria [34] and will be applied at the first 5 seconds. Distinct tests were conducted to assess robustness against force and torque disturbances. The external force is applied along the x and -z-axis in the inertia frame, while the external torque is exerted in the body frame's x and y directions. The constraint violation criterion triggers if angular velocities exceed the nominal thresholds.

4.5. Results

Simulation results for the robustness tests are summarized in Table 4.5. The maximum angular velocities observed in these robustness tests are 18.76 rad/s and 19.24 rad/s from our approach and the SOTA method, exceeding 25.07% and 28.33% of the nominal state constraints. Our method consistently showcases improved position tracking performance across all types and all levels of disturbances compared to the SOTA results. As the magnitudes of disturbances increase, both methods exhibit an increase in the average heading error. The SOTA results in smaller heading errors for 0.2 and 0.3 Nm external torque cases. In essence, this outcome suggests that the robust thrust control aspect of our method effectively ensures position tracking, albeit at the expense of yaw angle tracking accuracy when facing substantial disturbances.

In terms of robustly satisfying state constraints, our methodology demonstrates improvements. Specifically, when external torques are exerted, the frequency of constraint violation incidents in our approach is reduced compared to the state-of-the-art (SOTA) findings. Conversely, the introduction of external forces led to a rise in constraint violation instances, especially evident under the 5 and 10 N test scenarios. We postulate that this uptick in constraint violations is primarily due to the tilt-prioritized controller. This controller may introduce artificial torque disturbances, culminating in abrupt spikes in angular velocity.

Table 4.5: Robustness test results under external torques and forces.

	Position RMSE [m] (mean±std)		Heading Error [°] (mean±std)		Constraint Violation Cases	
	OUR	SOTA	OUR	SOTA	OUR	SOTA
Baseline	**0.14±0.075**	0.17±0.081	**1.72±1.96**	2.04±2.51	6	6
0.1 Nm	**0.14±0.075**	0.17±0.081	**2.63±1.95**	2.73±2.46	**6**	8
0.2 Nm	**0.14±0.075**	0.16±0.080	3.87±3.77	**3.45±3.20**	**6**	7
0.3 Nm	**0.14±0.075**	0.17±0.078	6.65±6.84	**5.48±5.69**	**8**	9
5 N	**0.16±0.062**	0.18±0.066	**2.70±3.06**	2.93±3.43	8	**7**
10 N	**0.19±0.050**	0.22±0.049	**3.58±4.27**	4.12±5.00	**11**	13
15 N	**0.37±1.05**	0.44±1.17	**5.23±7.60**	6.10±7.92	15	**13**

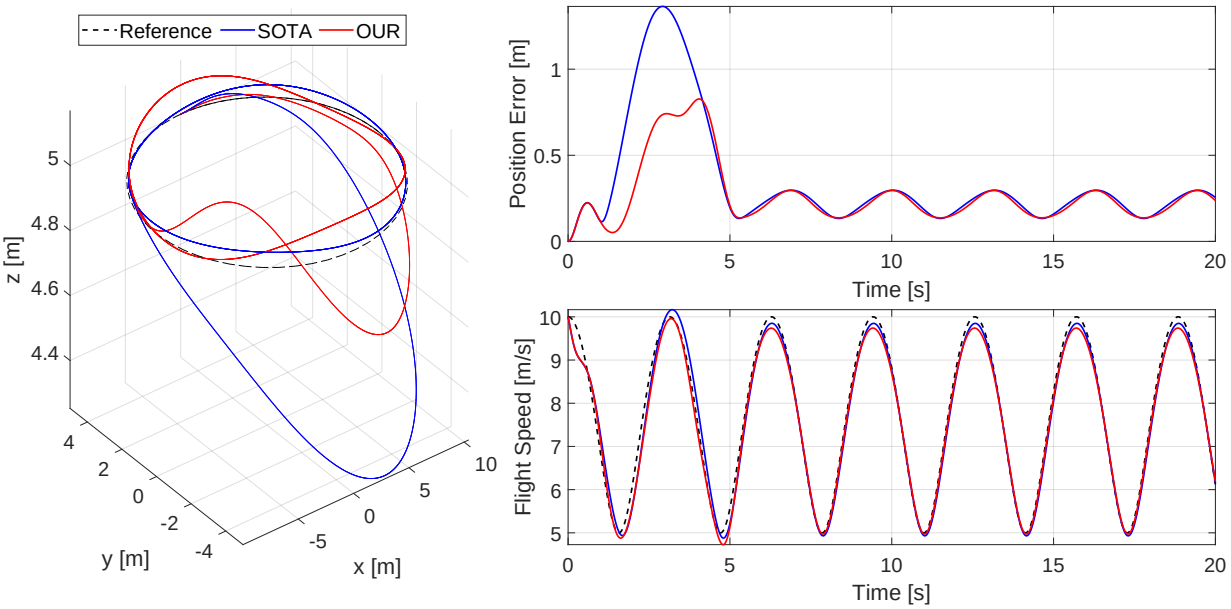


Figure 4.3: Horizontal trajectory tracking performance for case $a_{\max} = 10$, $V_{\max} = 10$, and $n = 2$ with 15 N force disturbances.

Figures 4.3 and 4.6 depict the tracking performance of both controllers when subjected to 15N distur-

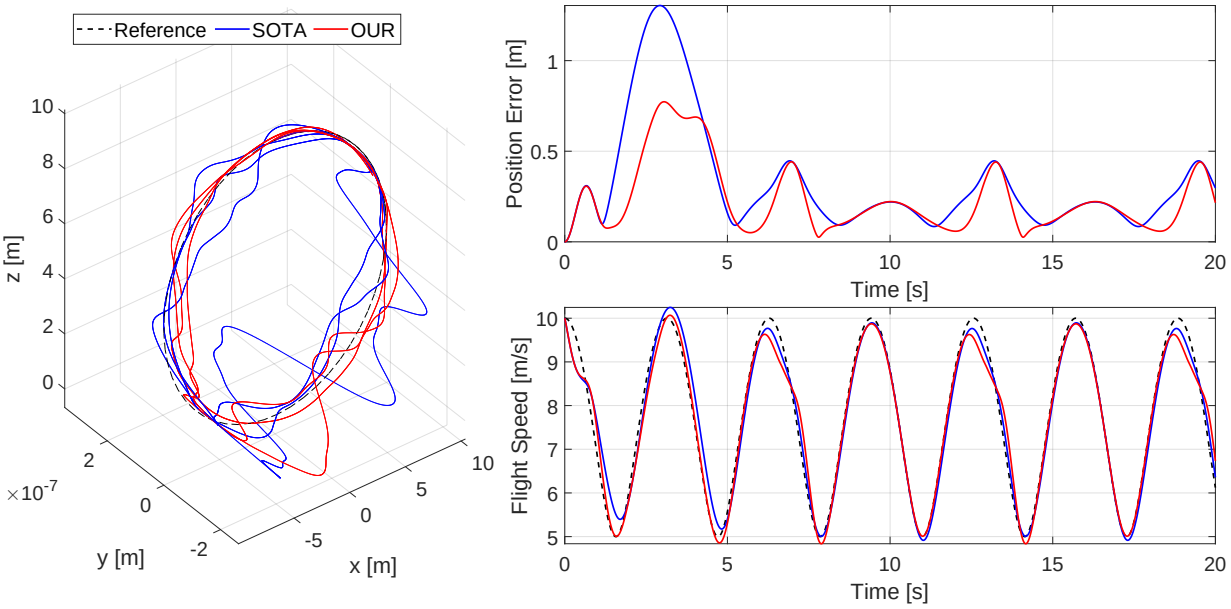


Figure 4.4: Vertical trajectory tracking performance for case $a_{\text{max}} = 10$, $V_{\text{max}} = 10$, and $n = 2$ with 15 N force disturbances.

bances across two trajectories. Disturbance-induced spikes are evident in both horizontal and vertical trajectories between 0 to 5s of the simulation. However, the proposed method adeptly mitigates these spikes, ensuring consistently low position errors throughout the simulation.

In some cases, both methods exhibit comparable performance. For instance, the horizontal trajectory subjected to a 0.3Nm disturbance, as depicted in Fig. 4.5, reveals position errors over time that are notably similar for both methods. However, when the same trajectory shape is adapted vertically under an identical disturbance, our method distinctly outperforms the alternative, as illustrated in Fig. 4.6.

The average results presented in Table 4.5, along with detailed case studies depicted in Figures 4.3, 4.4, 4.5, and 4.6, convincingly demonstrate that our approach offers superior state constraint robustness under torque disturbances. Furthermore, it excels in position tracking despite uncertainties stemming from actuator dynamics and additive disturbances.

4.6. Stability Analysis

In all simulation trials, our controller exhibited stable behaviour. Nevertheless, from a theoretical standpoint, the omission of the terminal constraint in Eq. (4.24) during the controller design phase implies that analytical stability cannot be assured.

4.7. Future Works

The robust NMPC scheme presented in Eq. (4.24) does not guarantee stability due to its lack of terminal constraints. To address this, our final analysis will consider the NMPC scheme without terminal conditions, allowing us to evaluate the performance quantitatively using the performance index, denoted as α_N from Theorem 3.8.6. This index serves as an indicator of the stability of the undisturbed NMPC.

While extending the prediction horizon and increasing the number of prediction stages is a potential method to ensure stability, it is computationally demanding. Such extensions could lengthen the runtime. Empirical studies have shown that by keeping angular velocities below 15 rad/s, we can enhance the tracking performance. Since the primary objective of the quadrotor is position tracking, integrating position constraints might further optimize its performance.

A potential but computationally intensive approach to ensure stability is to extend the prediction horizon and increase the number of prediction stages. However, this may come at the cost of increased runtime. An angular velocity constraint imposition stems from empirical findings suggesting that maintaining angular

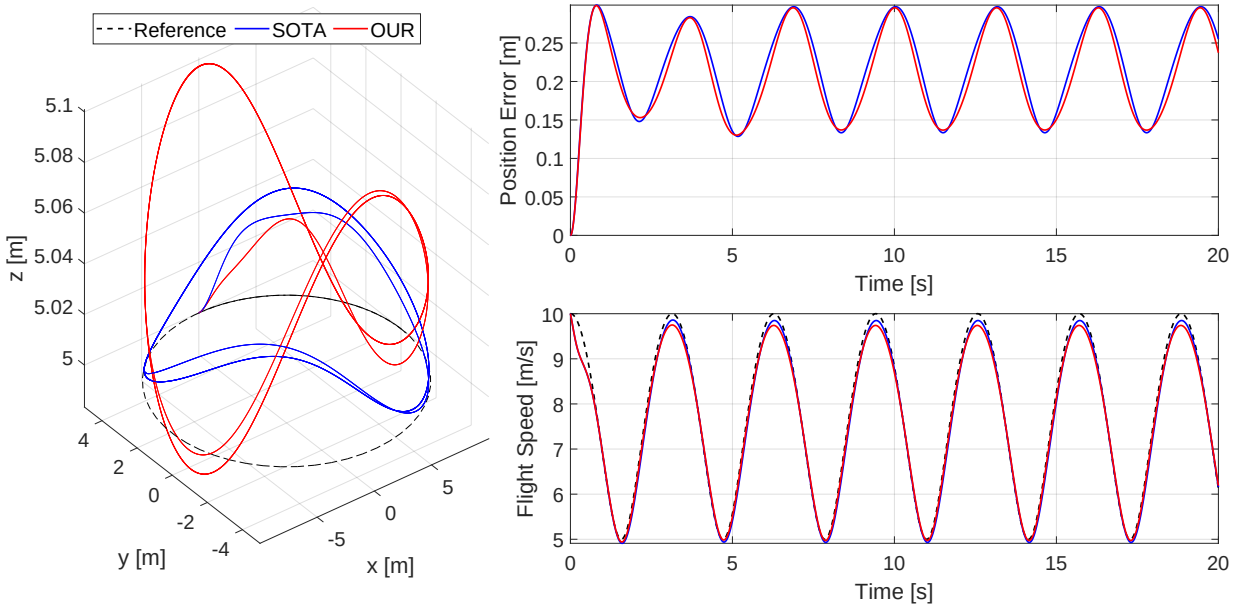


Figure 4.5: Horizontal trajectory tracking performance for case $a_{\text{max}} = 10$, $V_{\text{max}} = 10$, and $n = 2$ with 0.3 Nm torque disturbances.

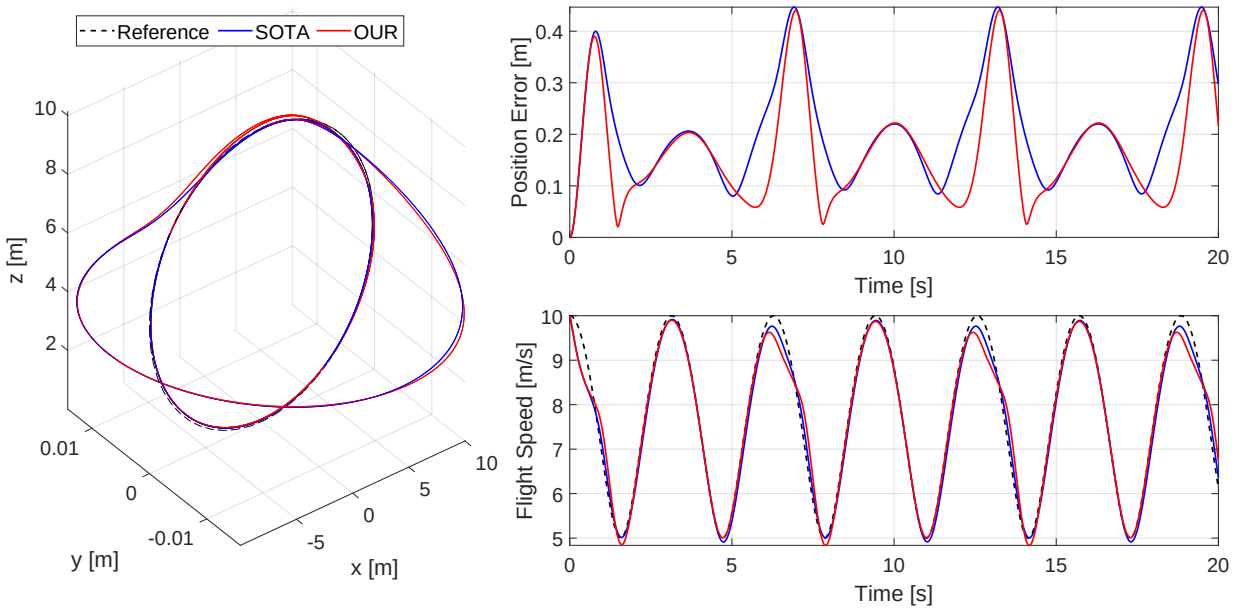


Figure 4.6: Vertical trajectory tracking performance for case $a_{\text{max}} = 10$, $V_{\text{max}} = 10$, and $n = 2$ with 0.3 Nm torque disturbances.

velocities below 15 rad/s yields improved performance. Given that the quadrotor's primary objective is position tracking, the inclusion of position constraints might yield further enhancements. However, it's important to note that adding more constraints might further increase the Lipschitz constant, rendering tightened constraints infeasible.

Within MATLAB, a 20-second simulation takes roughly 60 seconds to execute, largely attributed to the use of a slower sequential quadratic programming solver. The runtime could be diminished by using the real-time sequential quadratic programming solver in the Acodos toolbox and transitioning to more efficient, lower-level programming languages like C++ or C, which better optimize computational memory.

Part III

Additional Results

5

Simulations from two NMPC Setups

This Chapter provides additional simulations, building upon the foundation established in quadrotor control methodology from Section 2.6 and 2.7 and preliminary studies from Section 4.3 and 4.4. Our primary objective is to investigate the performances of two distinct NMPC schemes: one that incorporates a terminal cost and another that omits it.

While the simulation setups and test cases for the quadrotor applications are drawn directly from the configurations delineated in the scientific article Section 2.7, a notable modification has been made: the introduction of a terminal cost. Therefore, at its core, this chapter offers a comparative analysis between (5.1) (without terminal cost) and (5.2) (with terminal cost). It's pertinent to mention that throughout these comparisons, the INDI structure remains consistent and unaltered.

$$\begin{aligned} &\mathbf{u}_{\text{opt}}(x) = \underset{u}{\text{argmin}} \sum_{i=0}^{N-1} (||\bar{x}_i - x_{i,r}||_{Q_x} + ||u_i - u_{i,r}||_{Q_u}) \\ &\text{subject to } \dot{\bar{x}}(t) = f(\bar{x}(t)) + g(\bar{x}(t))u(\bar{x}(t)), \\ &\bar{x}_i \in \mathbb{X} \sim \mathcal{B}_x(i),\ u_i \in \mathbb{U} \sim \mathcal{B}_u(i) \end{aligned} \tag{5.1}$$

$$\begin{aligned} &\mathbf{u}_{\text{opt}}(x) = \underset{u}{\text{argmin}} \sum_{i=0}^{N-1} (||\bar{x}_i - x_{i,r}||_{Q_x} + ||u_i - u_{i,r}||_{Q_u} \\ &\qquad\qquad + ||\bar{x}_N - x_{N,r}||_{Q_x} \\ &\text{subject to } \dot{\bar{x}}(t) = f(\bar{x}(t)) + g(\bar{x}(t))u(\bar{x}(t)), \\ &\bar{x}_i \in \mathbb{X} \sim \mathcal{B}_x(i),\ u_i \in \mathbb{U} \sim \mathcal{B}_u(i) \end{aligned} \tag{5.2}$$

According to [20], introducing a terminal cost can potentially expand the domain of attraction. Moreover, this terminal cost may cause the NMPC feedback loop to guide the system towards the terminal equilibrium or a designated reference with greater speed, as suggested by [52]. Through our simulations, we aim to corroborate these assertions from the literature.

5.1. Tracking Agile Trajectories

Simulation results for quadrotor race track tracking tasks are presented in Table 5.1. The race tracks used are identical to those described in Section 2.7. The findings indicate only a marginal variance between the controllers: up to 6% differences in position error and 2% differences in heading error across all three testing cases. Figures 5.1 to 5.3 compare results between the proposed controllers with and without a terminal cost. Upon closer examination of these results, the controller without terminal cost displays enhanced precision in position tracking across all three race tracks.

On the other hand, introducing a terminal cost improves heading angle tracking. As the race track's agility increases, with RACE3 reaching speeds of up to 20 m/s, the heading tracking precision of the

Table 5.1: Average position and heading errors for tracking race tracks

	Average Position Error [m]		Average Heading Error [°]	
	No Terminal Cost	Terminal Cost	No Terminal Cost	Terminal Cost
RACE1	**0.071**	0.075	5.57	**5.55**
RACE2	**0.070**	0.074	8.12	**7.97**
RACE3	0.074	0.074	**10.24**	10.29

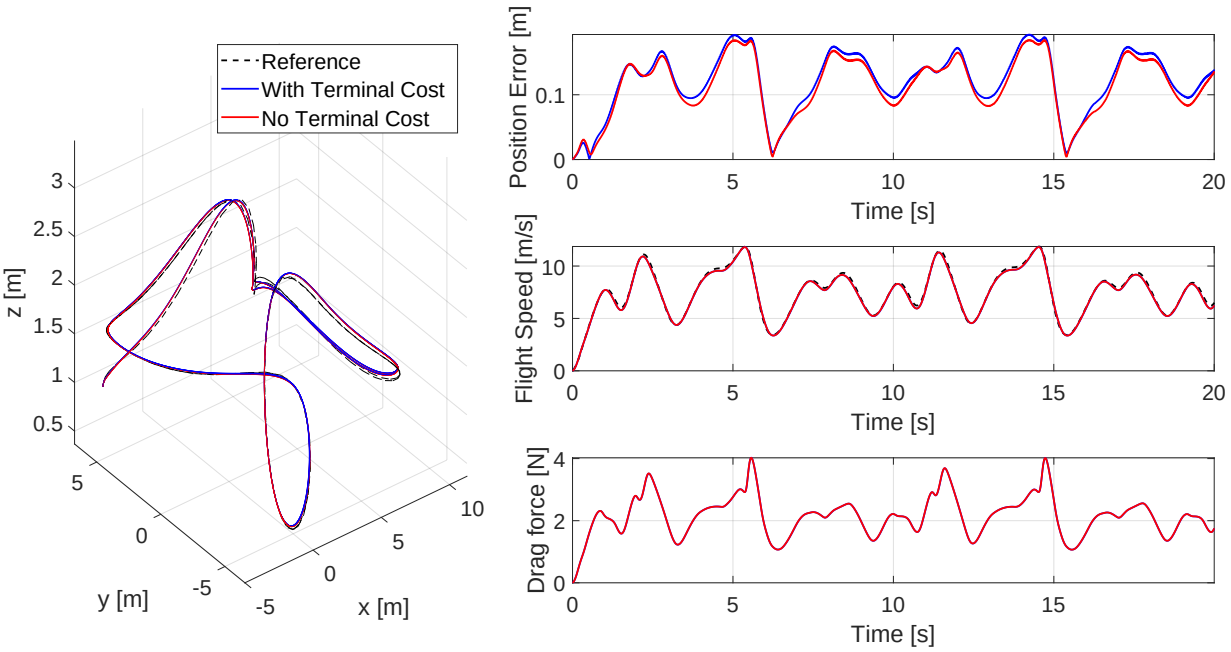


Figure 5.1: RACE1 results: comparing controllers with and without a terminal cost. The figure on the left shows the 3D trajectories, and the figures on the right have the evolutions of position error, flight speed, and drag force.

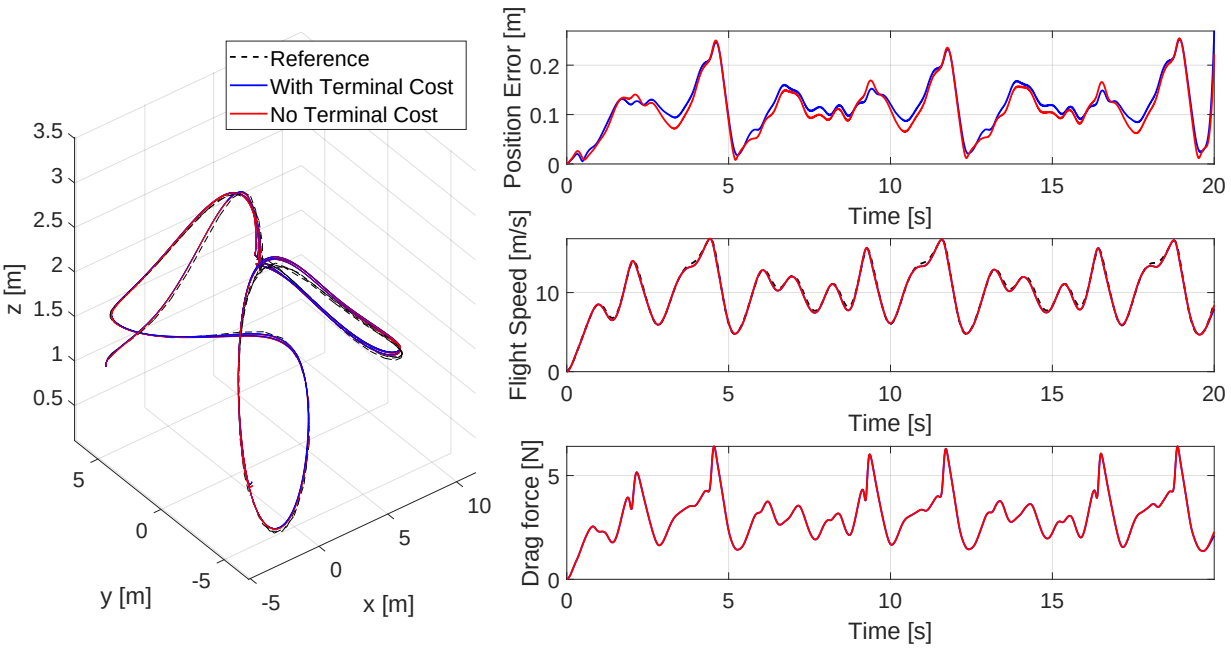


Figure 5.2: RACE2 results: comparing controllers with and without a terminal cost. The figure on the left shows the 3D trajectories, and the figures on the right have the evolutions of position error, flight speed, and drag force.

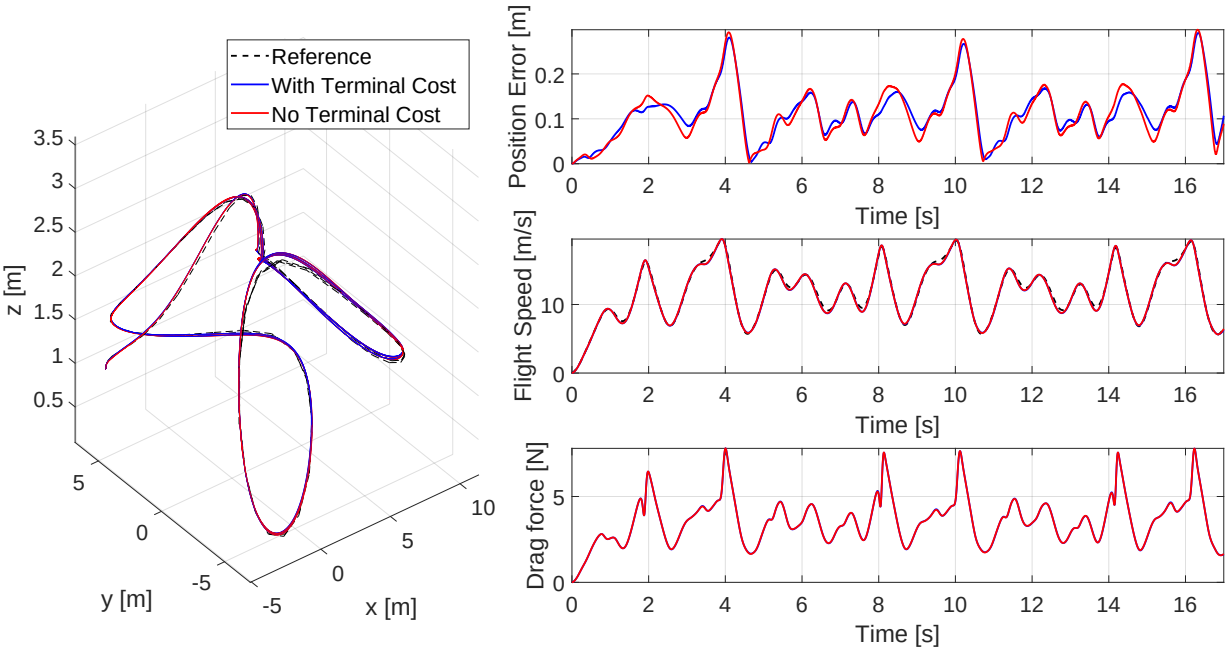


Figure 5.3: RACE3 results: comparing controllers with and without a terminal cost. The figure on the left shows the 3D trajectories, and the figures on the right have the evolutions of position error, flight speed, and drag force.

controller without the terminal cost emerges as superior. This suggests that, given the parameters employed and in disturbance-free scenarios, incorporating a terminal cost doesn't significantly alter the system's closed-loop performance.

5.2. Tracking Agile Trajectory with an Unknown Slung Load

Table 5.2: Average position and heading errors for tracking agile trajectories with an unknown slung load.

	Average Position Error [m]		Average Heading Error [°]	
	No Terminal Cost	Terminal Cost	No Terminal Cost	Terminal Cost
RACE1	**0.045**	0.049	7.14	**7.13**
RACE2	0.14	0.14	15.32	**14.39**

After examining the nominal cases, we tested both controllers under the condition where the quadrotor carried an unknown slung load, as detailed in Section 2.7. Notably, neither controller could track RACE3, as the quadrotor couldn't generate sufficient thrust to counteract disturbances. The results of the feasible cases are summarized in Table 5.2. The largest difference in position error between the two controllers is within 9%, and the maximum heading error difference is within 7%.

Visualizations of the trajectories, position errors, speeds and disturbances throughout the simulation are provided in Figs. 5.4 and 5.5. For the RACE1 case, the controller without a terminal cost exhibits minor position errors across the entire simulation. In contrast, for the RACE2 case, the position errors from both controllers overlap, with the implementation of a terminal cost improving the heading error by 6.46%. Overall, the subtle differences between the results suggest comparable robustness between the two controllers.

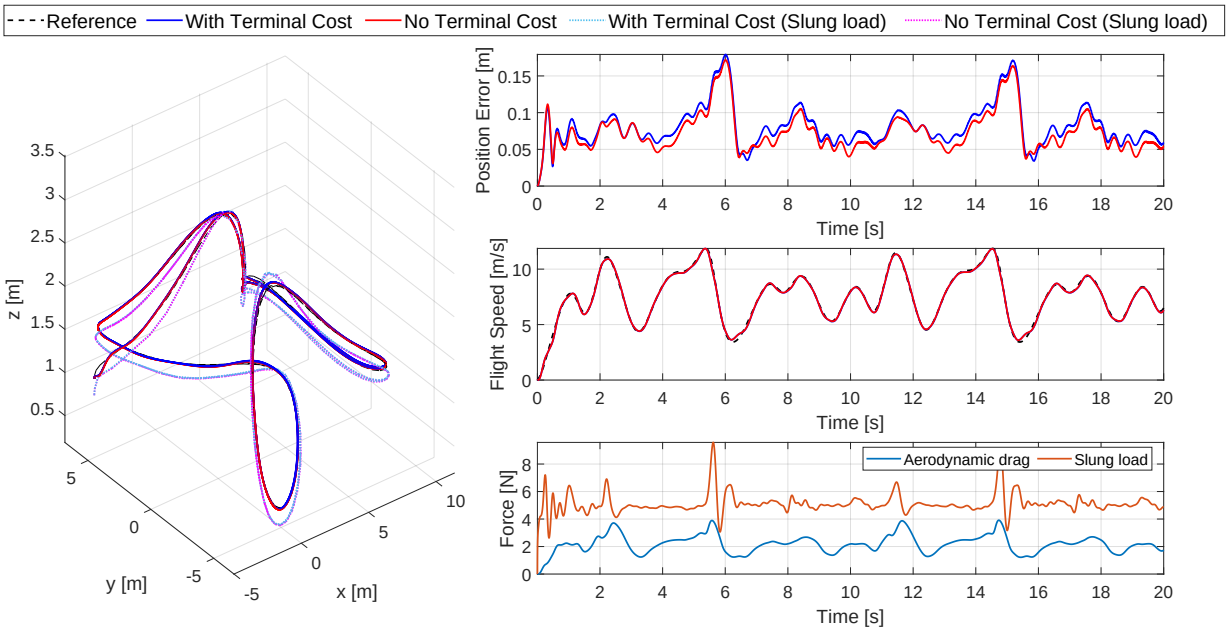


Figure 5.4: RACE1 results with an unknown slung load: comparing controllers with and without a terminal cost. The figure on the left shows the 3D trajectories. The figures on the right have the evolutions of position error, flight speed, drag force, and slung load induced force.

5.3. Robustness Analysis

We conducted comprehensive robustness tests on both controllers under various disturbances. A total of 76 ellipsoidal trajectory sets were tested, as detailed in Section 4.4.2. Results related to position error, heading error, and instances of constraint violations are summarized in Table 5.3.

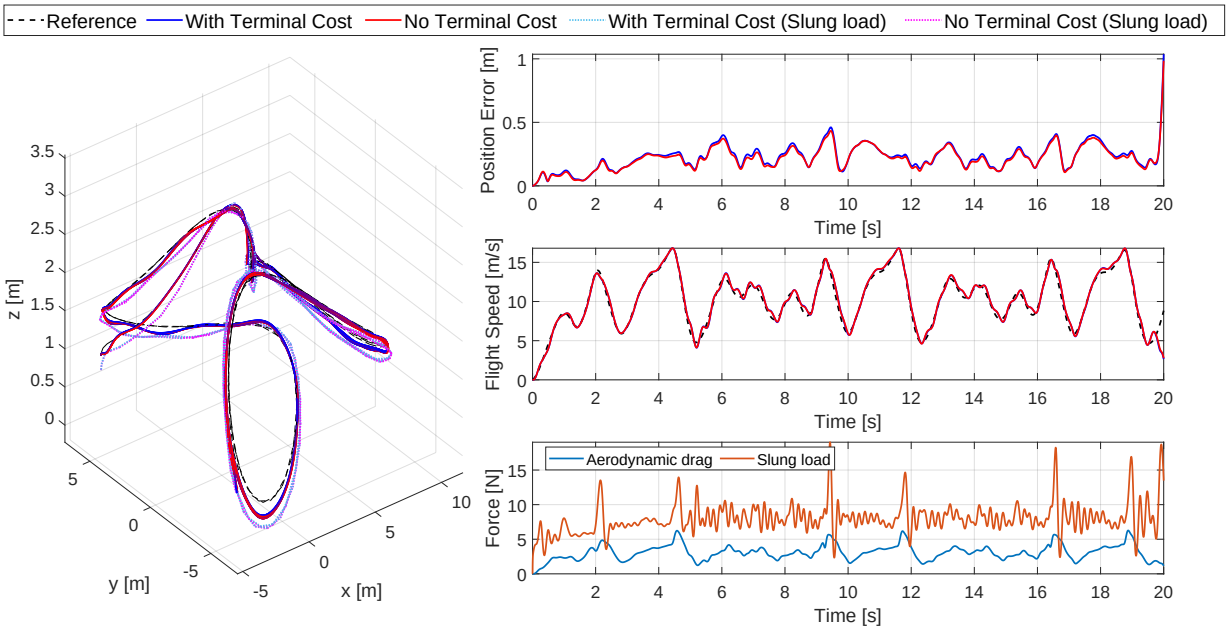


Figure 5.5: RACE2 results with an unknown slung load: comparing controllers with and without a terminal cost. The figure on the left shows the 3D trajectories. The figures on the right have the evolution of position error, flight speed, drag force and slung load induced force.

In terms of heading tracking, the controller with the terminal cost demonstrated superior precision, although the enhancement is subtle. Both controllers performed similarly in position tracking. Their robustness in constraint satisfaction was also comparable when exposed to the same disturbances, with one exception. Under 15 N disturbances, the controller without a terminal cost exhibited one less instance of constraint violation.

In summary, the difference in performance between the two proposed controllers—both with and without terminal cost—is minimal, with both showing comparable tracking accuracy and robustness.

Table 5.3: Robustness test results under external torques and forces.

Terminal Cost	Position RMSE [m] (mean±std)		Heading Error [°] (mean±std)		Constraint Violation Cases	
	No	Yes	No	Yes	No	Yes
Baseline	0.14±0.076	0.14±0.075	1.90±2.26	**1.72±1.96**	6	6
0.1 Nm	0.14±0.076	0.14±0.075	2.84±2.24	**2.63±1.95**	6	6
0.2 Nm	0.14±0.076	0.14±0.075	3.99±3.66	**3.87±3.77**	6	6
0.3 Nm	0.14±0.076	0.14±0.075	6.78±6.71	**6.65±6.84**	8	8
5 N	0.16±0.063	0.16±0.062	2.90±3.17	**2.70±3.06**	8	8
10 N	0.19±0.050	0.19±0.050	3.71±4.24	**3.58±4.27**	11	11
15 N	0.40±1.37	**0.37±1.05**	5.54±7.57	**5.23±7.60**	**14**	15

Part IV

Closure

6

Conclusions

In this concluding section, we return to the research questions posed in Section 1.2, providing comprehensive answers based on the evidence and analyses presented throughout the study.

1. **What types of disturbances should be considered and included?**

   In this thesis, the systems under consideration are assumed to have continuous dynamics and inputs. Such continuity allows most disturbances, such as numerical errors, external disturbances, uncertain parameters, and model measurement errors, to be represented as additive disturbances. A counterpoint might suggest that command input does not necessarily have to be continuous. While it is accurate that our optimal control approach generates discrete control actions, we have addressed this by integrating actuator dynamics, framed as a first-order lag system, during state propagation. As such, when the command input is filtered through the actuator dynamics, the resulting real input is rendered continuous.

   In the context of an aeroelastic wing, the predominant disturbances stem from wind gusts. To emulate this, we've incorporated horizon wind gusts represented by sinusoidal velocities. To further our robustness tests, vertical gust load disturbances and pitch moment disturbances are also added, both characterized by sinusoidal signals.

   In the quadrotor applications, actuator dynamics are interpreted as disturbances arising from model uncertainties, primarily due to the precise actuator models. In real-world experiments, a quadrotor inevitably experiences the impacts of aerodynamic drag. This drag force becomes significant during agile flights. While we have modelled the aerodynamic drag based on empirical data from the literature, our nominal model does not include this model. As a result, the aerodynamic drag is regarded as an additive force disturbance. To test the robustness of our controller, we subjected the quadrotor to external force and torque disturbances, described by a $1 -$ cos form and of varying amplitudes. The comprehensive inclusion of these disturbances guarantees that our simulations are of high fidelity.

2. **How can a stable cost function be constructed for the robust NMPC?**

   To convert the optimal cost function into a Lyapunov function and thereby ensure stability, there are two prevalent methods.

   The first method employs a terminal cost and positive invariant terminal constraints. By doing so, the terminal cost is designed to be a local Lyapunov function within the terminal constraints. Such conditions intrinsically guarantee that the optimal cost becomes a Lyapunov function and that nominal stability is achieved. To further ensure robust stability, the terminal constraints are designed again to be positively invariant even when subject to disturbances. Overall, with predesigned terminal conditions, the system is input-to-state practical stable. However, formulating these terminal costs and constraints for nonlinear systems poses significant challenges. Often, the validation of stability assumptions is relegated to simulations.

   The second method drops the terminal conditions, and the nominal stability relies on the inherent boundedness of infinite horizon optimal costs, optimal stage costs and optimal costs. If all assumptions are satisfied, there exists a performance index α_N. If $\alpha_N \in (0, 1]$, the closed-loop

system is asymptotically stable. When fortified with additional assumptions concerning bounded disturbances, this methodology paves the way for the establishment of robust stability in the sense of semiglobally practically asymptotically stable.

The current implementation follows the second method using NMPC without terminal constraints. Nominal stability is achieved by designing a quadratic stage cost function that intrinsically meets the boundedness criteria of the optimal stage cost. Further, the stability of the NMPC without terminal conditions is quantified using the performance index. However, the robust stability is only validated via simulations by comparing results with state-of-the-art methods in the literature.

3. **What types of incremental nonlinear control are to be synchronized with robust NMPC?**

   With the consideration of low model dependency and the capability of capturing both matched and unmatched disturbances, several candidates of increment control have been selected. These included the incremental sliding mode control (ISMC), the incremental nonlinear backstepping (INBS), and the incremental nonlinear dynamic inversion (INDI).

   While ISMC offers the advantage of low model dependency, it is not without its drawbacks. A significant issue with ISMC is the chattering phenomenon. Moreover, its low model dependency is achieved at the expense of utilizing high control gain. As such, ISMC may not be the most suitable choice. INBS is hindered by a scarcity of literature regarding its stability and limited documented applications. Conversely, INDI stands out with its comprehensive theoretical foundation. Its efficacy has been demonstrated through its widespread use in aerospace applications. Notably, when INDI is loosely integrated with NMPC, there is already a marked enhancement in performance. Given INDI's sophisticated theoretical development and proven effectiveness in disturbance rejection, it is one of the most suitable choices for integration with robust NMPC.

4. **How can the selected incremental nonlinear control be utilized to filter out disturbances and improve robustness?**

   As a sensor-based method, INDI is adept at online disturbance approximation using the nominal dynamics and real-time state measurements. The incremental control law formed by INDI effectively filters out these estimated disturbances. For underactuated systems, control allocation strategies are implemented with the objective of minimizing disturbance residue. While complete disturbance rejection might be elusive due to the underactuation and coupled dynamics, the control mechanism ensures a significant reduction in disturbance. Given that the disturbance residue is invariably smaller than the actual disturbance, this approach enhances system robustness.

5. **How to formulate tightened constraints to ensure robustness?**

   The constraint states are assumed to be Lipschitz continuous. Leveraging the disturbance residue combined with this continuity assumption allows us to establish an upper bound on the disturbance effects impacting the constraint states. This is achieved through the application of the Gronwall-Bellman inequality. Subsequently, the Pontryagin set difference can be employed to subtract the disturbance effects from the nominal constraints. The states that are confined in the tightened state constraints will never exceed nominal constraints, even if disturbances are presented. Therefore, robustness in constraint satisfaction is achieved.

Research objectives

The research objectives posed in Chapter 1 are repeated below for convenience.

> **"Development of nonlinear incremental optimal control for underactuated mechanical systems."**

During the work described in this thesis, a tightly coupled robust NMPC with INDI controller has been developed and evaluated. The proposed control framework was customized for two distinct underactuated mechanical systems: the aeroelastic wing and the quadrotor.

For the aeroelastic wing application, simulation studies highlighted its efficiency compared to the nominal NMPC, achieving a reduction of 37.60% in plunge motion errors and 40.00% in pitch motion errors, when a sinusoidal wind gust was presented.

Shifting the focus to the quadrotor, the controller was fine-tuned for trajectory tracking tasks. The outcomes were then compared with a benchmark control that loosely coupled NMPC with INDI. Extensive simulations were conducted, where the quadrotor was tasked to track agile trajectories under various disturbances. The results showed up to 79.58% reduction in position error and up to 44.08% reduction in heading error.

7

Recommendations

Recommendations for further research include the following. The constraint tightening procedure hinges on the assumption of Lipschitz continuity. However, there's a potential that the Lipschitz constant can be prohibitively large for practical implementation. In light of this, it's important to derive a less conservative estimation that more accurately bounds the impact of disturbances on the states. Regarding the command input, a control allocation strategy is requisite. Given that this method inherently accounts for the input constraints, there is potential to relax the tightened input constraints within the robust NMPC framework. However, before making any adjustments, further investigations are essential to understand the implications of removing the input constraint tightening procedure.

When discussing the control allocation strategy, particularly in the context of the quadrotor application, it's noteworthy that the dimension reduction from a 6×1 control input, which encompasses both thrust and torque vectors, to a 4×1 control input, containing only thrust magnitude and a torque vector, results in the omission of the coupling effects between translational and rotational dynamics. Our innovatively proposed tilt-prioritized INDI attitude control regains this coupling effect by allowing the attitude controller to track both the force vector and the desired attitude. This approach comprehensively addresses the challenges associated with underactuation. The mapping of thrust magnitude and toque vector to actual rotor speed is in full rank, and most of the time, disturbances can be rejected precisely. However, for the two states and one input aeroelastic wing model, addressing the coupling between lift and moment within the controller design remains a significant challenge. A notable concern is that any effort to reject force disturbances could inadvertently introduce artificial moment disturbances, further complicating the control process. Currently, the reliance is on the control allocation to navigate the coupling problem raised by the underactuation. Therefore, we recommend future research focus on the development of a novel controller that tackles the coupling problem as the tilt-prioritized controller does. On the other hand, a more refined control allocation strategy could also be a viable solution.

The sensor noise was not simulated in our study. While the inclusion of a Butterworth filter can effectively mitigate sensor noise, incorporating this noise directly into the system is recommended. This addition would further enhance the fidelity of the simulations. One of the inherent limitations of the NMPC is its comparatively slower runtime, attributed to its computational demands. In our study, the algorithm's execution time was not incorporated into the controller design considerations. Yet, when transitioning from simulations to real-world applications, ensuring the speed of the control algorithm becomes paramount. We recommend that future research should emphasize finding the optimal prediction stages and horizons to strike a balance between performance and speed. For the simulation setup, it is our assertion that forthcoming research should prioritize the utilization of the ACADOS toolbox over the SQP solver provided by MATLAB. This recommendation stems from our observation that solvers in ACADOS exhibit superior computational efficiency compared to the aforementioned MATLAB solver.

Lastly, the robust stability of the proposed method remains partially unaddressed, given the inherent theoretical complexities. Users are granted discretion in their choice to incorporate terminal conditions within the NMPC framework or not. In scenarios where NMPC is deployed with terminal conditions, there's an explicit need to formulate a solution that identifies positive invariant terminal constraints. Alternatively, when employing NMPC without terminal conditions, we suggest that the robust stability of our controller could be built upon the concepts of practical stability and robust-optimal feasibility.

References

[1] Reza Olfati-Saber. “Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles”. PhD thesis. Massachusetts Institute of Technology, 2001.

[2] J. Hauser et al. “Nonlinear control via approximate input-output linearization: the ball and beam example”. In: *IEEE Transactions on Automatic Control* 37.3 (1992), pp. 392–398. DOI: 10.1109/9.119645.

[3] R. Olfati-Saber. “Global configuration stabilization for the VTOL aircraft with strong input coupling”. In: *IEEE Transactions on Automatic Control* 47.11 (2002), pp. 1949–1952. DOI: 10.1109/TAC.2002.804457.

[4] Neil Holden Getz. “Dynamic inversion of nonlinear maps with applications to nonlinear control and robotics”. PhD thesis. University of California at Berkeley, 1995.

[5] Mark W. Spong. “Energy Based Control of a Class of Underactuated Mechanical Systems”. In: *IFAC Proceedings Volumes* 29.1 (1996). 13th World Congress of IFAC, 1996, San Francisco USA, 30 June - 5 July, pp. 2828–2832.

[6] H. Ashrafiuon et al. “Shape change maneuvers for attitude control of underactuated satellites”. In: *Proceedings of the 2005, American Control Conference, 2005.* 2005, 895–900 vol. 2. DOI: 10.1109/ACC.2005.1470073.

[7] Héctor Ríos et al. “Robust regulation for a 3-DOF Helicopter via sliding-modes control and observation techniques”. In: *Proceedings of the 2010 American Control Conference*. 2010, pp. 4427–4432. DOI: 10.1109/ACC.2010.5531051.

[8] Christian Hoffmann et al. “A Survey of Linear Parameter-Varying Control Applications Validated by Experiments or High-Fidelity Simulations”. In: *IEEE Transactions on Control Systems Technology* 23.2 (2015), pp. 416–433. DOI: 10.1109/TCST.2014.2327584.

[9] Yu Liu et al. “Integrated LPV control for underactuated ship linear track keeping and main diesel engine”. In: *2008 7th World Congress on Intelligent Control and Automation*. 2008, pp. 1840–1845. DOI: 10.1109/WCICA.2008.4593203.

[10] A.A.G. Siqueira et al. “Nonlinear H_∞ control via quasi-LPV representation applied in a underactuated manipulator”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 3. 2002, 2169–2174 vol.3. DOI: 10.1109/IRDS.2002.1041588.

[11] Ewoud JJ Smeur et al. “Cascaded incremental nonlinear dynamic inversion for MAV disturbance rejection”. In: *Control Engineering Practice* 73 (2018), pp. 79–90.

[12] Ezra Tal et al. “Accurate Tracking of Aggressive Quadrotor Trajectories Using Incremental Nonlinear Dynamic Inversion and Differential Flatness”. In: *IEEE Transactions on Control Systems Technology* 29.3 (2021), pp. 1203–1218. DOI: 10.1109/TCST.2020.3001117.

[13] Paul Acquatella et al. “Incremental backstepping for robust nonlinear flight control”. In: *Proceedings of the EuroGNC* 2013 (2013).

[14] Xuerui Wang et al. “Incremental backstepping sliding mode fault-tolerant flight control”. In: *AIAA Scitech 2019 Forum*. 2019, p. 0110.

[15] Xuerui Wang et al. “Incremental Sliding-Mode Fault-Tolerant Flight Control”. In: *Journal of Guidance, Control, and Dynamics* 42.2 (2019), pp. 244–259. DOI: 10.2514/1.G003497. eprint: https://doi.org/10.2514/1.G003497. URL: https://doi.org/10.2514/1.G003497.

[16] Xuerui Wang et al. “Stability analysis for incremental nonlinear dynamic inversion control”. In: *Journal of Guidance, Control, and Dynamics* 42.5 (2019), pp. 1116–1129.

[17] Byoung-Ju Jeon et al. “Understandings of classical and incremental backstepping controllers with model uncertainties”. In: *IEEE Transactions on Aerospace and Electronic Systems* 56.4 (2019), pp. 2628–2641.

[18] David Q Mayne et al. “Constrained model predictive control: Stability and optimality”. In: *Automatica* 36.6 (2000), pp. 789–814.

[19] Lalo Magni et al. “Stability margins of nonlinear receding-horizon control via inverse optimality”. In: *Systems & Control Letters* 32.4 (1997), pp. 241–245.

[20] D. Limon et al. “On the stability of constrained MPC without terminal constraint”. In: *IEEE Transactions on Automatic Control* 51.5 (2006), pp. 832–836. DOI: 10.1109/TAC.2006.875014.

[21] Lars Grüne et al. “Stability and Suboptimality Without Stabilizing Constraints”. In: *Nonlinear Model Predictive Control: Theory and Algorithms*. London: Springer London, 2011, pp. 113–163. DOI: 10.1007/978-0-85729-501-9_6.

[22] D.L. Marruedo et al. “Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties”. In: *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.* Vol. 4. 2002, 4619–4624 vol.4. DOI: 10.1109/CDC.2002.1185106.

[23] Kunwu Zhang et al. “Robust Nonlinear Model Predictive Control Based Visual Servoing of Quadrotor UAVs”. In: *IEEE/ASME Transactions on Mechatronics* 26.2 (2021), pp. 700–708. DOI: 10.1109/TMECH.2021.3053267.

[24] Matteo Rubagotti et al. “Robust model predictive control with integral sliding mode in continuous-time sampled-data nonlinear systems”. In: *IEEE Transactions on Automatic Control* 56.3 (2010), pp. 556–570.

[25] Huahui Xie et al. “Disturbance Rejection MPC Framework for Input-Affine Nonlinear Systems”. In: *IEEE Transactions on Automatic Control* 67.12 (2022), pp. 6595–6610. DOI: 10.1109/TAC.2021.3133376.

[26] Sihao Sun et al. “A comparative study of nonlinear mpc and differential-flatness-based control for quadrotor agile flight”. In: *IEEE Transactions on Robotics* 38.6 (2022), pp. 3357–3373.

[27] R.E. Kalman. “On the general theory of control systems”. In: *IFAC Proceedings Volumes* 1.1 (1960). 1st International IFAC Congress on Automatic and Remote Control, Moscow, USSR, 1960, pp. 491–502.

[28] S. Bouabdallah et al. “Design and control of an indoor micro quadrotor”. In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004*. Vol. 5. 2004, 4393–4398 Vol.5. DOI: 10.1109/ROBOT.2004.1302409.

[29] S. Bouabdallah et al. “Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor”. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. 2005, pp. 2247–2252. DOI: 10.1109/ROBOT.2005.1570447.

[30] Emil Fresk et al. “Full quaternion based attitude control for a quadrotor”. In: *2013 European Control Conference (ECC)*. 2013, pp. 3864–3869. DOI: 10.23919/ECC.2013.6669617.

[31] Dario Brescianini et al. *Nonlinear quadrocopter attitude control: Technical report*. Tech. rep. ETH Zurich, 2013.

[32] Mark Wilfried Mueller. “Multicopter attitude control for recovery from large disturbances”. In: *arXiv preprint arXiv:1802.09143* (2018).

[33] Dario Brescianini et al. “Tilt-Prioritized Quadrocopter Attitude Control”. In: *IEEE Transactions on Control Systems Technology* PP (Dec. 2018), pp. 1–12. DOI: 10.1109/TCST.2018.2873224.

[34] *Easy Access Rules for Large Aeroplanes*. Standard. European Union Aviation Safety Agency 2023, 2022.

[35] Xiping Zhang et al. “Stability analysis of LPV time-delayed systems”. In: *International journal of control* 75.7 (2002), pp. 538–558.

[36] Jeff S Shamma. “An overview of LPV systems”. In: *Control of linear parameter varying systems with applications* (2012), pp. 3–26.

[37] László Bencsik et al. “Stabilization of internal dynamics of underactuated systems by periodic servo-constraints”. In: *International Journal of Structural Stability and Dynamics* 17.05 (2017), p. 1740004.

[38] Pham Van Trieu et al. “Adaptive fractional-order fast terminal sliding mode with fault-tolerant control for underactuated mechanical systems: Application to tower cranes”. In: *Automation in Construction* 123 (2021), p. 103533.

[39] Vadim I. Utkin. “Manipulator Control System”. In: *Sliding Modes in Control and Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 239–249.

[40] S. J. Mija et al. “Reaching law based sliding mode control for discrete MIMO systems”. In: *2010 11th International Conference on Control Automation Robotics & Vision*. 2010, pp. 1291–1296. DOI: 10.1109/ICARCV.2010.5707278.

[41] A. Bartoszewicz. “Discrete-time quasi-sliding-mode control strategies”. In: *IEEE Transactions on Industrial Electronics* 45.4 (1998), pp. 633–637. DOI: 10.1109/41.704892.

[42] Katarzyna Adamiak. “Reference Sliding Variable Based Chattering-Free Quasi-Sliding Mode Control”. In: *IEEE Access* 8 (2020), pp. 133086–133094. DOI: 10.1109/ACCESS.2020.3010900.

[43] P. Kachroo. “Existence of solutions to a class of nonlinear convergent chattering-free sliding mode control systems”. In: *IEEE Transactions on Automatic Control* 44.8 (1999), pp. 1620–1624. DOI: 10.1109/9.780438.

[44] Rong-Jong Wai. “Fuzzy Sliding-Mode Control Using Adaptive Tuning Technique”. In: *IEEE Transactions on Industrial Electronics* 54.1 (2007), pp. 586–594. DOI: 10.1109/TIE.2006.888807.

[45] Q.P. Ha et al. “Fuzzy sliding-mode controllers with applications”. In: *IEEE Transactions on Industrial Electronics* 48.1 (2001), pp. 38–46. DOI: 10.1109/41.904548.

[46] Vadim Utkin. “Discussion Aspects of High-Order Sliding Mode Control”. In: *IEEE Transactions on Automatic Control* 61.3 (2016), pp. 829–833. DOI: 10.1109/TAC.2015.2450571.

[47] Daniel Liberzon. “The Linear Quadratic Regulator”. In: *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. USA: Princeton University Press, 2011. Chap. 6, pp. 180–199.

[48] Chen Ci. “On the Robustness of the Linear Quadratic Regulator via Perturbation Analysis of the Riccati Equation”. PhD thesis. Dublin City University, 2014.

[49] Russ Tedrake. *Underactuated Robotics. Algorithms for Walking, Running, Swimming, Flying, and Manipulation*. 2023. URL: https://underactuated.csail.mit.edu.

[50] E.G. Gilbert et al. “Linear systems with state and control constraints: the theory and application of maximal output admissible sets”. In: *IEEE Transactions on Automatic Control* 36.9 (1991), pp. 1008–1020. DOI: 10.1109/9.83532.

[51] J.B. Rawlings et al. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2017. Chap. 2, pp. 89–172.

[52] Lars Grüne et al. “Feasibility and Robustness”. In: *Nonlinear Model Predictive Control: Theory and Algorithms*. London: Springer London, 2011, pp. 211–250. DOI: 10.1007/978-0-85729-501-9_6.

[53] T. Parisini et al. “A receding-horizon regulator for nonlinear systems and a neural approximation”. In: *Automatica* 31.10 (1995), pp. 1443–1451. DOI: https://doi.org/10.1016/0005-1098(95)00044-W. URL: https://www.sciencedirect.com/science/article/pii/000510989500044W.

[54] Andrew R. Teel. “Discrete Time Receding Horizon Optimal Control: Is the Stability Robust?” In: *Optimal Control, Stabilization and Nonsmooth Analysis*. Ed. by Marcio S. de Queiroz et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 3–27.

[55] H. Chen et al. “A game theoretic approach to nonlinear robust receding horizon control of constrained systems”. In: *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)*. Vol. 5. 1997, 3073–3077 vol.5. DOI: 10.1109/ACC.1997.612023.

[56] L. Magni et al. “Robust model predictive control for nonlinear discrete-time systems”. In: *International Journal of Robust and Nonlinear Control* 13.3-4 (2003), pp. 229–246.

[57] Davide Martino Raimondo et al. “Min-max Model Predictive Control of Nonlinear Systems: A Unifying Overview on Stability”. In: *European Journal of Control* 15.1 (2009), pp. 5–21. DOI: https://doi.org/10.3166/ejc.15.5-21. URL: https://www.sciencedirect.com/science/article/pii/S0947358009707034.

[58] The MathWorks Inc. *MATLAB version: 9.13.0 (R2022b)*. Natick, Massachusetts, United States, 2022. URL: https://www.mathworks.com.

[59] Robin Verschueren et al. “acados – a modular open-source framework for fast embedded optimal control”. In: *Mathematical Programming Computation* (Oct. 2021). DOI: 10.1007/s12532-021-00208-8. URL: https://doi.org/10.1007/s12532-021-00208-8.