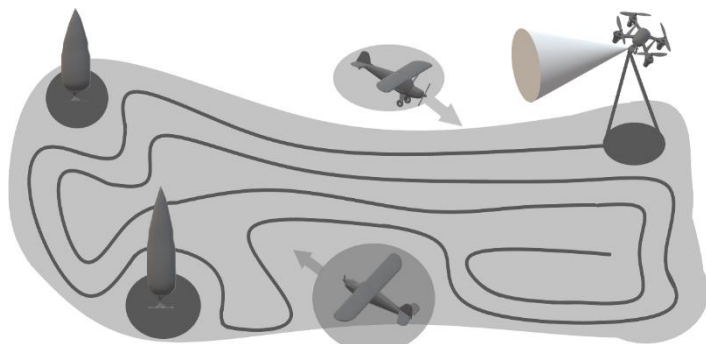
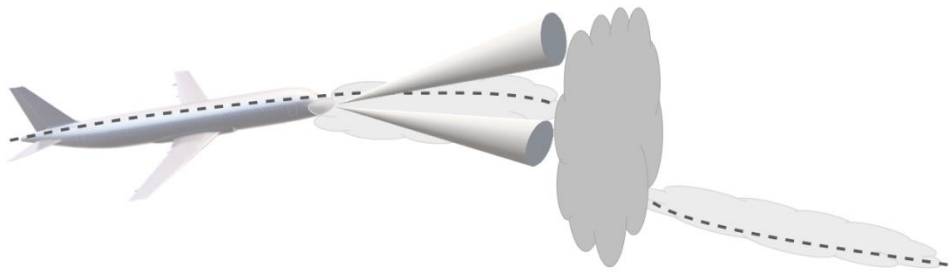




Mohamed Ibrahim

Real-Time Moving Horizon Planning and Control of Aerial Systems under Uncertainties



Real-Time Moving Horizon Planning and Control of Aerial Systems under Uncertainties

Dissertation

zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)

von

Mohamed Ibrahim, M.Sc.

geboren am 5. Jan. 1987 in Minia, Egypt

genehmigt durch die Fakultät für Elektrotechnik und Informationstechnik der
Otto-von-Guericke-Universität Magdeburg

Gutachter:

Prof. Dr.-Ing. Rolf Findeisen

Prof. Ilya Kolmanovsky, Ph.D.

Priv.-Doz. Dr. sc. techn. ETH habil. Eric Bullinger

eingereicht am 30. September 2020

Promotionskolloquium am 02. November 2020

Contents

Abstract	vii
Deutsche Kurzfassung	ix
List of abbreviations and symbols	xi
1 Introduction, Motivations, and Objectives	1
1.1 Outline and Contributions	4
2 Aircraft Planning and Control Systems	6
2.1 Aircraft, Aircraft Design, and Gust Load	6
2.2 Autonomous Aerial Vehicles	8
2.3 Aircraft Guidance and Control Systems	9
2.4 Guidance and Motion Planning Systems	11
2.5 Flight Control System	13
2.6 Load Alleviation System	15
2.7 Aircraft Modeling, Linearization, and Reduction	16
2.8 Summary	17
3 Model Predictive Control	18
3.1 Overview of Model Predictive Control	18
3.2 Basic Constituents and Mathematical Formulation	21
3.3 Theoretical Aspects	24
3.4 Implementation Aspects	27
3.5 Summary	31
4 Model Predictive Control for Gust Load Alleviation	32
4.1 Introduction and Motivations	32
4.2 Model Predictive Control for Load Alleviation	35
4.3 Implementation	45
4.4 Wind-Tunnel Validation	51
4.5 Discussion and Conclusion	57
5 Moving Horizon Path Planning	58

5.1	Introduction and Motivation	58
5.2	Mixed Integer Programming Moving-Horizon Planning	60
5.3	Robust Moving-Horizon planning	66
5.4	Planning for Area Coverage	71
5.5	Simulation Results	76
5.6	Discussion and Conclusion	78
6	Fusing Planning and Path Following Control	81
6.1	Introduction and Motivation	81
6.2	Robust Planning and Control with Guarantees	82
6.3	Discussion and Conclusion	94
7	Conclusion and Future Perspectives	95
7.1	Directions for Future Research	96
Appendix A	Gust Load Alleviation	98
A.1	Green Regional Aircraft States	98
A.2	Aircraft Parameters	98
Appendix B	Quadcopter UAV Case-Study	100
Bibliography		101

Abstract

The growing use of aeronautical systems, spanning from passenger airplanes to autonomous drones, increases the awareness for ecological and economic aspects, e.g., to increase autonomy, endurance, energy efficiency, and flexibility. Therefore, the aviation industry aims to develop advanced (robust/reliable) planning and control systems to guarantee flexible operation, flight safety, and robustness despite a multitude of disturbance and environmental challenges.

This work proposes model predictive control as a suitable method to tackle combined planning and control challenges for aircraft applications. Model predictive control allows systematically considering the vehicle dynamics and constraints, e.g., load envelopes and obstacle avoidance, as well as optimizing a performance index, such as energy consumption, tracking error, ride comfort, and load reduction. Furthermore, the predictive nature makes model predictive control suitable for the motion planning and usable in each of the main control layers present in aerospace systems.

The first part of this work considers the problem of load alleviation. Load alleviation system aims to reduce the structural loads, oscillation, and the passenger discomfort resulting from atmospheric turbulence. Model predictive control allows including preview of gust information ahead of the aircraft to improve the prediction accuracy, thereby to improve the load alleviation performance. Using model predictive control for load alleviation allows enlarging the flight envelopes and improving fatigue life and aircraft performance, such as passenger comfort. Reducing the load allows designing new aircraft configurations, like lighter structures, without compromising the safety regulations. Consequently, this reduces the fuel expenditure and thereby the operating cost that makes the aircraft more eco-efficient.

The second part of this thesis presents an approach to interface the planning and control systems of autonomous vehicles moving through uncertain/dynamic environments. The developments aim towards a hierarchical control framework that is robust and flexible, yet safe. The contributions in this part are refinements of existing planning techniques that significantly improve performance and computational efficiency. These refinements speed up the online optimization and allow for online planning on embedded platforms. Constraint tightening is shown to guarantee constraint satisfaction (collision avoidance) despite uncertainty. This planning approach is then applied and expanded towards area coverage.

Fusion of path planning and path following is achieved by “contracts”, which guarantee vehicle safety and constraint satisfaction, e.g., obstacle avoidance despite uncer-

tainty. The design of the proposed controllers takes into account the interconnections in terms of dynamic constraints and reference definitions between them. In particular, a hybrid moving-horizon planner can switch between different controllers in a predictive and optimized way, taking into account the controller capabilities to maintain planning feasibility, i.e., vehicle safety even in cluttered environments.

In summary, this thesis provides an integrative, real-time feasible model predictive control approach to path planning and path following, suitable for embedded implementation. Several experimental results (wind-tunnel and flight-tests) demonstrate that the proposed algorithms successfully overcome various implementation challenges and uncertainties.

Deutsche Kurzfassung

Der vermehrte Einsatz von Luftfahrtsystemen, die von Passagierflugzeugen bis hin zu autonomen Drohnen reichen, führt zu einer verstärkten Betrachtung ökologischer und ökonomischer Aspekte, wie z.B. Autonomie, Reichweite, Energieeffizienz und Flexibilität. Daher ist die Luftfahrtindustrie bestrebt, fortschrittliche (robuste/ zuverlässige) Planungs- und Regelungssysteme zu entwickeln, die einen flexiblen Betrieb, Flugsicherheit und Robustheit, trotz der Vielzahl von Störungen und Umweltveränderungen, gewährleisten.

In dieser Arbeit wird als geeignete Methode zur Bewältigung von kombinierten Planungs- und Steuerungsherausforderungen für Flugzeuganwendungen der Einsatz der modellprädiktiven Regelung vorgeschlagen. Diese ermöglicht die systematische Berücksichtigung der Fahrzeugdynamik und -beschränkungen, z.B. des sogenannten zulässigen Belastungsbereichs und der Vermeidung von Hindernissen, sowie die Optimierung eines Leistungsindex, z.B. des Energieverbrauchs, des Trajektorienfolgefehlers, des Fahrkomforts oder einer Lastenreduktion. Darüber hinaus ist die modellprädiktive Regelung dank ihres vorausschauenden Aspekts auch für die Bewegungsplanung geeignet. Somit ist die modellprädiktive Regelung in allen wichtigen Reglerebenen, die in Luft- und Raumfahrtsystemen vorhanden sind, einsetzbar.

Der erste Teil dieser Arbeit befasst sich mit dem Lastabminderungssystem, das die strukturellen Belastungen und Schwingungen sowie das durch die atmosphärische Turbulenzen verursachte Unwohlsein der Passagiere verringert. Die modellprädiktive Regelung ermöglicht die Berücksichtigung von Böeninformationen vor dem Flugzeug, das die Vorhersagegenauigkeit verbessert und die mögliche Lastabminderung erhöht. Dies wiederum führt zu einer Vergrößerung des zulässigen Flugbereiches und einer reduzierten Materialermüdung, und somit zu einer erhöhten Lebensdauer und besseren Flugzeugleistung sowie einem erhöhten Passagierkomfort. Durch die Reduzierung der Last können neue Flugzeugkonfigurationen wie leichtere Strukturen unter Einhaltung der Sicherheitsbestimmungen entworfen werden. Dies führt zu reduziertem Treibstoffverbrauch und damit niedrigeren Betriebskosten, was das Flugzeug ökoeffizienter macht.

Der zweite Teil dieser Arbeit stellt einen Ansatz zur Verknüpfung von Planungs- und Steuerungssystemen autonomer Fahrzeuge vor, die sich durch unsichere/ dynamische Umgebungen bewegen. Die Entwicklungen führen zu einem hierarchischen Ansatz, der robust, flexibel und dennoch sicher ist. Die Beiträge in diesem Teil sind Verbesserungen bestehender Planungstechniken, mit denen Leistung und Recheneffizienz erheblich ver-

bessert werden können. Diese Verbesserungen beschleunigen die Online-Optimierung und ermöglichen es, die Online-Planung auf eingebetteten Plattformen zu nutzen. Des Weiteren wird gezeigt, dass eine geeignete Anpassung von Beschränkungen die Erfüllung der Kollisionsvermeidung auch unter Unsicherheiten garantiert. Der entwickelte Planungsansatz wird weiterhin auf das Problem der Flächenabdeckung erweitert.

Die Verschmelzung von Pfadplanung und Pfadverfolgung wird durch eine Formulierung als “Verträge” erreicht, die die Fahrzeugsicherheit und die Erfüllung von Beschränkungen gewährleisten. So lassen sich z.B. Hindernisse trotz Unsicherheit vermeiden. Der Reglerentwurf berücksichtigt, dass Pfadplanung und Pfadverfolgung durch dynamische Einschränkungen und Referenzdefinitionen verknüpft sind. Insbesondere kann der hybride Planer, der auf einem beweglichen Horizont agiert, auf prädiktive und optimierte Weise zwischen verschiedenen Regelungskonfigurationen wechseln, wobei die Eigenschaften des Reglers berücksichtigt werden, um die Durchführbarkeit der Planung, auch unter Unsicherheiten in Umgebungen mit hoher Aktivität oder mit vielen Objekten, sicher zu stellen.

Zusammenfassend bietet diese Arbeit einen integrativen, in Echtzeit realisierbaren modellprädiktive Regelungsansatz für die Pfadplanung und die Pfadverfolgung, der für eine eingebettete Implementierung geeignet ist. Die Wirksamkeit und Robustheit der vorgeschlagenen Planungs- und Kontrollstrategien wird anhand von Simulationsergebnissen validiert. Mehrere experimentelle Ergebnisse (Windkanal- und Flugtests) zeigen, dass die vorgeschlagenen Verfahren den Implementierungsherausforderungen und auftretenden Unsicherheiten gerecht werden.

List of abbreviations and symbols

Abbreviations and Acronyms

GLA	Gust Load Alleviation
LAS	Load Alleviation System
LIDAR	Light Detection And Ranging
LMPC	Linear Model Predictive Control
LP	Linear Program
MAL	Maneuver Load Alleviation
MILP	Mixed-integer Linear Program
MIP	Mixed-integer Program
MPC	Model Predictive Control
NMPC	Nonlinear Model Predictive Control
NP-hard	Non-deterministic Polynomial acceptable problems
OCP	Optimal Control Problem
QP	Quadratic Program
UAV	Unmanned Aerial Vehicle

Symbols

t	Time variable
t_k	Recalculation time instant
n_x	Dimension of the real valued state vector
n_y	Dimension of the real valued output vector
n_u	Dimension of the real valued input vector
x	State vector $x \in \mathbb{R}^{n_x}$
y	Output vector $y \in \mathbb{R}^{n_y}$
u	Input vector $u \in \mathbb{R}^{n_u}$
x_0	Vector of the real valued initial state
f	Vector field for the system dynamics $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$
g	Vector field for the system output $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_y}$
\mathbb{X}	Set defining the state constraints $\mathbb{X} \subseteq \mathbb{R}^{n_x}$
\mathbb{U}	Set defining the input constraints $\mathbb{U} \subseteq \mathbb{R}^{n_u}$
\mathbb{X}_T	set the terminal state constraints $\mathbb{X}_T \subseteq \mathbb{R}^{n_x}$
$Q \geq 0$	Positive semi-definite matrix Q
$R > 0$	Positive semi-definite matrix R
$\ x\ $	L^2 -norm of a vector $x \in \mathbb{R}^n$

$\ x\ _Q^2$	Notation for $x^T Q x$, $Q \geq 0$
\bar{x}	Predicted states vector
\bar{y}	Predicted output vector
\bar{u}	Predicted inputs vector
$J(\cdot)$	Cost-function of an optimal control problem
$V(x(\cdot))$	Optimal value of the cost-function for a measured state x at time t_i
$\ell(x(\cdot), u(\cdot))$	Stage cost function $\ell(x(\cdot), u(\cdot)) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}_0^+$
$E(x(\cdot))$	Terminal Cost Function $E(x(\cdot)) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}_0^+$
$u^*(\cdot)$	Optimal input sequence, with $\tau \in [t_i, t_{ii}]$, $t_{ii} > t_i$ starting at initial state $x(t_i)$
N	Prediction horizon
T_s	Sampling time
T_d	Delay time
$(k + j k)$	j -th future time step predicted at the current time step k
ω	uncertainty
ω_g	Gust velocity
κ_T	Terminal control law
δ_i	Control surface i

1 Introduction, Motivations, and Objectives

Driven by ecological and economic considerations, the aviation industry aims towards developing efficient “flexible” aircraft, e.g., to increase endurance, safety, and reliability and to reduce fuel consumption [1–4]. Furthermore, there are appealing applications of unmanned aerial vehicles (UAVs) for scientific and commercial applications, see [5–11]. By now, UAVs are used for search-and-rescue missions [12–14], weather forecast and meteorological monitoring, surveillance, and mapping [15].

Both flexible aircraft and UAVs must operate under a wide variety of operating conditions and sustain different uncertainties and environmental disturbances. Guaranteeing the flight safety and robustness despite these uncertainties and disturbances is essential. For example, flight safety and regulations require maintaining that the vehicle’s position and loads stay within a defined safety region under all conditions [1].

Improving aircraft efficiency requires significant efforts, e.g., new configurations, advanced guidance and control systems, and sensing and actuation technologies. In general, aircraft development is time-consuming, with a typical period spanning up to decades from conception design to first flight. Indeed, the aircraft design and control systems have a significant impact on the weight and performance in addition to the development and maintenance costs [16]. Flexible control system architectures become elementary to meet the environmental and economic requirements. As a result, significant efforts are underway to develop and deploy new control methods for operating aircraft and UAVs.

From the control point of view, there are different control objectives to consider, spanning from disturbance rejection, stabilization to long-term mission planning and fast real-time control. To improve the existing control systems, it becomes important to operate near or at the constraints, such as maximum surface deflections and maximum permissible load [17]. Furthermore, it is essential to develop controllers that can cope in a flexible way with a wide range of operation conditions, constraints, and nonlinearities [17]. So far, only a few control methodologies can simultaneously achieve these requirements in a systematic manner [17].

In this work, we propose model predictive control (MPC) as a suitable control solution to tackle these problems in a unified way. Specially, we consider load alleviation, planning, and low-level control. As shown, MPC has the ability to tackle all these problems by allowing a suitable interaction between the different control layers and the different control objectives.

MPC is based on a repetitive decision-making process, i.e., MPC repeatedly solves

a finite-horizon optimal control problem (OCP) applying the first part of the solution, see e.g., [17–30]. MPC allows taking in a systematic way the vehicle dynamics and constraints into account to optimize performance objectives, such as energy, tracking error, passenger comfort, or load reduction. The predictive principle allows using MPC for both motion planning and load alleviation. By solving an OCP online, using the system model for prediction allows taking the aircraft constraints into account and permits system operation closer to constraint boundaries, thereby improving the performance. In this context, obstacle avoidance constraints can be handled by MPC, allowing achieving safe and autonomous operation.

Furthermore, MPC can be applied hierarchically at different levels, spanning from strategic (planning), supervisory control, to tactical (executional), and low-level control. Moreover, MPC allows for different decision architectures, such as centralized, decentralized, and distributed decision. This allows MPC to consider and handle other agents' future actions and objectives while making local decisions. Due to this flexibility, MPC has revolved as a solution to many constrained multivariate control problems in a wide variety of applications from aerospace, robotics, and chemical processes. This is also driven by the emergence of fast numerical algorithms, see, e.g., [1, 7, 17, 24, 30].

The main purpose of this thesis is to examine the use of MPC for different aircraft applications, underlining its wide applicability.

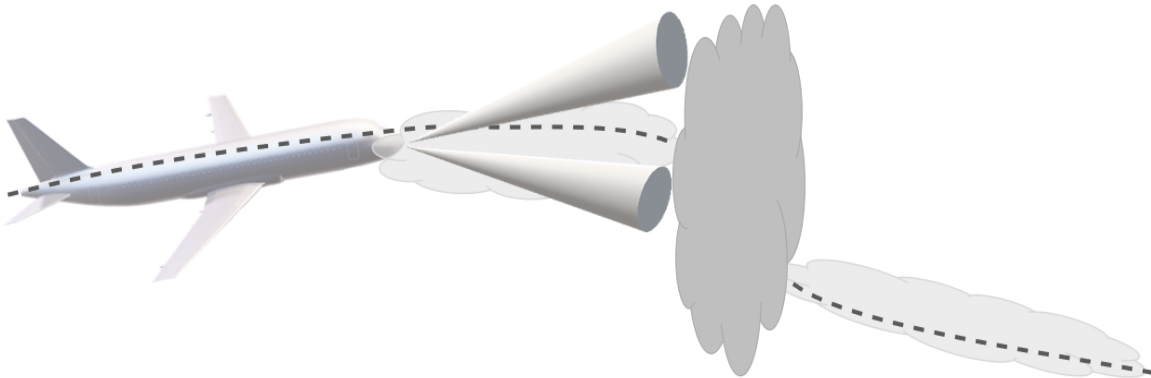


Figure 1.1: Predicting the upcoming gust using LIDAR sensor, which can be included in MPC.

In the first part of this thesis, we propose to use MPC as a suitable control approach for load alleviation, focusing on atmospheric turbulence. Using MPC for load alleviation is motivated by the possibility to include the gust preview information to improve the alleviation performance, see Fig. 1.1. Reducing loads allows considering new aircraft configurations, i.e., lighter structures, without compromising safety regulations. Consequently, this allows reducing the fuel expenditure and thereby the operating cost, which makes the aircraft more eco-efficient.

In the second part of this thesis, we consider the use of MPC for path/mission planning, focusing on UAVs. Most planning approaches, e.g., [6–8, 31], do not take the

vehicle capabilities into account, which might lead to dynamically infeasible paths. This issue is essential, especially in the case of fast dynamics and in changing environments [8]. We propose a model-based planning algorithm, i.e., MPC, which takes the environment and the vehicle dynamics and constraints into account, e.g., maneuverability, velocity, acceleration, and energy limitations, see [32–34]. We also consider different objectives, such as point-to-point movement and area coverage.

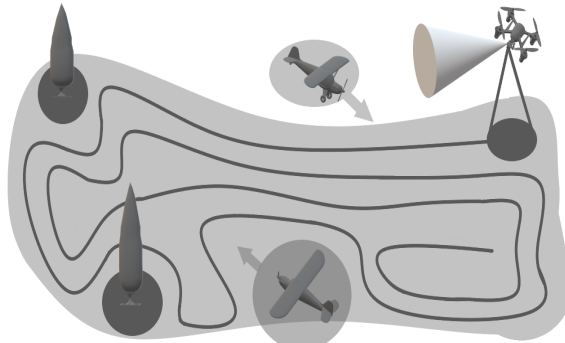


Figure 1.2: Moving-horizon planning approach in a dynamic environment avoiding static and dynamic obstacles [35].

Autonomous vehicles often operate in a cluttered (non-convex) environment, as illustrated in Fig. 1.2. Then, the vehicle’s knowledge about the environment may be incomplete at the start of a mission. This makes the planning of the entire mission challenging if not impossible [6, 8, 36]. MPC allows solving a reactive planning problem in a receding-horizon manner, i.e., to generate/update the path online repetitively using the latest environment information. This leads to an indirect feedback control law that allows compensating environmental disturbances, thereby ensuring collision avoidance even in dynamic environments [32–34].

In practical applications, the vehicle might deviate from the planned trajectory due to uncertainties such as wind disturbance. This causes degradation of the performance and might affect vehicle stability, especially in dynamic environments [6, 8, 31, 37]. To improve the performance, we propose to intertwine the path planning and control. To ensure “consistency” between the different layers, we propose that the moving-horizon planner and the low-level controller agree on a “contract” regarding the precision [33]. The high-level moving-horizon planner takes the contract into account by constraint tightening. Obstacles on the planning are enlarged by a safety set considering the precision contract to generate a collision-free optimal reference. The safety bounds are provided and ensured by a low-level tube-based MPC [38], i.e., under certain conditions, robust constraint satisfaction is guaranteed despite uncertainties considering an adequate level of accuracy.

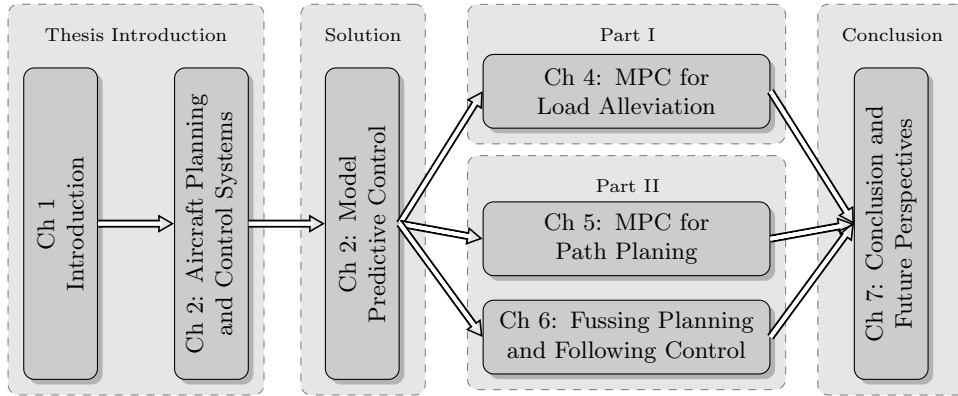


Figure 1.3: Thesis Structure and dependency of each chapter in the thesis.

1.1 Outline and Contributions

As outlined, the main contributions of this thesis can be divided into two main parts: The use and the experimental validations of MPC for load alleviation, and the use of MPC to fuse planning and control in a structured and theoretical way.

The thesis is organized in seven chapters, see Fig. 1.3:

Chapters 1 introduces the motivation, objective, and a summary of the contributions covered in this thesis.

Chapter 2 introduces the essentials of aerial vehicles that are required to understand the work in the following chapters.

Chapter 3 provides an introduction to MPC and outlines the basic components and mathematical formulation of MPC, which forms the base for the developments in this thesis. Besides theoretical aspects such as robustness and stability, we outline some implementation challenges of MPC for aerospace applications, with a special emphasis on the real-time feasibility. Subsequently, we present an overview of some efficient solution methodologies and the available numerical tools used in this work.

In **Chapters 4-6**, we will discuss the design and implementation of MPC for aerial vehicles, spanning from load alleviation and planning to low-level control.

Chapter 4 illustrates the design and real-time implementation of MPC to alleviate atmospheric loads. First, we present the load alleviation problem and briefly review the common existing load alleviation approaches. Then we present the proposed MPC setup for load alleviation. We furthermore discuss how to use the preview information to predict the disturbances. We consider two different aircraft as case studies to demonstrate the effectiveness of the proposed approach through numerical simulations with different gust conditions. We furthermore provide one of the first wind-tunnel investigations to validate the real-time capability of MPC to reduce gust loads.

Chapters 5-6 focus on the use of MPC for moving-horizon planning and control

problem of autonomous vehicles in uncertain dynamic environments. The major contributions are the use of a hierarchical MPC strategy to couple planning and low-level control under flight uncertainties.

Chapter 5 commences with a background survey of the common planning approaches. Contrary to most planning approaches, the proposed moving-horizon planning strategy allows taking the vehicle dynamics and constraints into account. To do so, we reformulate the problem as a mixed-integer programming. We propose a tuple binary encoding that decreases the solution time by reducing the number of binary variables.

Furthermore, a constraint tightening approach is introduced to guarantee constraint satisfaction despite the present uncertainties, with a feasibility proof of the proposed approach. Thereby, each obstacle is enlarged with a safety margin to ensure collision avoidance. We furthermore explain how to solve the area coverage problem. The efficiency and real-time feasibility of the proposed approaches are emphasized through several numerical simulations and experimental results using quadcopter flying in a cluttered environment.

Chapter 6 describes the synthesis of a hierarchical MPC strategy that fuses the planning and control layers. To ensure compatibility between the layers, we consider the problem of recursive feasibility to guarantee vehicle safety and constraint satisfaction, e.g., obstacle avoidance despite uncertainty. A hybrid moving-horizon planning approach is proposed, which allows switching between different low-level controllers in a predictive and optimized way, taking into account the controller capabilities. As such, it allows for more flexible and less conservative solutions to improve the applicability of the planner for maintaining vehicle safety through cluttered environments.

The effectiveness and robustness of the proposed planning and control strategies are validated via simulation results. Although the planning and control methodologies developed in this thesis can be applied to many vehicle types, we focus on applications and scenarios for quadcopter, operating in cluttered/uncertain environments. Experimental results and flight-test demonstrate that the proposed algorithms can be implemented on limited onboard computation and allow handling different uncertainties, e.g., external disturbance and measurement noise.

Chapter 7 concludes with remarks on the thesis contributions and discusses possible future developments.

In summary, this thesis explores the real-time implementation of MPC for the control of aerial vehicles, focusing on load alleviation, planning, and control subject to different environmental uncertainties.

2 Aircraft Planning and Control Systems

The focus of this thesis is to design and implement an MPC scheme for aerial vehicles such as passenger and unmanned aircraft. This chapter aims to introduce the fundamental topics of guidance and flight control systems, spanning from aerodynamics and aircraft modeling to the autopilot and low-level flight control. In doing so, we hope to provide motivations for the research topics presented in this thesis. This chapter describes two parts; the gust load alleviation system for passenger aircraft and planning and control systems for autonomous aerial vehicles.

2.1 Aircraft, Aircraft Design, and Gust Load

The aviation industry is currently driven by a multitude of environmental and economic demands. In response, optimistic targets and stringent regulations are set to reduce the environmental footprint of aviation and improve aircraft efficiency, e.g., reduce pollution emissions [1–4]. For instance, in Europe, ACARE has established broad objectives for air transport over the next few decades in their Flightpath 2050 report [39], while NASA periodically updates its technology roadmap spanning over 20-year horizons [40]. So the aircraft industry aims to develop more efficient and flexible “greener” aircraft, to meet the new challenging requirements concerning quality, safety, and environment impacts [1–4], and to allow widespread use from passenger transport to delivery and surveillance.

Developing aircraft requires passing through many stages; see Fig. 2.1, e.g., conceptual, preliminary, and detailed design process [16]. During these processes, the aircraft specifications are verified and validated at different levels up to the first flight test [16]. The desired qualities are derived from marketing studies (previous experience or customer needs) that define the aircraft characteristics, e.g., payload, range, speed, and maximum take-off weight [16], as well as a multitude of regulations.

All aircraft systems impact the weight and performance in addition to the development and maintenance costs [16]. For this reason, it is typical to take into account the aircraft flight systems, e.g., planning and control systems, during the conceptual design to improve the overall design processes and to prevent the extra work and integration issues. Furthermore, early exploration of system architectures ensures an efficient development process to improve aircraft specifications supported by modern sensing and actuation technologies.

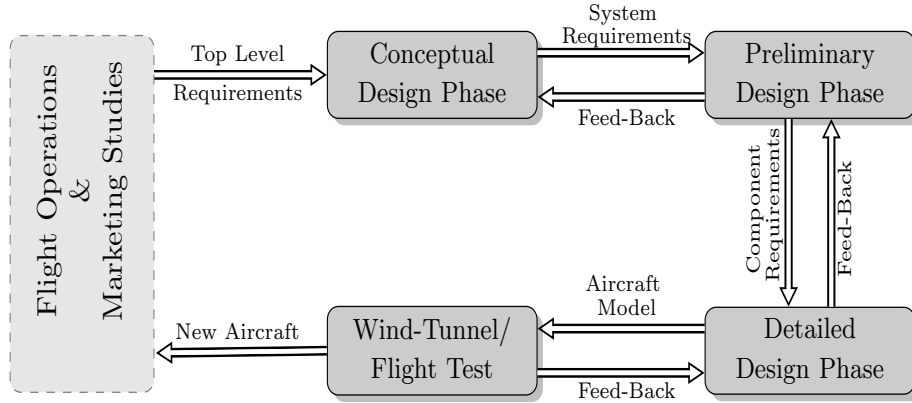


Figure 2.1: Aircraft development cycle.

In the conceptual phase of the aircraft design, different configurations are generated and evaluated. Then the best configurations are selected to proceed to the preliminary design stage according to the performance requirements. Mainly, the aircraft efficiency depends on the aerodynamic performance [41], which can be enhanced by developing new structural configurations [42] or taking the flight dynamics into account. For instance, high-aspect-ratio design leads to maximizing aerodynamic efficiency, i.e., reducing the aerodynamic drag and thereby the fuel consumption [43]. Using the same design concept, there is increasing interest in developing more efficient aircraft, e.g., high-altitude long-endurance [2], high-altitude performance demonstrator [44], and solar-powered aircraft [45], see Fig. 2.2. More complex configurations, so-called *sensorcrafts*, aim at equipping UAVs with all the sensing devices usually included only in large transport aircraft [46], to enable them for long endurance missions [4].

Practically, aircraft are exposed to different operating conditions, system uncertainties, and environmental disturbances [5, 6, 47, 48]. These aircraft are characterized by high structural flexibility, which leads to significant inertial coupling between the structural and rigid-body dynamics [1]. This coupling can lead to critical structural issues, e.g., more stress loads at the wing-root and large wing deformations [49]. Consequently, these deflections might lead to dynamic instabilities, which can be catastrophic in the case of large disturbances, for example, in the Helios accident, see Fig. 2.2 [50].

To address these challenges, load alleviation systems (LAS) play a crucial role in alleviating the loads and variations resulting from the aggressive maneuver or atmospheric turbulence [3]. Therefore, considering these control systems early during the conceptual phase can extend the lifetime of the airframe structure and enhance overall aircraft performance. This may also allow decreasing the overall costs related to fuel consumption by reducing the operation-empty-weight [43, 51]. In the first part of the thesis, we aim to design and implement MPC to alleviate the gust loads for passen-

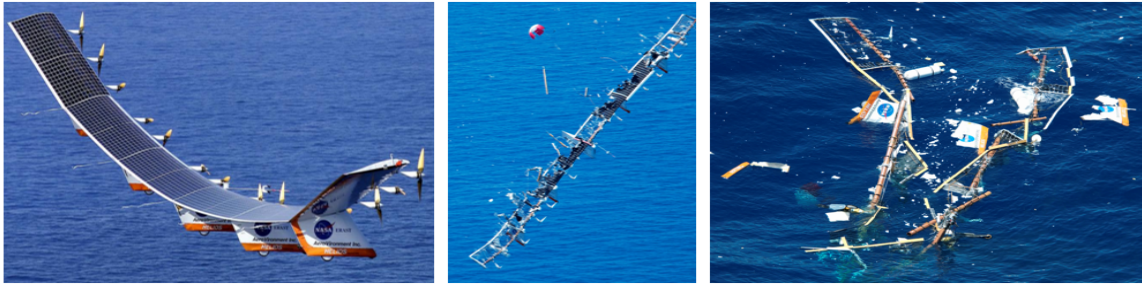


Figure 2.2: Flexible High-Altitude Aircraft: a Helios incident (source of all three photos: [50]).

ger aircraft, while the second part is related mainly to autonomous aircraft and their efficient operations in complex environments. The following section provides a brief introduction to autonomous vehicles.

2.2 Autonomous Aerial Vehicles

Recently, there has been growing interest in the role of autonomous vehicles in a wide range of applications [5–15]. Due to the increasing use of autonomous vehicles, many national laws and regulations have been developed to organize the civil UAVs usage [52]. For instance, they regulate flight safety to maintain the vehicle’s position and velocity within a set of safe margins. UAVs also need to have the ability to detect/sense and avoid airborne hazards, i.e., by updating the environment knowledge via sensor information [13] and reacting to unforeseen changes. As a result, great efforts have been invested in improving the efficiency of the autonomous vehicle, e.g., designing new configuration, advancing the guidance and control systems, and improving the sensing and actuation capabilities.

Definition 2.1 (*Autonomous Vehicles*): *powered vehicles with the necessary supporting capabilities, e.g., sensing, actuation, and computation, to achieve a specific task autonomously, i.e., without external human assistance [7].*

Definition 2.2 (*Autonomous Systems*): *a group of vehicles, controlled by a ground/air control station via communication architecture with support equipment [48].*

Because of their potential applications, a wide variety of autonomous vehicle are currently in use or being developed. Generally, the autonomous vehicles can be classified according to their structure, capability, environment, or operation, as ground, underwater, or aerial vehicles [6, 7]. Comparing to ground robots, UAVs have gained considerable attention owing to their features, such as expendable operation range and long flight mission duration [48]. UAVs have different design configurations, e.g., fixed, rotor, or flapping wings. However, extra factors need to be considered for UAVs’ operation as they fly in 3D environments and need to achieve safe operation despite

atmospheric turbulence. For this reason, this thesis aims to develop robust planning and control strategies for autonomous vehicles.

The following section will provide a breve introduction to the different control systems present in aerial vehicles.

2.3 Aircraft Guidance and Control Systems

The aircraft performance and operating costs depend on aircraft flight systems. The flight system spans from flight control to guidance and navigation systems, see Fig. 2.3. The aircraft control system is decomposed in a hierarchy architecture to handle the complexity. Such a decomposition decouples the fundamental planning and control challenges into sub-levels of tractable problems, allowing each to be pursued independently. This hierarchy-based design effectively reduces the computational burden, paving the way for real-time implementation of mission planning and control systems. It also allows reducing the hardware and software complexity.

The global decision-making layer operates at the slowest time scale, which provides global decisions concerning objectives based on the environment information, e.g., obstacle location, see Fig. 2.3. The reference planning layer generates offline/online the vehicle trajectory depending on the mission and the environment, i.e., static or dynamic. The reference governor modifies online the reference commands based on the current measurements considering simplified vehicle capabilities [53]. At the mid-level, the reference tracking layer determines more detailed state trajectories, e.g., velocity, angles, and rates, to achieve the designed reference. The vehicle is controlled via the local controller, which acts on a fast time scale to compute the input signal to the actuators, e.g., throttle, elevator, and drives.

In practice, common problems for the control of UAVs are unmodeled dynamics due to external disturbances, e.g., wind, uncertainty, and model errors. Neglecting the uncertainties and environmental changes can result in performance degradation, e.g., loss of feasibility or instability of the control systems. Therefore, a disturbance rejection module is often added to decrease the influence of external disturbance [34].

Typically, the planning and control layers take into account different vehicle models and different environment information, i.e., situation awareness. For instance, the obstacles information coming from the radar or camera sensors are mainly used in the high-level planner, while the low-level controller uses the local measurements for feedback, e.g., surface positions. The hierarchy, shown in Fig. 2.3, is often used in the aerospace and robotics applications. Extra modules and layers can be added to perform a specific function, e.g., fallback controller [54].

Practically, these layers and systems are highly interconnected. However, often the interaction between planning and control is not exploited directly. For example,

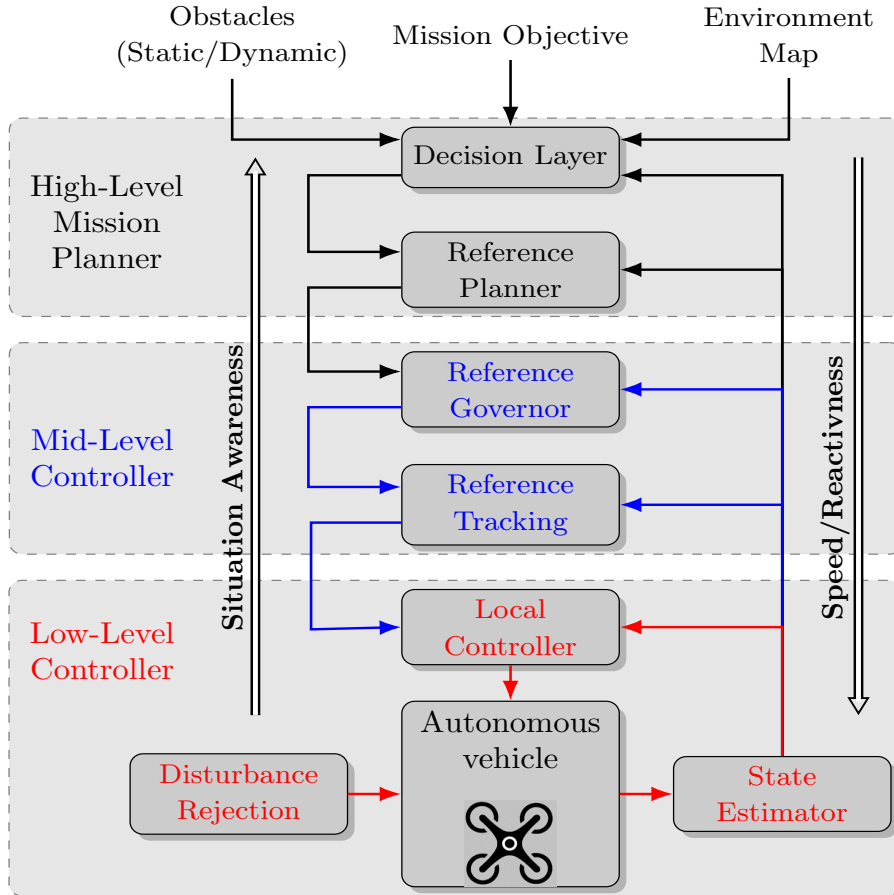


Figure 2.3: Aircraft planning and control systems: the low-level (red) utilizes a faster sampling frequency than the mid-level (blue) and the higher-level planner (black). The speed increases downwards, while the situation awareness and intelligence increase upward.

the planner sends information to the control layer, but the control layers do not send direct information to the planner. The separation between these systems leads to possibly unsafe overall behavior of the aircraft, especially for autonomous vehicles in dynamic environments. We propose an effective interface between different aircraft layers to overcome these limitations. For instance, increasing performance demands require path planning and control to be closely interwoven, taking into account all available information and vehicle dynamics and constraints. As another example, the load alleviation, e.g., disturbance rejection, can be integrated with the primary flight control system to ensure safe operation concerning structural integrity while improving passenger comfort. These objectives can be achieved by direct consideration of environmental disturbances and maneuver commands from the pilot.

In this work, we introduce MPC as a suitable approach to achieve all these objectives, e.g., load alleviation, planning, and low-level controller. We exploit MPC formulations on both the planning and control layers, which generate plans/controls

robust to the aircraft uncertainties and environmental disturbance. Note that MPC allows taking the vehicle dynamics and constraints into account to optimize the objective index. Furthermore, MPC has the ability to achieve a suitable interaction between the different control layers. In this hierarchy, the planning and control systems are often designed using different complexity levels of models of the aircraft dynamics and physical capabilities [32–35]. To ensure “consistency” between the levels and guarantee safety, we let the different systems communicate each layer using the measurement information from the navigation system.

The navigation system is responsible for determining the vehicle’s state (position, angles, velocity, and loads) and the surrounding environment data, e.g., obstacle position and wind speed. The navigation system includes onboard sensors, e.g., inertial measurement unit, and environment detection sensors, e.g., lidar, radar. For instance, accelerometers measure the mass displacement, which is proportional to experienced acceleration. Pressure sensors provide indications of the gust speed and the altitude and airspeed of the aircraft. The aircraft heading is usually indicated using a compass via measuring the direction of the magnetic field. The global positioning system is a satellite-based navigation system that provides a 3D position for outdoor applications [47]. In real-time applications, direct measurements of all actual states are usually not available, and measurement delays are present. As a result, it is necessary to combine the control scheme with a stable state estimation or filter. We proceed to describe each layer of the hierarchy.

2.4 Guidance and Motion Planning Systems

There is growing interest in increasing the autonomy level of many autonomous systems, e.g., aerospace [6–8], and automotive applications [55, 56]. Concerning them, one refers to autonomy, guidance, and motion planning as

Definition 2.3 (*Autonomy*): *the ability of being self-governing and includes sensing, perception, communication, planning, decision-making, and execution. It also includes task assignment and coordination in the case of multiple vehicles [6, 57].*

Enhancing vehicle autonomy requires improving the planning capabilities, making the system independent from a ground/air base station [6, 7].

Definition 2.4 (*Guidance system*): *exercises planning and decision-making functions to accomplish the assigned mission using the navigation data in diverse operating and environmental conditions [6, 7].*

Definition 2.5 (*Motion Planner*): *generates and updates reference commands and collision-free references to guide the vehicle through clutter environments to achieve its task [6, 7, 32, 33].*

Often, motion planning algorithms are divided into trajectory generation and path planning approaches.

Definition 2.6 (*Trajectory Generator*): *computes a time-dependent trajectory towards a desired state or configuration [6].*

Definition 2.7 (*Path Planner*): *determines a geometric path without timing information [6]; the time dependency is used as an additional degree of freedom.*

Most existing planning approaches do not take the vehicle dynamics and capabilities into account in the planning phase, which might lead to infeasible motions or poor performance. This issue is growing in importance, especially in the case of fast dynamics and in time-critical missions [8].

During the mission, the generated reference should also optimize performance criteria, e.g., time, energy, and uncovered area in search operations, see [32–35]. Generally, optimization-based planning techniques have become more common for real-time applications, facilitated by recent advancements in algorithms, computational power, and numerical implementations.

It is challenging to solve an optimization-based planning problem with many non-convex constraints, i.e., no-fly zones. One solution is reform to mixed-integer programming (MIP) [32–34, 58, 59]. MIPs allow formulating and solving the hybrid optimization problems using both discrete decisions and continuous variables, e.g., [33, 36, 57]. The discrete decisions “schedule” waypoints, which should be visited, thereby constructing the path with a large sampling time [32, 33]. In literature, MIP approaches have been implemented in many applications, e.g., planning formulations of multi-vehicles for the search and rescue mission [13, 47, 58]. MIP optimization-based formulations have been introduced with different operational features, e.g., minimum time problems, turn rate constraints [60], in case of 2D [13, 47, 58], and 3D environments [57, 60, 61].

MIPs are often “NP-hard problems”, i.e., the computation time increases exponentially with the number of integer variables. Practically, the variables number depends on the problem size, e.g., number of constraints, obstacles, and vehicles, and the area size [36, 62, 63]. Moreover, the non-convex configuration space, i.e., clutter environment, makes planning the entire mission prohibitively computationally expensive to find the global minimum by introducing large numbers of binary variables [6, 8, 36].

Moreover, it is challenging to represent the operating environment in many applications, where the vehicle’s knowledge may be incomplete and/or uncertain at the start of a mission. Typically, new information, e.g., obstacle locations, becomes available as the vehicles enter the operation region, i.e., the situational awareness change [6, 57]. For this reason, we propose to solve a reactive planning problem in a receding horizon scheme, i.e., to generate/update the path online based on the latest information

about the environment dynamics to achieve the avoidance requirements, see [32–35]. Furthermore, moving-horizon planning takes the vehicle dynamics, constraints, and environment descriptions into account, contrary to many existing planning approaches.

Autonomous vehicles are often exposed to uncertainties, e.g., wind disturbances and modeling error. Neglecting the uncertainties is detrimental, especially in cluttered and dynamic environments, and can result in performance degradation or complete failure. To overcome these challenges, robust planning approaches and control systems are needed to obtain safe and plausible references and control inputs to satisfy constraints and improve performance. In this work, the high-level planner uses the constraint tightening, i.e., encountered obstacles are enlarged by a safety bound to provide online a collision-free reference to the low-level controller. The safety bounds are provided and ensured by the low-level tube-based MPC [38], i.e., robust constraint satisfaction despite uncertainties by an adequate level of accuracy bounded in a tube.

The optimal solution of the planning problem, i.e., an energy-optimal path that avoids collisions with (dynamic) obstacles, is provided to the low-level controller.

2.5 Flight Control System

This section aims to clarify the motivation of the proposed control methodology. As mentioned above and illustrated in Fig. 2.3, the vehicle is controlled via the local controller, i.e., inner-loop, which acts on a fast time scale to compute the input signal to the actuators, e.g., throttle, elevator, and rudder.

Definition 2.8 (*Flight Control System*): *generates the control signal (forces and moments) utilizing the navigation data to satisfy specified control objectives.*

For example, the objective is to control the flight parameters to follow the desired reference, to improve the airplane stability, or to reject atmospheric disturbance [64]. To achieve these objectives, the flight control system often includes subsystems, e.g., stability augmentation system, controllability augmentation system, and autopilot.

Definition 2.9 (*Stability Augmentation System*): *regulates the vehicle state to equilibrium by suppressing any motion perturbations to stabilize the aircraft [64].*

This system eliminates high frequencies modes by providing artificial damping, e.g., pitch, roll and, yaw dampers [43].

Definition 2.10 (*Controllability Augmentation System*): *The system allows the aircraft to execute the desired maneuver easier and more precise, i.e., leads to superior pilot commands in all operation modes [43, 64].*

Definition 2.11 (*Autopilot*): *The system performs the autonomous functions, such as keeping constant altitude or attitude, or automatic take-off and landing modes [43, 64].*

These functions are necessary for the autonomous vehicles during all flight phases and for passenger aircraft to reduce the pilot workload [43, 47]. The flight control system mainly encompasses some actuating means, e.g., engine thrust and the control surfaces, in addition to redundant distribution systems to ensure aircraft safety.

In literature, the aircraft dynamics are usually decomposed into longitudinal (pitching and climbing) motions and lateral (rolling and heading) motions, including different modes, e.g., phugoid, short-period, and dutch-roll [47, 64]. Then classical controllers are designed independently for each mode using different control actuators [64]. For instance, the longitudinal controller has different feedback loops, e.g., airspeed hold using throttle and altitude hold via pitch attitude using an elevator [47, 64]. While the lateral-directional control is designed based on other loops, e.g., roll attitude, and course or sideslip hold via aileron and rudder [47, 64]. This approach is not valid for modern aircraft with complex dynamics, i.e., the control surfaces are nonlinearly coupled [65]. Moreover, the objective and performance of the control system depend on the aircraft dynamics and constraints, which might vary significantly according to the flight conditions. For this reason, it is recommended to design a unified control system with multi-objective and multi-input-multi-output, e.g., MPC approach.

According to the aircraft specifications, the flight control system is often designed to achieve some control requirements, e.g., response speed, relative stability, and system accuracy. For instance, the control system has to be designed to operate at low bandwidth to prevent exciting extra structural vibrations and flutter [65].

Furthermore, it is important that the controller does not generate too large commands that could damage the aircraft's structure. Therefore, the control system has to take the aircraft (state and inputs) constraints into account, which are due to the physical limitations of the structure, aerodynamics, or performance requirements. Traditionally, conventional control techniques usually decrease the possibility of constraint violations via choosing operating points far enough away from constraint boundaries, which might cause impaired performance, e.g., system instability. MPC allows the system to operate near to the constraint boundary; this improves operation efficiency.

Autonomous vehicles are often exposed to many types of uncertainties, e.g., wind disturbances. Neglecting the uncertainties is critical, especially in cluttered and dynamic environments, and can result in performance degradation or complete failure. For this reason, this work proposes a disturbance rejection strategy that combines optimal control with machine learning to improve the control performance and robustness by mitigating the effect of the disturbances. The next section describes the control methodology used to handle the gust disturbance for passenger aircraft.

2.6 Load Alleviation System

For ecological and economic aspects, efficient aircraft design tends to use a more flexible and lighter structure. The aircraft structure has to be strong and robust to withstand the maneuver and gust loads. One way to cope with this problem is to install a load alleviation system (LAS). LAS is mainly used to alleviate the structural loads and variations resulting from the aggressive maneuver or atmospheric turbulence [2, 3]. These control systems allow improving the aircraft fatigue life and performance, i.e., flying qualities [3, 43, 51]. Guaranteeing load reduction allows reducing the weight, i.e., by designing a lighter structure without compromising the safety standards. Consequently, weight reduction reduces the fuel expenditure and thereby the operating cost, which makes the aircraft more eco-efficient. For these reasons, these control schemes should be taken into account early in the design process of aircraft production. Hence, the interactions between LAS and the primary flight control system have to be taken into account to achieve the required performance and to ensure safe operating conditions. LAS is mainly divided into gust load alleviation (GLA) and maneuver load alleviation (MAL) systems.

Definition 2.12 (*Gust Load Alleviation*): *structural-mode suppression system is used to attenuate the structural vibrations and loads resulting from the gust turbulence.*

GLA allows enhancing aircraft stability and improving ride comfort, e.g., reducing cabin noise [1]. Therefore, GLA enhances the aircraft design, i.e., makes it more efficient and lighter, while still assuring compliance with the certification standards [66].

Definition 2.13 (*Maneuver Load Alleviation*): *system reduces the maneuver loads by redistributing the aerodynamic load via actuating control surfaces [43].*

Practically, the control performance depends on the actuator's surface effectiveness, which mainly depends on the actuator position, e.g., flaps, ailerons, elevators, and rudders. Using these actuation surfaces for different functions, e.g., primary control system and alleviating the gust loads, simultaneously might lead to undesired performance. For instance, using an elevator to alleviate the gust loads can increase the internal stresses in the horizontal tail root. So it is preferable to use redundant control effectors [2]. Another challenge in the aircraft with coupled flight-dynamics/aeroelastic response is the synthesis of appropriate control algorithms for stabilization, flutter suppression, and load alleviation.

For these reasons, we introduce MPC as a suitable control solution to the load alleviation problem for a flexible aircraft. Since the MPC framework has adaptation and reconfiguration capability, e.g., specifying certain actuators for load alleviation, this improves the functionality of the actuation system. Furthermore, MPC allows optimizing online the performance criterion, e.g., the load reduction at relevant stations,

e.g., wing root. Simultaneously, the actuator fatigue can also be addressed, e.g., by including the input rates in the performance criterion. The key advantages of MPC for this application are also the ability to consider multiple-input-multiple-output aircraft dynamics. Moreover, MPC allows systematically handling the aircraft (state and inputs) constraints, e.g., actuators limitations both in terms of operability range and response speed. MPC also has the ability to handle the model uncertainties and the external disturbance in a robust framework. Additionally, MPC allows utilizing the look-ahead information about the atmospheric turbulence, e.g., LIDAR, to improve prediction accuracy [67, 68]. This thesis focuses on the application aspects and the overall achievable performance by applying MPC-based GLA schemes.

The first step to design the proposed planning and control systems is to develop appropriate mathematical models of the underlying aircraft and autonomous vehicles.

2.7 Aircraft Modeling, Linearization, and Reduction

An aircraft is often considered as a rigid-body structure whose equations of motion (EOM) can be derived using physics principles, e.g., Newton's laws [47, 64]. Developing this mathematical model requires identifying the physical properties, e.g., geometry, mass, and aerodynamic coefficients. Traditionally, these coefficients are often obtained via system identification techniques using wind-tunnel or flight test data.

During flight, forces and moments acting on the aircraft's structure include distributed loads, e.g., aerodynamics and inertial, and concentrated loads, e.g., propulsive and gravitational. The aircraft loads are usually highly nonlinear functions of the aircraft geometry, flight conditions, and control surface deflections. As a result, the aircraft models usually include nonlinear differential equations for rigid-body kinematics and dynamics presenting the aerodynamic forces and moments [47, 64].

From the control perspective, the main challenge is how to approach the complex high-order models of the aircraft, often with time-varying parameters [69]. Therefore, different simplification and approximation methods are utilized to reduce the model complexity to facilitate control synthesis. For instance, the equations of aircraft motion are often linearized at different trim conditions, e.g., steady horizontal flight, steady climb flight, and steady horizontal turn [47, 64]. The flight trims are the steady-state conditions where the aircraft is in equilibrium, i.e., flies without translational or angular acceleration [47, 64]. To improve the computational efficiency, the model order can be reduced by projecting the full order models onto a subspace of the original physical space [4, 70]. This subspace encapsulates the fundamental dynamics of the aircraft with a small number of state [70].

For load alleviation in Chapter 4, the aircraft dynamics are defined by high-order aeroelastic models, which represent the flight-mechanics (rigid-body motions) and the

aeroelastic modes (elastic vibrations). In Chapter 5, the strategic-level control, i.e., planning layer, uses linear kinematic models representing the closed-loop behavior of the vehicle dynamics and the low-level controller, e.g., [32–34]. The kinematic modeling is based on the geometric information of the system, e.g., invoking the point-mass assumption, which decreases the computation time [71]. The main simplifications are to ignore the complex equations, which are force and moment dependent. In this setup, the inputs are commanded accelerations and the outputs are the airspeed and inertial position. In Chapter 6, the “tactical” levels, i.e., low-level controllers, are designed based on the rigid-body dynamics and ignoring the flexible effects, see [32, 34, 35].

2.8 Summary

This chapter presented an introduction to the basic of aircraft, guidance, and flight control systems. The following chapter provides an introduction to MPC principles that underlay the thesis contributions.

3 Model Predictive Control

We focus on exploring real-time implementation of MPC for flight control and guidance systems. This chapter outlines the basic MPC concepts that underlie all of the thesis contributions. Section 3.1 provides an introduction to MPC. Section 3.2 outlines the basic elements and mathematical formulation of MPC. Section 3.3 discusses application aspects and implementation challenges of MPC, such as robustness, stability, and real-time implementation. Finally, we present a selective overview of efficient solution methods, architectures, and the available numerical tools for MPC.

3.1 Overview of Model Predictive Control

MPC is based on a repetitive decision-making process, see Fig. 3.1. MPC scheme solves a finite-horizon optimization problem repeatedly, taking the system dynamics and constraints into account. There are many excellent reviews on the MPC main principles, theoretical aspects, and potential applications [17–30]. MPC was first utilized as a solution to tackle multivariate control problems for slow industrial processes [27]. In the 1970s and 1980s, MPC gained much interest in the process industry. Approaches such as dynamic matrix control [72], model predictive heuristic control [73], and generalized predictive control [74] were originated at that time. Following these applications, MPC strategies have been rapidly adopted due to the simplicity and the possibility to use dynamics models. Recently, MPC is used in a diverse range of applications, such as aerospace and robotics, driven by the emergence of fast numerical solution algorithms and embedded computing platforms.

3.1.1 Model Predictive Control Principle

In MPC for discrete-time systems, the future system behavior is predicted over a finite period, the so-called prediction horizon $T = N \cdot T_s$, using a system model and state measurements at the current time t_k , see Fig. 3.1. An optimal control sequence $\mathbf{u}^* = \{u(t_k), u(t_{k+1}), \dots, u(t_{k+N_c})\}$ is calculated solving an OCP, taking the system dynamics and (input and/or state) constraints into account. The solution of the optimization problem, if it exists, ensures the satisfaction of system constraints and boundary conditions and optimizes a cost functional, which represents the performance index involving objectives on both state and inputs, e.g., energy, tracking error, comfortable ride, and load reduction, see [17–27].

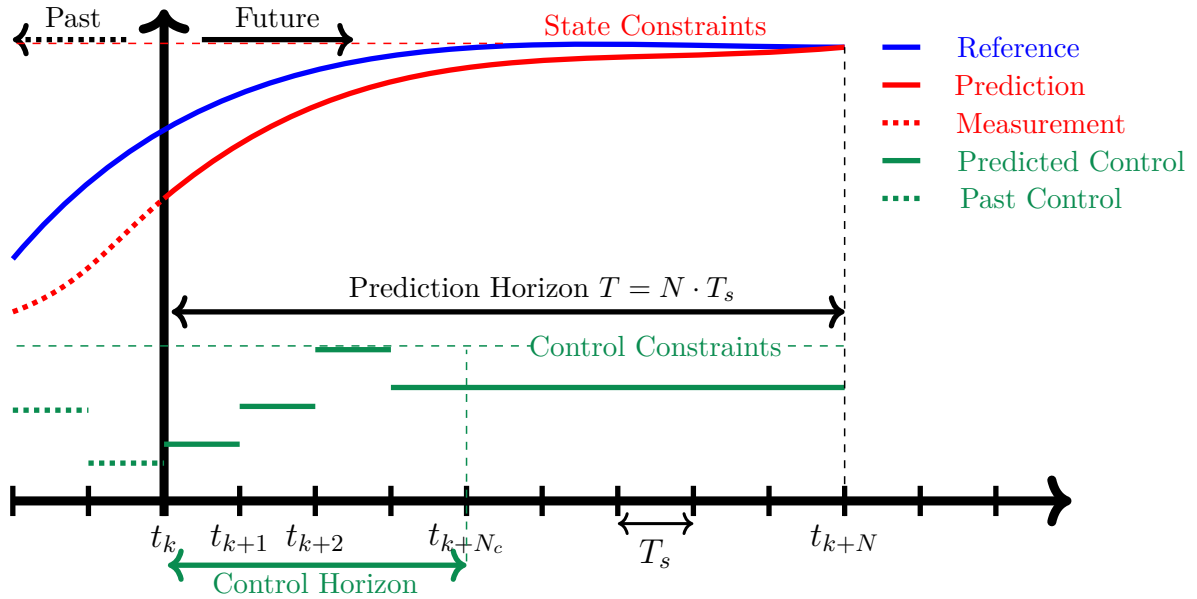
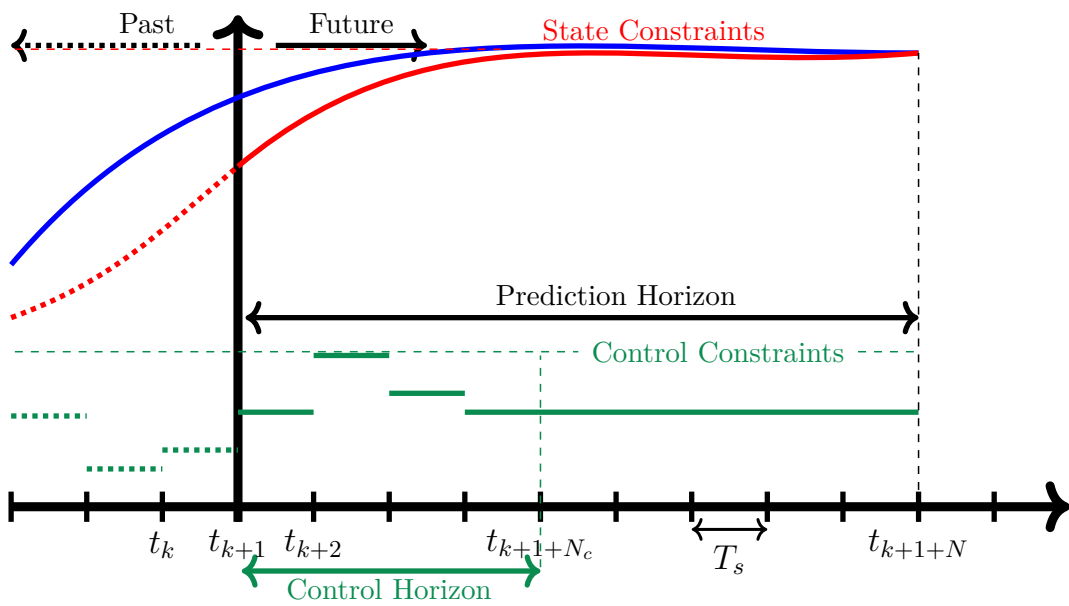
(a) At time t_k (b) At time t_{k+1}

Figure 3.1: Principle of model predictive control: The future system behavior is predicted over the horizon $T = N \cdot T_s$, based on a system model and state measurements at the current time t_k . The optimal control sequence $\mathbf{u}^* = \{u(t_k), u(t_{k+1}), \dots, u(t_{k+N_c})\}$ optimizes a cost functional, i.e. the performance index, which in general depends on both state and inputs.

The essential difference between MPC and standard optimal control is that MPC solves an OCP online at every sampling time [20]. Unmeasured disturbances or model inaccuracy lead to a mismatch between the predicted and actual system response. To counteract this deviation, MPC applies only the first term of the control sequence, i.e., $u(t_k)$, to the system until the next sample t_{k+1} . Then, the prediction and optimization are repeated, see Fig. 3.1b, in a moving-horizon fashion, using an updated state measurement [7, 17]. For this reason, MPC is also known as the *receding horizon control* (RHC). The main idea is summarized in the following Algorithm [7, 48]:

Algorithm 1: Basic MPC Algorithm

1. Measure/estimate the system state $x(t_k) = x_0$ at the current time t_k ;
 2. Solve a (finite-horizon) OCP by minimizing/maximizing a cost/reward function subject to the system model and constraints;
 3. Of the resulting sequence of optimal control inputs \mathbf{u}^* , only apply the first $u(t_k)$;
 4. Go to step 1, repeat the optimization.
-

3.1.2 Model Predictive Control Potential and Possibilities

MPC has been attracting increasing attention in the control community due to the following advantages [7, 17, 48, 75]:

- *Use of previews and predictions:* MPC optimizes the predicted system behavior over the prediction horizon to yield the optimal control action. This predictive nature allows utilizing available preview information, such as predicted gust disturbances or moving obstacles.
- *Systematic treatment of constraints:* This allows considering constraints of maneuverability, obstacle avoidance, load, and actuator limitations.
- *Optimized performance:* MPC optimizes online the performance criterion, e.g., energy, tracking error, comfort ride, or load reduction.
- *Wide spectrum of models:* MPC can handle multi-input-multi-output, linear, nonlinear, or hybrid dynamics model.
- *Simple reconfigurability:* MPC can reconfigure the controller by merely updating the system model, cost function, and/or using different constraints, e.g., specifying certain actuators for load alleviation to improve the functionality of the actuation system.

- *Hierarchical applicability*: MPC can be applied at different levels, e.g., strategic (planning), supervisory controller, or tactical (executorial) controller. MPC provides a suitable interaction between the different control layers considering different objectives.
- *Possibilities to handle uncertainty*: Robust and stochastic MPC approaches ensure a priori absolute or probabilistic guarantees despite bounded (set membership) or probabilistic (stochastic) types of uncertainty.
- *Consideration of time delays*: MPC can handle delays e.g., due to communication, sensors, and actuators.

3.2 Basic Constituents and Mathematical Formulation

The mathematical formulation of MPC consists of a set of OCP elements [20]. Often, MPC is formulated in a discrete-time setting, i.e. generates control actions at discrete time intervals $\{t_k, t_{k+1}, t_{k+2}, \dots, t_{k+N}\}$.

Definition 3.1 *Sampling time*: Time interval between two time-steps, $T_s = t_{k+1} - t_k$.

Definition 3.2 *Prediction horizon*: Time over which the future system behavior is predicted $T = N \cdot T_s$.

Definition 3.3 *Control horizon*: N_c defines the number of actuation signals to be determined in the future.

The prediction horizon and sampling time are typically connected. Often, a faster sampling rate is desired to handle high-frequency perturbations for disturbance rejection applications. However, it typically increases the computation-time as it requires more prediction steps for a given prediction duration. Often, increasing the prediction horizon might be necessary to achieve the performance demands, e.g., sufficient foresight to avoid a collision in the planning formulation [32, 33].

One way to reduce the computational burden is to fix the control input beyond the control horizon, $N_c \leq N$, as this reduces the number of decision variables [17].

3.2.1 System Model and Prediction

One of the core aspects of MPC, model-based control, is to utilize the system model to predict future behavior starting from an initial measured/estimated state, see Fig. 3.1. According to these predictions, the optimal control sequence is determined, so the optimization program is constrained by the model while still fulfill the constraints.

For choosing an appropriate model, there is a trade-off between prediction accuracy and computation complexity. For instance, using more accurate models results in more computation time to solve the optimization problem, and it affects the control performance and stability [17]. A reasonable rule is to choose the simplest possible model that is accurate enough to make reasonable predictions. In this work, different MPC approaches have been proposed based on distinct types of models:

1. We use nonlinear discrete-time models of the form:

$$x(k + j + 1|k) = f(x(k + j|k), u(k + j|k)), \quad (3.1a)$$

$$y(k + j|k) = g(x(k + j|k), u(k + j|k)), \quad (3.1b)$$

where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, $y \in \mathbb{R}^{n_y}$ represent the controlled variables (state), the manipulated variables (control inputs) and outputs vectors with appropriate dimensions n_x, n_u, n_y respectively. The argument $(k + j|k)$ denotes the j -th future time step prediction made at the time step k . The map $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ represents the system dynamics, which defines the successor state in terms of the current state and input. The function $g(\cdot, \cdot) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_y}$ represents the system output as a function of the state and inputs.

2. *Linear models* are often used to reduce the computation time by linearizing the nonlinear one (3.1), either around the operating point or around a trajectory.

$$x(k + j + 1|k) = A(k + j|k)x(k + j|k) + B(k + j|k)u(k + j|k), \quad (3.2a)$$

$$y(k + j|k) = C(k + j|k)x(k + j|k) + D(k + j|k)u(k + j|k). \quad (3.2b)$$

Here, the system matrix $A \in \mathbb{R}^{n_x \times n_x}$, the input matrix $B \in \mathbb{R}^{n_x \times n_u}$, and the output matrices $C \in \mathbb{R}^{n_y \times n_x}$, $D \in \mathbb{R}^{n_y \times n_u}$ could be time-variant or invariant matrices, which leads to the different models, e.g., linear time-invariant, linear parameter-varying, or linear time-varying.

3.2.2 State and Input Constraints

MPC allows taking constraints explicitly into account. Those might represent the physical limitations of the state, e.g., maximum velocity and accelerations, stall angle of attack, or desired operation modes, and inputs, e.g., actuators restrictions, fuel, and energy [17]. Furthermore, safety constraints, e.g., obstacles, can be formulated as mixed state-control constraints [32–34]. Theoretically, constraints can also represent some controller design objectives, e.g., used to achieve recursive feasibility [33, 59]. Generally, the constraints can be classified as:

Stage Constraints represent the state and input constraints over the horizon.

$$x(k + j|k) \in \mathbb{X}(k + j|k) \subset \mathbb{R}^{n_x}, \quad \forall j \in \{0, \dots, N - 1\}, \quad (3.3a)$$

$$u(k + j|k) \in \mathbb{U}(k + j|k) \subset \mathbb{R}^{n_u}, \quad \forall j \in \{0, \dots, N - 1\}. \quad (3.3b)$$

The sets \mathbb{X} and \mathbb{U} are the state and input constraints, which could be time-varying due to moving obstacles [32] or constraint tightening [33].

Terminal Constraints represent constraints, which must be fulfilled at the end of the prediction horizon, e.g. $x(N) = x_T \in \mathbb{R}^n$. This constraint is in some cases relaxed by introducing a so-called terminal region constraint, e.g. $x(N) \in \mathbb{X}_T \subset \mathbb{R}^n$ [20].

3.2.3 Objective Function

According to the control goals, the objective function contains the performance specifications to be optimized. Typically, the finite-horizon cost functional $J(\cdot)$

$$J(x(k), u(k)) = \sum_{j=1}^{N-1} \ell(x(k + j|k), u(k + j|k)) + E(x(k + N|k)), \quad (3.4)$$

is decomposed into a stage and a terminal cost.

Stage Cost Function $\ell(x(\cdot), u(\cdot)) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}_0^+$ is typically formulated as a function of both the state x and control variables u to achieve the desired performance. For instance, it could penalize the deviation from the reference (desired state and controls), which could include the steady-state in the case of stabilization. For technical reasons, it is often assumed that the stage cost is strictly greater than zero everywhere (except at origin).

Terminal Cost Function $E(x(\cdot)) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}_0^+$ is employed to specify the state's contribution beyond the horizon. Terminal penalty (cost-to-go) is often incorporated with a short horizon to reduce the computational burden compared to the long horizon.

The overall mathematical formulation for MPC is summarized in a discrete-form as:

$$\min_{x, u} \sum_{j=1}^{N-1} \ell(u(k + j|k), x(k + j|k)) + E(x(k + N|k)), \quad (3.5a)$$

$$\text{s.t. } x(k|k) = x_0, \quad (3.5b)$$

$$x(k + j + 1|k) = f(x(k + j|k), u(k + j|k)), \quad j \in \{0, \dots, N - 1\}, \quad (3.5c)$$

$$y(k + j|k) = g(x(k + j|k), u(k + j|k)), \quad j \in \{0, \dots, N - 1\}, \quad (3.5d)$$

$$x(k + j|k) \in \mathbb{X}(k + j|k), \quad j \in \{0, \dots, N - 1\}, \quad (3.5e)$$

$$u(k + j|k) \in \mathbb{U}(k + j|k), \quad j \in \{0, \dots, N - 1\}, \quad (3.5f)$$

$$x(k + N|k) \in \mathbb{X}_T, \quad (3.5g)$$

where, x_0 represents the initial condition of the system.

Typical stability and feasibility properties can be guaranteed by stability conditions on the cost function $E(\cdot)$, and a positive invariance condition on the terminal set \mathbb{X}_T [21]. The MPC performance depends on properly tuning the weight parameters in the cost function $J(\cdot)$.

3.3 Theoretical Aspects

The thesis's focus is to design and real-time implement MPC schemes for aircraft and autonomous vehicles. Thereby, this section aims to outline some theoretical aspects of MPC implementation, e.g., stability and handling uncertainties.

3.3.1 Stability

Solving online an infinite-horizon OCP is often (computationally) not possible. Using a finite-horizon in MPC in general results in instability and feasibility problems [20]. Many approaches have been proposed to achieve stability in MPC, e.g., [20, 29, 76–78], e.g. by careful selection of the terminal cost and set as well as the weighting matrices.

For instance, stability can be achieved by *zero terminal constraints* forcing the state to be zero $x(t_k + N) = 0$ at the end of the prediction horizon at every time step. However, satisfying this constraint is in general difficult, if not impossible, for a short prediction horizon [20]. To avoid this problem, the terminal region constraint and/or the terminal penalty have been proposed which determines an upper bound of the infinite-horizon cost. These ensure that the value function decreases as the horizon recedes in time [20].

In the *Lyapunov function approach* [79], the terminal cost is related to a (known) *Lyapunov-function* to enforce MPC stability.

Dual-mode MPC was developed to guarantee the system stability by forcing the system state to end in a terminal region where a controller is switched to which enforces local stability. The system state will remain inside this terminal region using an appropriate local linear controller. A terminal penalty is used to ensure the convergence to the terminal region, i.e., decreasing the value function [80].

To enforce system stability, *contraction constraints* [81] can be added to explicitly enforce both the actual and predicted state to contract every recalculation instant [81]. However, this approach does not guarantee the recursive-feasibility without other assumptions on the system.

3.3.2 Model Predictive Control under Uncertainties

As MPC is a model-based controller, its performance depends on the model quality, i.e., the mismatch between the real system and the prediction model [17]. In practical

applications, model quality may be affected by external disturbances, inherent discrepancies, or measurement noise [17]. Other sources that lead to prediction error can be parameter identification errors, changes in the operating conditions, and modeling assumptions, e.g., neglecting high-order dynamics [17]. Additionally, often the surrounding environments are highly uncertain, or only limited information is available.

In principle, nominal MPC has inherent robustness properties to cope with the system uncertainties [20], but under certain conditions [82]. However, in the presence of uncertainty, MPC may perform poorly or drive the system into an infeasible region [83].

To guarantee feasibility, robust MPC (RMPC) approaches (e.g., [38, 83–96]) have been proposed to systematically “robustify” the original MPC formulation by modifying the objective function and constraints. This leads to input sequence or policies, which satisfy constraints despite the uncertainty [83]. The main issues of most approaches are conservativeness or the computational time to achieve the desired robustness properties [17]. The appropriate choice depends on various factors, e.g., optimization run-time and achievable performance [83].

For constraints satisfaction, RMPC approaches use different mathematical descriptions of uncertainty. For instance, parameter uncertainty can be represented by

$$x(k+1) = f(x(k), u(k), \mathcal{P}(k)), \quad \mathcal{P} \in \mathbb{P},$$

where the uncertain parameter \mathcal{P} is bounded in a compact set \mathbb{P} [21]. Additive state disturbance is commonly represented by

$$x(k+1) = f(x(k), u(k)) + \omega(k), \quad \omega(k) \in \mathbb{W},$$

where the disturbance ω is assumed to be bounded in a compact set \mathbb{W} , which contains the origin [17]. Many results in RMPC [38, 86–88] have focused on handling bounded uncertainties [17]. Other RMPC approaches, e.g., in [89, 90], consider state-dependent (bounded) uncertainties. Common RMPC approaches can be classified as follows.

Min-Max RMPC approaches are based on minimizing the cost function concerning the worst case of the admissible disturbance sequence [86]. This ensures robust feasibility by guaranteeing constraint satisfaction for all disturbance realizations [17]. However, this approach might be excessively conservative (i.e., large infeasible regions) and computationally intense because it attempts to optimize over every possible disturbance sequence. Often, Min-Max RMPC is not suitable for online optimization [86, 90].

Affine Feedback Policy based MPC restricts the anticipated growth of the uncertainty to overcome the conservativeness limitation of the open-loop prediction [97]. This can be achieved by inserting an affine feedback policy [86]. Several parametrizations of affine approaches have emerged, and the feedback term can be optimized online or offline [83]. However, the complexity of such approaches grows exponentially with the problem size, limiting their online implementations.

Linear matrix inequalities are an alternative approach of affine policies to frame the robust problem as computationally tractable [85, 98]. As in the min-max formulations, this approach minimizes the upper bound on the cost function (worst-case) over all possible disturbance sequences using state feedback policies. Stability and robustness can be guaranteed using convex optimization techniques [99].

Constraint Tightening approaches solve the nominal open-loop optimization problem and achieve robustness by systematically shrinking the system constraints. This leads to a guarantee to satisfy the original constraints despite any possible realization of the future uncertainties (unknown but bounded), which can be rejected using a future feedback correction, e.g., in [57, 83]. Offline procedures have been proposed [57] to compute the feedback policy to determine the tightening amount. This feedback improves the performance and satisfies various design criteria, e.g., tolerating much stronger disturbances. The bounded uncertainty effects in the case of a linear system can be represented by disturbance-invariant sets, which is used to shrink the system constraints, by the Pontryagin difference operator [100]. Many constraint tightening approaches have been presented, considering varying constraints [59], constant or time-varying state feedback [59, 88], and time-varying disturbance feedback [57, 100]. An essential advantage of this approach is less computationally intensive, suitable for real-time applications [17]. Constraint tightening methods do not significantly increase the online optimization complexity, i.e., the decision space and the number of the constraints. However, they might be conservative (i.e., smaller feasibility regions) introduced to account for the disturbances along the prediction horizon [17].

Tube RMPC allows handling directly bounded uncertainties, e.g., [38, 91, 92, 97]. The core of *tube* RMPC is to replace individual trajectories with an envelope that includes all possible trajectories [38]. This can be achieved by replacing the predicted state's trajectories with robust forward invariant tubes [87]. The initial state condition is also relaxed such that the predicted state's trajectory defines a "tube center", which does not necessarily need to originate at the current state. In this context, tube MPC involves the control parametrization into a feed-forward term and a feedback term that regulates the true state of the system to the predicted tube center.

To ensure a feasible solution, tube-based MPC modifies the system constraints, closely related to constraint tightening methods. In general, constraint tightening and tube RMPC approaches can handle bounded uncertainty without requiring a disturbance estimator, enabling more aggressive performance [38, 100]. In linear systems, constraint tightening with a fixed state feedback gain is less conservative than tube MPC. However, both approaches increase conservativeness.

Stochastic and Probabilistic RMPC approaches have been proposed to handle the uncertainties, which cannot be represented by sets [96]. In conventional robust formulations, the uncertainty bounds are assumed to be known a priori or be estimated online, which are used to modify the robustness bounds [88]. The resulting problems

are typically formulated as stochastic optimization involving probabilities [93–95].

Chance constraints have been proposed to provide a systematic way for a probabilistic interpretation of constraints violation, e.g., [101]. In cases of uncertainties with an arbitrary distribution, sample-based methods have been proposed to approximate the state distributions with particles, which leads to a deterministic optimization problem.

3.4 Implementation Aspects

In critical applications, it is necessary to investigate the reliability of the underlying control system from both the control and implementation perspectives. Thereby, this section aims to outline some implementation aspects, e.g., numerical solvers and optimization algorithms.

It is necessary to verify and validate the reliability of the MPC algorithms, especially in cases of safety-critical applications, e.g., load alleviation and planning, which endangers human safety. From a control perspective, the MPC algorithm should satisfy all control requirements based on some theoretic metrics, e.g., guarantees of optimality, invariance, stability, convergence, and verifiability [17]. From a practical perspective, MPC algorithms can be verified using software-in-the-loop tests, demonstration tests [102], Monte-Carlo simulations [103], or software verification tools [104].

Practically, MPC applications are mainly restricted by the onboard capabilities, e.g., computational power, software architectures, communication capabilities, and the available memory to store the computation data. For online computations, the required memory depends on the prediction horizon, the system dimension (i.e., number of inputs, state, outputs, and constraints), the required bytes [105]. This also depends on how the program has been written, i.e., using specific structures, e.g., sparsity [20]. This variety of aspects has led us to approach the MPC implementation in a holistic manner to achieve real-time implementation.

Implementing NMPC requires solving a nonlinear (non-convex) OCP, which increases the problem complexity and leads to reliability and computational/numerical issues, e.g., a guarantee of global optimum [17]. This limits the generic implementations of NMPC in real-time applications. To reduce the computational burden, many MPC approaches are based on simplification and approximation of the system dynamics or the calculated control law.

Using linear models and constraints with quadratic cost functions leads to a quadratic programming (QP) (i.e., convex optimization) appropriate for the real-time MPC implementation using efficient tools, e.g., [106, 107].

3.4.1 Available Solution Methods

Different methods have been proposed to solve OCP in general and MPC problems in particular, for more details, the reader is referred to [24, 108] for an introduction on numerical methods, and [17, 109] for embedded optimal control. Many techniques have been developed to enable real-time MPC implementation by solving the optimization problem by implicit or explicit methods or use a combination of both methods [17].

Implicit methods solve online the underlying OCP for the current state at every time instance. This solution can require high computational power, which could be challenging on embedded platforms [17]. Therefore, *explicit methods* have been proposed to solve the underlying OCP offline to avoid the online optimization [107, 110].

This reduces online calculation to look-up table evaluations of the optimal control function for the current state. This reduces the online computational complexity [17, 110]. However, the achievable performance depends on the size and resolution of these look-up tables, which depends on the problem complexity, i.e., horizon, number of state, inputs, and constraints. Therefore, there is a trade-off between optimality and memory, especially for high-dimensions systems, e.g., flexible aircraft. For these reasons, researchers proposed combining the explicit and implicit methods, i.e., pre-compute part of the optimization offline, to decrease the online computation time [17].

For the iterative solution, so-called *warm start* approaches use the previous solution of the MPC problem for the next control action to reduce the required time-to-solve optimization problems.

Computations requirements have been reduced with the help of tailored sequential and simultaneous methods [106, 111]. For example, sequential quadratic programming (SQP) methods solve the MPC problem by repeatedly approximating the KKT system that is compatible with QP solver [55]. For NMPC, QP solvers can use Hessian approximation approaches, e.g., quasi-Newton methods [108], or generalized Gauss-Newton Hessian approximations [55]. Moreover, NMPC problems can also be solved using iterative solutions of QP-based subproblems [17], i.e., the solution approaches can be roughly divided along the lines of interior-point and SQP methods [55].

Real-time iteration methods have been developed for NMPC applications to relax the optimality condition [55]. *Multi-level iteration* methods [112] subdivide the derivative computations into two or more levels (increasing complexity) to be executed in parallel with different sampling rates [55]. The main idea is to update some parameters while keeping others fixed according to two update strategies, one to improve the feasibility, and another improves optimality [55]. In the same concept, *mixed-level iteration* methods [113] assign a different level of parameter update to different segments of the NMPC prediction horizon. In addition to the above solution methodologies, many numerical algorithms have been developed to reduce the required computational time for solving the underlying OCP, see the following section.

3.4.2 Efficient Software Solution Tools

The verification and real-time applicability of the MPC algorithms mainly depend on the reliability of the optimization algorithm [17]. Most numerical solvers are potential candidates to be tailored specifically for MPC implementation [17]. We provide a short review of the available numerical solvers with a particular focus on the tools used in this thesis, see Fig. 3.2.

The real-time MPC implementation of the GLA system, in Chapter 4, uses the code generation software μ AO-MPC [114]. μ AO-MPC is a code generation tool, which provides a highly portable library-free C-code. This allows implementing MPC on embedded applications with low computational requirements [114]. The optimization problem is solved online by a QP solver based on an augmented Lagrangian method with Nesterov's gradient method. The constraint equations, gradient vector, and Hessian matrix of the optimization problem are calculated in a condensed form.

The MIP planning problems, in Chapter 5, are solved using YALMIP [115], which provides solutions of convex and non-convex mixed-integer optimization problems [115]. To find a solution to these MIP problems, it exploits a powerful optimization software, e.g., Gurobi optimizer [116]. Gurobi can efficiently solve LP, QP, MILP, and MIQP with a reasonable size, i.e., the number of binary variables. There are different methods to solve the MIP, e.g., branch-and-bound algorithm, cutting-plane algorithm, and branch-and-cut algorithm [117]. This thesis uses the branch-and-bound algorithm [118] to solve MILP problems efficiently in reasonable computation times (on the order of milliseconds). This method allows using integer relaxation to accelerate the solution process by splitting the problem into sub-problems [117].

Tube-based MPC, in Chapter 6, is implemented using the multiparametric programming MPT toolbox [119].

We tackled the NMPC path following problem, in [32, 35], by the ACADO Toolkit [120]. ACADO provides a Matlab interface to generate C++ code for NMPC. ACADO implements the real-time iteration scheme with different algorithms to solve OCP. It incorporates efficient numerical integrators in auto-generated C-code [120, 121]. The extension of ACADO, called ACADOS [122], provides a computationally efficient modular framework for direct optimal control methods, which is applicable for embedded applications [122].

CasADi [123], another open-source tool, provides a framework for numerical optimization and automatic computing the Jacobian matrices. CasADi uses C-code generation and interfaces codes, e.g., SUNDIALS [124], and IPOPT [125]. IPOPT is an interior point nonlinear programming solver used in conjunction with MATLAB fmincon solvers and the BLOM software package [126]. BLOM is also used for modeling nonlinear systems for NMPC applications. Many software packages are based on QP solver, e.g., qpOASES [127] and HPIPM [128]. For implicit MPC design, qpOASES is

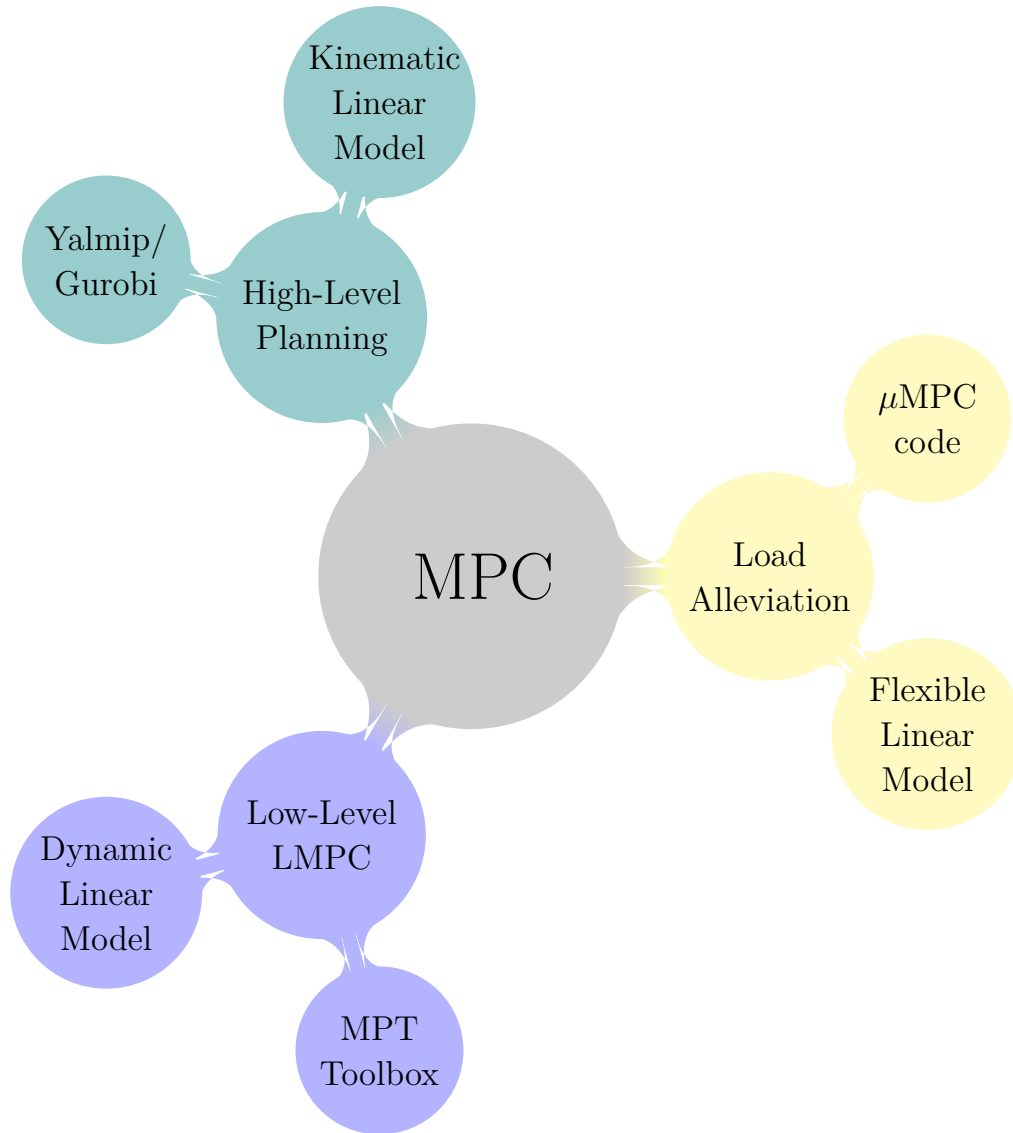


Figure 3.2: Software tools and approaches used in this thesis.

the default QP solver used in the ACADO Toolkit, and used to implement an active set strategy online in the C++ code.

For small to medium scale problems, HPIPM [128] is an efficient interior-point QP solver based on BLASFEO [129], which is a set of linear algebra routines for embedded optimization applications. Another QP-based software is the MATLAB MPC toolbox [130], which provides C-code generation to calculate explicit solutions. Other MATLAB-based software packages for MPC with linear systems are, e.g., *fast mpc* [107], *jMPC Toolbox* [131], *FiOrdOs* [132] and *PnPMPC toolbox* [133].

Different tools are available to solve convex optimization problems, e.g., *CVX* [134], *ECOS* [135], *FORCES* [136], and *FiOrdOs* [132], which are particularly suitable to generate C code for implementing MPC on embedded platforms. For real-time NMPC applications, other software packages, e.g., *ICLOCS* [137], and *PROPT* [138], *MUSCOD-*

II [139], and OptCon [140] are based on the multiple-shooting optimization approach.

These advanced solvers use the recent developments in the computation technology, e.g., electronic processor and memory, for real-time implementation in embedded platforms. Nevertheless, with the increasing autonomy and performance demands, MPC applications become more challenging considering more complex dynamics and objective function, or long prediction horizon. So there still exist significant challenges, which need more advanced solutions to have emerged in the future.

3.5 Summary

This chapter outlined MPC formulations and key ingredients required in the following chapters. First, we introduced the MPC, the underlying tools used in this thesis. Afterward, we outlined implementation challenges, efficient solution methodologies, and numerical tools. In the next chapters, we will discuss how to design and implement an MPC scheme to solve different problems, e.g., load alleviation, planning, and control.

4 Model Predictive Control for Gust Load Alleviation

Ecological and economic aspects are a driver in the development of flexible aircraft, which are exposed to critical loads resulting from, e.g., gust. This motivates us, in this chapter, to design and implement an MPC scheme to attenuate atmospheric loads. First, we outline the load alleviation problem in Section 1 and briefly review the common GLA approaches. Section 2 presents the control setup of MPC for the GLA design. Section 3 illustrates numerical simulation scenarios, and then Section 4 shows the results of wind-tunnel experiments, which demonstrate the real-time capability of the proposed methodology to reduce the gust loads. Finally, Section 5 provides discussion and concluding remarks.

4.1 Introduction and Motivations

The aviation industry is continuously driven to decrease the ecological footprint and improve economics. More efficient “greener” aircraft need to be developed to reduce fuel consumption while enhancing endurance and passenger comfort [4, 16]. This requires enhancing the aerodynamics and propulsion efficiency as well as the advanced control systems. Many research activities focus on reducing the structure weight and designing aircraft with high-aspect-ratio wings to reduce the aerodynamic drag and fuel consumption [141]. However, this increases the structural flexibility, i.e., aircraft becomes more sensitive to atmospheric disturbances. Due to flexibility, aerodynamic loads changes leading to further elastic deformation. This influences the rigid-body aerodynamic and might decrease flying performance, e.g., passenger comfort [1]. For instance, lighter wings are more exposed to unstable performance, i.e., flutter and high-frequency oscillation.

Such effects might lead to critical stresses on the structure, e.g., wing-root, causing fatigue problems, i.e., reduce the aircraft lifetime.

This can be compensated using active control techniques that allow increasing the lifetime of the aircraft [51]. Load reduction can allow reducing the weight, i.e., design lighter structure, without compromising the safety standards. Consequently, this reduces the fuel expenditure and thereby the operating cost, making the aircraft more eco-efficient. Therefore, these active control systems can be essential parts of the developing process of any new aircraft design [142].

The importance of GLA system mandates to be included early in the design process to achieve the required performance and ensure safe operating conditions. This is challenged by the often present classical control strategies, the primary flight control system can overrule the GLA system [67].

The following section gives a brief review of the existing GLA approaches.

4.1.1 Load Alleviation State of the Art

Historically, load alleviation systems have been implemented to alleviate the structural loads resulting from atmospheric turbulence [3, 142]. Many aircraft in-service are equipped with load alleviation systems to mitigate the effects of atmospheric disturbances. This leads to increase wing fatigue life and/or enhance the flight quality [3]. For instance, Airbus A320 aircraft have been equipped with a load alleviation function, increasing the maximum take-off weight by 1.3% [3, 141].

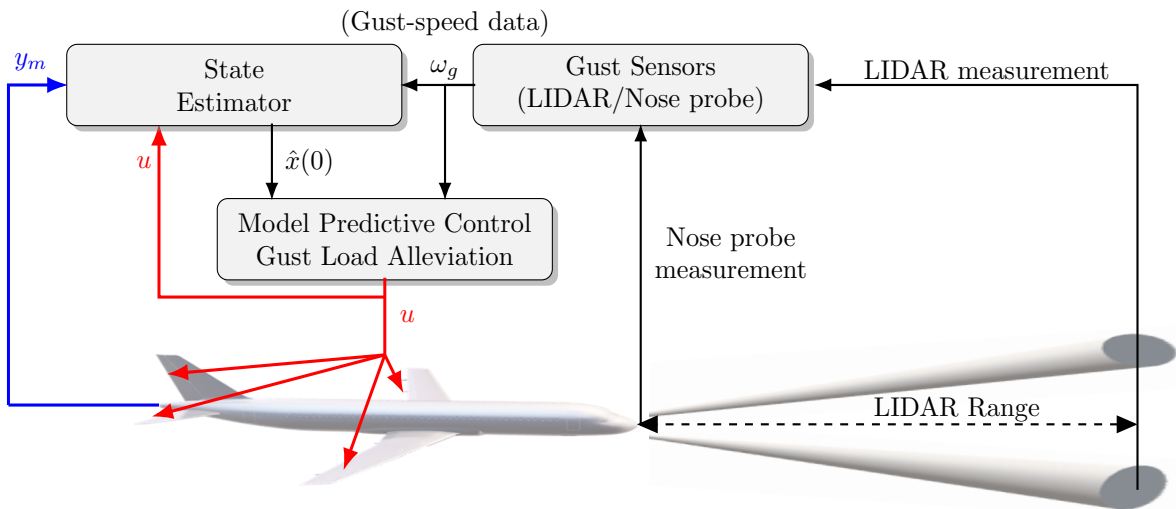


Figure 4.1: Proposed scheme of GLA/MPC utilizing the gust measurement via a nose probe and LIDAR sensor. The main idea is to design an MPC scheme to minimize the relevant loads via the actuation signals u using the available onboard measurements y_m , and the gust information ω_g .

As full state measurements are challenging for aeroelastic applications, static output feedback controllers have been proposed, e.g., [143]. This feedback GLA system uses the available sensor measurements to control specific surfaces, e.g., the load factors to the longitudinal controls (elevators or canards), or sideslip angle to the lateral controls (rudders). The Boeing 787 aircraft incorporates static-air data to counteract the turbulence using ailerons, spoilers, and elevons [3]. These controllers are simple but highly dependent on sensor selection and positioning. Recently, active aeroelastic

control has been proposed to collocate both sensors and actuators at the same location, e.g., acceleration feedback and piezoelectric patches on the wing [144].

To improve the GLA performance, feed-forward controllers have been implemented using gust measurements, e.g., from an onboard angle-of-attack (AoA) probe, or LIDAR [145], see Fig 4.1. These sensors provide lead-time to anticipate future loads to control the actuator surfaces [146]. Therefore, early gust measurements allow improving control performance, i.e., to reduce the effect of the incoming turbulence and decrease the structural loads [67, 145]. In this context, a combination of feed-forward and feedback approaches have been proposed to enhance the control performance [146]. In this setup, the feed-forward controller is used to alleviate the disturbance via the control surfaces directly. The feedback controller uses small additional deflection of the control surfaces to increase the damping of elastic modes.

In aerospace applications, linear controllers, e.g., proportional integral derivative, have been used, e.g., for GLA systems [144] thanks to their simplicity and intrinsic robustness. Often, aircraft nonlinear dynamics can be represented as a set of linear models for different flight conditions, each associated with a linear control law [70]. Hence, a suitable scheduling/adaptive scheme can be used to maintain an acceptable performance in the transition between the operating conditions [147].

Practically, flexible aircraft are associated with different uncertainties, e.g., dynamics approximations and estimation errors [1]. Neglecting the uncertainties can result in performance degradation or instability of the control systems. So robust controllers, e.g., \mathcal{H}_2 and \mathcal{H}_∞ [148, 149], have been proposed to determine a stabilizing solution under all uncertainties less than worst-case uncertainties.

Recently, there is an increasing interest in optimization-based GLA approaches, e.g., linear quadratic control (LQR) and linear quadratic Gaussian (LQG). They are based on minimizing a quadratic objective function to alleviate the structural loads and to control both rigid-body and flexible modes [148, 149]. However, these approaches do not allow considering constraints. To do so, MPC approaches solve an OCP online over a finite prediction horizon, taking into account the system dynamics and constraints [1, 67, 78, 145, 150, 151]. A receding horizon approach was recently proposed to account for stochastic gust disturbances and maneuvers with limited preview [2].

In this work, we propose to use MPC as a suitable solution to the load alleviation problem. MPC allows optimizing online the performance criterion, e.g., reducing acceleration and loads, at relevant stations, e.g., wing-root, and at other stations on the aircraft structure. Simultaneously, penalization on the control inputs can be imposed to prevent actuator fatigue [67]. Furthermore, MPC also has the ability to consider multiple-input-multiple-output systems, e.g., flexible aircraft dynamics [150]. Moreover, MPC allows utilizing the preview information about the atmospheric turbulence, see Fig. 4.1, to improve prediction accuracy and control performance [67, 68, 145]. Furthermore, MPC has adaptation and reconfiguration capabilities to handle actuator

failure and specify certain actuators for load alleviation.

4.2 Model Predictive Control for Load Alleviation

MPC for GLA minimizes the gust loads via suitably calculated actuation signals u using the measurements y_m , and the gust information ω_g . The mathematical formulation of MPC for GLA is considered in this work as follows.

$$\min_{x,u} J = \min_{x,u} \sum_{j=1}^{N-1} \ell(y_m(k+j|k), y_l(k+j|k), u(k+j|k)) + E(x(k+N|k)), \quad (4.1a)$$

$$\text{s.t. } \forall j \in \{0, \dots, N-1\}$$

$$x(k|k) = \hat{x}(k), \quad (4.1b)$$

$$x(k+j+1|k) = f(x(k+j|k), u(k+j|k), \omega_g(k)), \quad (4.1c)$$

$$y_m(k+j|k) = g_m(x(k+j|k), u(k+j|k)), \quad (4.1d)$$

$$y_l(k+j|k) = g_l(x(k+j|k), u(k+j|k)), \quad (4.1e)$$

$$x(k+j|k) \in \mathbb{X}, \quad u(k+j|k) \in \mathbb{U}, \quad \Delta u(k+j|k) \in \Delta \mathbb{U}, \quad (4.1f)$$

$$x(k+N|k) \in \mathbb{X}_T. \quad (4.1g)$$

Here, $x \in \mathbb{R}^{n_x}$ are the aircraft states, e.g., describing the flight-mechanics (rigid-body motions) and the aeroelastic modes. $x(k|k)$ is the initial condition and \hat{x} the estimated value. $u \in \mathbb{R}^{n_u}$ are control inputs, e.g., signals to throttle, elevator, flaps and aileron, see Fig. 4.2. $\omega_g(k) \in \mathbb{R}^{n_g}$ is gust velocity. The map $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_g} \rightarrow \mathbb{R}^{n_x}$ is the rigid-body dynamics (i.e., EOM) and flexibility characteristics, i.e., structural deflections. $g_m(\cdot, \cdot) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_m}$ and $g_l(\cdot, \cdot) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_l}$ represent the measured and performance output as functions of the state and inputs. The measured outputs $y_m \in \mathbb{R}^{n_m}$ represent the information of the sensors, e.g., rigid-body state, aerodynamic angles, and structural variables and loads. The performance output $y_l \in \mathbb{R}^{n_l}$ defines the quantities of interest desired to be optimized, and they may be unmeasurable.

Eq. (4.1f) are constraints on the aircraft state $x(j)$ and control inputs on the deflection range $u(j)$ and the response rate $\Delta u(j)$, respectively. The functional $\ell : \mathbb{R}^{n_m} \times \mathbb{R}^{n_l} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}_0^+$ represents the objective cost, e.g., loads reduction y_l , control input u , and the deviations from the trim conditions y_m . The terminal constraint (4.1g) and cost $E : \mathbb{R}_x^n \rightarrow \mathbb{R}_0^+$ are mainly imposed for stability purposes.

At every time k , the optimization problem (4.1) is solved online to determine the optimal control sequence $\mathbf{u}^*(k) = \{u^*(k), u^*(k+1), u^*(k+2), \dots, u^*(k+N)\}$. Only the first term $u^*(k)$ is applied to the aircraft actuators, then both prediction and optimization are repeated at the next time instant $k+1$ with a receding horizon

fashion when new measurements are available. Before examining the MPC approach for load alleviation, it is important to outline the main coordinate frames used to describe the aircraft motions, i.e., rigid and flexible bodies.

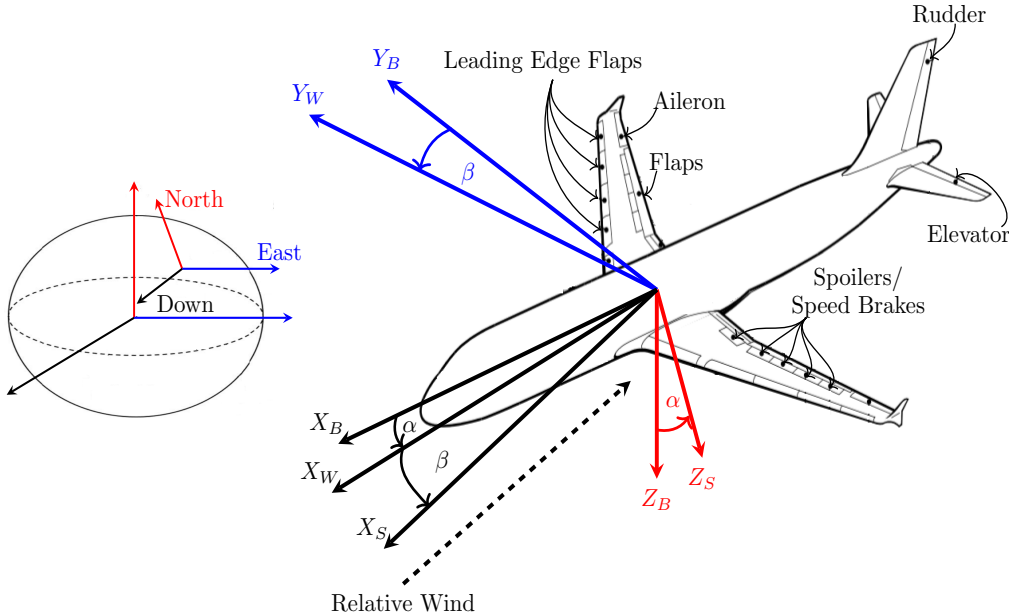


Figure 4.2: Aircraft reference frames, aerodynamic angles, and control surfaces, adapted from [47, 64].

4.2.1 Aircraft Reference Axes Frames

The aircraft motions are usually described in terms of translation and rotation variables in the reference frames, see Fig. 4.2. For instance, mission requirements, map information, and EOM are represented in the *inertial coordinate frame* with an origin at a convenient point on the Earth's surface [47, 64]. Here, the horizontal axes X_I, Y_I are pointing to the north and east, respectively. As the Z_I axis is pointing downward to the earth center, this reference frame is often known as a *north-east-down* frame [47, 64]. Sensor measurements and the actuators' forces and torques are defined with respect to the *body reference frame* whose origin is fixed at the aircraft's CG [47, 64] and whose axes X_B, Y_B, Z_B are towards the longitudinal (airframe nose), lateral (right-wing), and downward directions, respectively. The $X_B - Z_B$ plane is coplanar with the symmetry plane.

Generally, the aerodynamic forces and moments are defined by the aerodynamic angles, AoA α and the sideslip angle β , represented in the *wind axis frame*, where the origin is at the aircraft's CG with its x-axis along the velocity direction [47, 64]. The x-axis of the *stability frame* is set in alignment with the relative wind according

to aircraft AoA. This angle, between the wings and airspeed vector, is required to generate lift. The transformations between these coordinate frames are performed using the aircraft attitude, i.e., Euler angles, e.g., roll, pitch, and yaw angles [47, 64]. In this thesis, we use the coordinate transformation also to define the coupling between the planning and the low-level control layers in the next chapters.

4.2.2 Flexible Aircraft Model

The first step to designing a model-based control approach is to develop an appropriate aircraft model used to predict future behavior. According to these predictions, the optimal control sequence is determined, so the optimization program is constrained by the model while fulfilling the constraints. For choosing an appropriate model, one must consider many factors, e.g., prediction accuracy, computation complexity, and control performance and stability [17]. A reasonable rule is to choose the simplest possible model accurate enough to make reasonable predictions. MPC-based GLA systems have been proposed using different aircraft models, e.g., geometrically-nonlinear beam models [152], and aeroelastic models [51], or linear, aeroelastic, reduced-order models [67, 145].

For flexible aircraft, the resulting elastic deformation changes the aerodynamic loads, which leads to further changes in elastic deformation. These interactions can lead to structural oscillation and change rigid-body aerodynamic and flight modes [1]. Therefore describing the flexible structure interaction requires a unified aeroelastic analysis incorporating simultaneously geometrically elastic modules (structural dynamics) with rigid-body aerodynamics [51]. The structural model provides distributed mass and stiffness information of the airframe to calculate the structural deformations resulting from external forces. As a result, the airframe is often considered as an elastic body acted upon by external forces, depending on the structural state, e.g., geometry and velocity. The nonlinear aircraft dynamics (4.1c), used in the simulation environment, is represented as [153, 154]:

$$\begin{bmatrix} m \left(\dot{V}_B + \Omega_B \times V_B + T_{BE} [0 \ 0 \ g]^\top \right) \\ I_B \dot{\Omega} + \Omega_B \times (I_B \Omega_B) \end{bmatrix} = M_{rb}^\top F_{ext}, \quad (4.2a)$$

$$M_{fm} \ddot{a}_f + M_{fd} \dot{a}_f + M_{fs} a_f = M_{fb}^\top F_{ext}, \quad (4.2b)$$

where m and I_B are the aircraft mass and mass moment of inertia matrix, respectively. $V_B = [V_x \ V_y \ V_z]^\top$ and $\Omega_B = [p \ q \ r]^\top$ are the translational and rotational velocities in the x, y, z body axes, respectively. T_{BE} is the transformation matrix from the earth E to body B axes. M_{rb}, M_{fb} represent the modal matrix for the rigid and flexible modes, respectively. The structural model (4.2b) includes the distributed mass M_{fm} , damping M_{fd} and stiffness M_{fs} matrices of the flexible modes, to calculate the structural

deformations at generalized coordinates a_f resulting from external forces F_{ext} .

Practically, each aircraft is equipped with different control surfaces, e.g., aileron, flaps, and elevator, see Fig. 4.1, to achieve the control commands by deflecting the surface with a certain time delay. Moreover, every actuator has physical constraints, e.g., deflection limits and rate saturation. Therefore, it is necessary to take the actuator model, time-delay, and constraints into account inside the MPC formulation. In our study, there are two aircraft with different actuation system, see Section 4.3.

4.2.3 Aircraft Model Linearization and Reduction

To facilitate the control synthesis, the nonlinear model (4.1c) is linearized at the trim conditions, e.g., steady horizontal flight. Thereby, we consider a steady-state condition in the absence of gust effects, i.e., $\omega_g = \mathbf{0}$:

$$\begin{aligned} V_y = V_z = \phi = \psi = p = q = r = 0, \\ \dot{p}_d = \dot{V}_x = \dot{V}_y = \dot{V}_z = \dot{\phi} = \dot{\theta} = \dot{\psi} = \dot{p} = \dot{q} = \dot{r} = 0. \end{aligned}$$

Then both the rigid-body dynamics and flexible modes are taken into account to obtain the linear time-invariant model of the underlying flexible aircraft as:

$$\dot{x}(t) = Ax(t) + Bu(t) + B_g\omega_g(t), \quad (4.3a)$$

$$y_m(t) = C_mx(t) + D_mu(t) + D_{mg}\omega_g(t), \quad (4.3b)$$

$$y_l(t) = C_lx(t) + D_lu(t) + D_{lg}\omega_g(t), \quad (4.3c)$$

where $A, B, B_g, C_m, D_m, C_l, D_l, D_{mg}$, and D_{lg} are the linear model matrices.

The structural dynamics models have many structural states [4]. To improve the computational speed, the model order can be reduced by projecting full order models onto a subspace, which encapsulates the fundamental dynamics of the aircraft with a smaller number of state [70].

The linear model (4.3) is then discretized with a sampling time T_s

$$x(k+1) = A(T_s)x(k) + B(T_s)u(k) + B_g(T_s)\omega_g(k), \quad (4.4a)$$

$$y_m(k) = C_m(T_s)x(k) + D_m(T_s)u(k) + D_{mg}(T_s)\omega_g(k), \quad (4.4b)$$

$$y_l(k) = C_l(T_s)x(k) + D_l(T_s)u(k) + D_{lg}(T_s)\omega_g(k). \quad (4.4c)$$

This model is used within the MPC formulation (4.1) to predict the future response.

The main idea of the MPC scheme, see Fig. 4.1, is to design a feedback control strategy using state information. However, the direct measurements of all state of the flexible aircraft are usually not available [1]. Therefore, it is necessary to combine the MPC scheme with a stable state estimator.

4.2.4 State Estimation

A state estimation calculates dynamically the estimated state $\hat{x}(k)$ via fusing process model with the available measurement information y_m , gust measurements $\omega_g(k)$, and actuation signals $u(k)$. In this linear case, one can use a so-called Luenberger observer:

$$\hat{x}(k+1|k) = A\hat{x}(k|k-1) + Bu(k) + B_g\omega_g(k) + L(y_m(k) - \hat{y}_m(k|k-1)), \quad (4.5a)$$

$$\hat{y}_m(k|k-1) = C_m\hat{x}(k|k-1) + D_mu(k|k-1) + D_{mg}\omega_g(k), \quad (4.5b)$$

where $\hat{x}(k+1|k)$ is the estimated value of the state x at time $k+1$ based on the available information at time k . $\hat{y}_m(k|k-1)$ is the output estimation at time k based on the available information at previous time $k-1$. The objective is that the estimation error $e(k) = x(k) - \hat{x}_m(k|k-1)$, i.e., the difference between actual state vector x and its reconstructed counterpart \hat{x} converges to zero. L is a feedback gain matrix, necessary for convergence in case of an unstable A , and otherwise improves the estimation speed, as the Luenberger observer is stable if the matrix $\hat{A} - L\hat{C}_m$ has all the eigenvalues inside the unit circle. Then, the error between the estimated state and the actual one converges to zero.

4.2.5 Control Objectives Setup for Load Alleviation

The control objectives focus on alleviating the structural loads and stresses without decreasing the handling qualities and passenger comfort. To achieve the real-time requirements in this work, we consider the quadratic cost function

$$J(x(j), u(j)) = \sum_{j=1}^{N-1} \left(y_m^\top(j) Q_m y_m(j) + y_l^\top(j) Q_l y_l(j) + u^\top(j) R_u u(j) \right) + x^\top(N) Q_N x(N).$$

where $Q_m, Q_l, Q_N \geq 0$, and $R_u > 0$ are the weighting matrices for the measured and performance output, terminal state, and control, respectively. Picking the cost functional, i.e., the weighting matrices, suitably is essential to achieve stability and good control performance. For instance, the wing-root is the most critical station exposed to the highest loads, so alleviating the loads at the wing-root leads to an increase in the lifetime of the aircraft structure. The aircraft structure is practically designed to withstand high shear forces, so they are not included in the cost function. Therefore, the performance output y_l includes some loads criteria, e.g., bending M_b^i , and torsional moments M_t^i as response to the atmospheric gust at station $i \in \{1, \dots, N_s\}$, e.g., the wing root, the mid-wing, and the horizontal tail:

$$y_l = [M_b^1, M_t^1, \dots, M_b^i, M_t^i, \dots, M_b^{N_s}, M_t^{N_s}]^\top,$$

where N_s is the stations number. The control objective penalizes the deviations of the aircraft altitude h and velocity V_t from the trim point by considering the measured output vector

$$y_m = [h, V_t, \alpha, n_z(x^i, y^i, z^i)]^\top.$$

Furthermore, one needs to ensure that the aircraft variables are within the safety range. The passenger comfort criterion depends mainly on the vertical acceleration $n_z(x^i, y^i, z^i)$, at some points (x^i, y^i, z^i) , e.g., the nose, CG, and the tail section.

The third term $u^\top(j)R_u u(j)$ accounts for the control inputs to minimize energy consumption. However, using some actuators, e.g., elevator, to alleviate the gust loads, can increase internal stresses in the horizontal tail root. Therefore, it is recommended to specify a different weight for each control input.

The terminal constraint (4.1g) and cost E are used to enforce stability.

4.2.6 Stability of MPC/GLA Approach

MPC solves an OCP over a finite prediction horizon, which may lead to stability loss, i.e., the finite-horizon solution does not always lead to a stable closed loop [20]. Therefore, we investigate the stability of the closed-loop MPC scheme (4.1) before the real-time implementation. This investigation is inspired by the previous theoretical studies concerning MPC application [21, 155], particularly for the GLA problem, e.g., [51, 78]. To do so, we suppose the following assumptions:

Assumption 4.1 (*Stabilizability*) *the pair (A, B) is stabilizable.*

Assumption 4.2 (*Full-state measurement*) *the exact information of the aircraft state $x(k)$ is available at each time k .*

Assumption 4.3 (*Terminal set*) *there exists an admissible terminal set $\mathbb{X}_T \subset \mathbb{X}$ (4.1g) and a terminal control law $\kappa_T(x) = Kx \in \mathbb{U}, K \in \mathbb{R}^{n_u \times n_x}, \forall x \in \mathbb{X}_T$ such that if $x \in \mathbb{X}_T$, then $Ax + B\kappa_T(x) \in \mathbb{X}_T$, i.e., \mathbb{X}_T is positively invariant under the control law $\kappa_T(\cdot)$.*

Assumption 4.4 (*Terminal set reachability*) *the admissible terminal set $\mathbb{X}_T \subset \mathbb{X}$ (4.1g) is reachable for any prediction horizon $N > 1$.*

Assumption 4.5 (*Weighting matrices*) *assume the matrices $Q_m \geq 0, Q_l \geq 0, R_u > 0$, $Q = C^\top Q_y C$, $Q_y = \begin{bmatrix} Q_m & \mathbf{0} \\ \mathbf{0} & Q_l \end{bmatrix} \geq 0$, and $(A, Q^{1/2})$ is detectable.*

Assumption 4.6 (Terminal penalty) the terminal penalty

$$E(x(N)) = x^\top(N)Q_Nx(N) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}_0^+, \quad Q_N > 0,$$

is assumed to be equal to the value function V_{uc}^* of the unconstrained infinite-horizon OCP $\mathbb{P}_{\text{uc}}^\infty$:

$$V_{\text{uc}}^* := \min_{\kappa_{\text{T}}(x)} \sum_{j=1}^{j=\infty} \ell(x(j), u(j)) = \min_{\kappa_{\text{T}}(x)} \sum_{j=1}^{j=\infty} \frac{1}{2} (x^\top(j)Qx(j) + u^\top(j)Ru(j)),$$

where $R = R_u + D^\top Q_y D > 0$ and $u = \kappa_{\text{T}}(x)$ is the optimal controller for OCP $\mathbb{P}_{\text{uc}}^\infty$.

Assumption 4.7 (No prediction error) there is no system-model mismatch and no gust, i.e., $\omega_g(k+j) = 0, \forall j \in \{1, \dots, N\}$.

Proposition 4.1 (Nominal stability)

Let Assumptions 4.1-4.7 hold, then the MPC scheme (4.1) is exponentially stabilizing the system (4.4) for any prediction horizon $N > 1$.

Proof As the system is linear and constrained, Assumption 4.6 implies that the terminal penalty

$$E(x(N)) = x^\top(N)Q_Nx(N) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}_0^+,$$

is chosen to be equal to the value function V_{uc}^* for the unconstrained infinite horizon optimal control problem $\mathbb{P}_{\text{uc}}^\infty$.

Assumption 4.7 implies no gust, i.e. $\omega_g(k+j) = 0$, Therefore, we can reformulate the stage cost functional $\ell(y_m(j), y_l(j), u(j))$ as:

$$\ell(y_m(j), y_l(j), u(j)) = y_m^\top(j)Q_m y_m(j) + y_l^\top(j)Q_l y_l(j) + u^\top(j)R_u u(j),$$

as $Q_m \geq 0, Q_l \geq 0$, and take $y = \begin{bmatrix} y_m \\ y_l \end{bmatrix}$, and $Q_y = \begin{bmatrix} Q_m & \mathbf{0} \\ \mathbf{0} & Q_l \end{bmatrix} \geq 0$, then

$$\begin{aligned} \ell(y_m(j), y_l(j), u(j)) &= y^\top(j)Q_y y(j) + u^\top(j)R_u u(j) \\ &= \left[\begin{array}{cc} C & D \end{array} \begin{bmatrix} x(j) \\ u(j) \end{bmatrix} \right]^\top Q_y \left[\begin{array}{cc} C & D \end{array} \begin{bmatrix} x(j) \\ u(j) \end{bmatrix} \right] + u^\top(j)R_u u(j) \\ &= \left[\begin{array}{cc} x^\top(j) & u^\top(j) \end{array} \begin{bmatrix} C^\top \\ D^\top \end{bmatrix} \right] Q_y \left[\begin{array}{cc} C & D \end{array} \begin{bmatrix} x(j) \\ u(j) \end{bmatrix} \right] + u^\top(j)R_u u(j) \\ &= \left[\begin{array}{cc} x^\top(j) & u^\top(j) \end{array} \begin{bmatrix} C^\top Q_y C & C^\top Q_y D \\ D^\top Q_y C & D^\top Q_y D \end{bmatrix} \begin{bmatrix} x(j) \\ u(j) \end{bmatrix} \right] + u^\top(j)R_u u(j) \\ &= x^\top(j)C^\top Q_y C x(j) + u^\top(j)(R_u + D^\top Q_y D)u(j) + 2x^\top(j)C^\top Q_y D u(j) \end{aligned}$$

where $Q = C^\top Q_y C x(j)$, $R = R_u + D^\top Q_y D$, and $N = C^\top Q_y D$. It then holds that

$$\ell = x^\top(j)Qx(j) + u^\top(j)Ru(j) + 2x^\top(j)Nu(j). \quad (4.6)$$

As $Q_y \geq 0$,

$$Q = C^\top Q_y C \geq 0. \quad (4.7)$$

The next step is to prove that

$$Q - NR^{-1}N^\top \geq 0 \Leftrightarrow \begin{bmatrix} C^\top Q_y C & C^\top Q_y D \\ D^\top Q_y C & D^\top Q_y D \end{bmatrix} \geq 0. \quad (4.8)$$

As

$$\begin{aligned} & \begin{bmatrix} x^\top(j) & u^\top(j) \end{bmatrix} \begin{bmatrix} C^\top Q_y C & C^\top Q_y D \\ D^\top Q_y C & D^\top Q_y D \end{bmatrix} \begin{bmatrix} x(j) \\ u(j) \end{bmatrix} \\ &= \begin{bmatrix} x^\top(j)C^\top & u^\top(j)D^\top \end{bmatrix} \begin{bmatrix} Q_y & Q_y \\ Q_y & Q_y \end{bmatrix} \begin{bmatrix} Cx(j) \\ Du(j) \end{bmatrix} \\ &= \begin{bmatrix} \bar{x}^\top(j) & \bar{u}^\top(j) \end{bmatrix} \begin{bmatrix} Q_y & Q_y \\ Q_y & Q_y \end{bmatrix} \begin{bmatrix} \bar{x}(j) \\ \bar{u}(j) \end{bmatrix}. \end{aligned}$$

we calculate the eigenvalues of the matrix $\begin{bmatrix} Q_y & Q_y \\ Q_y & Q_y \end{bmatrix}$:

$$\begin{aligned} \det \begin{bmatrix} \lambda \mathbf{I} - Q_y & -Q_y \\ -Q_y & \lambda \mathbf{I} - Q_y \end{bmatrix} &= \det \left((\lambda \mathbf{I} - Q_y)^2 - Q_y^2 \right) \\ &= \det (\lambda^2 \mathbf{I} - 2\lambda Q_y) = \det \lambda^{n_y} (\lambda \mathbf{I} - 2Q_y). \end{aligned}$$

As $Q_y \geq 0$, also $2Q_y \geq 0$ and $\begin{bmatrix} Q_y & Q_y \\ Q_y & Q_y \end{bmatrix} \geq 0$. Therefore,

$$\begin{bmatrix} C^\top Q_y C & C^\top Q_y D \\ D^\top Q_y C & D^\top Q_y D \end{bmatrix} \geq 0 \Leftrightarrow Q - NR^{-1}N^\top \geq 0. \quad (4.9)$$

Due to Assumption 4.5 and Equations (4.7) and (4.8), the optimal solution of the OCP (4.6) stabilizes the system (4.3), see [156].

As a result, the weighting matrix Q_N of the terminal penalty $E(\cdot)$ satisfies:

$$Q_N \geq Q + K^\top R K + (A - BK)^\top Q_N (A - BK). \quad (4.10)$$

Thus, $E(\cdot)$ is a Lyapunov function, i.e.

$$E\left(f(x(k), \kappa_T(x(k)))\right) - E(x(k)) + \ell\left(x(k), \kappa_T(x(k))\right) \leq 0, \forall x \in \mathbb{X}_T,$$

and the cost monotonicity condition of the cost function is satisfied, i.e.,

$$J(u^*; x(k); \omega_g = 0; N + 1) \leq J(u^*; x(k); \omega_g = 0; N).$$

Assumption 4.5 allows applying LaSalle's invariance principle, which shows that the closed-loop system is exponentially stable with a domain of attraction \mathbb{X}_T [155]. As this set is by Assumption 4.3 reachable from any $x \in \mathbb{X}$, the closed loop is globally exponentially stable. \blacksquare

Next, we discuss stability in the presence of bounded finite-time gusts.

Assumption 4.8 (*Bounded finite-time gusts*) *The gusts acting on the aircraft are bounded and vanish after a finite time N_g , i.e.,*

$$\omega_g(k+j) \begin{cases} \in \mathbb{G} \subset \mathbb{R}^{n_g}, & \forall j \in \{1, \dots, N_g\}, \\ = 0, & \forall j \in \{N_g + 1, \dots, N\}. \end{cases}$$

Proposition 4.2 (*Nominal Stability in presence of bounded finite-time gusts*) *Let Assumptions 4.1-4.5 hold, assume finite bounded gusts, i.e., Assumption 4.8, and choose the terminal penalty $E(x(N)) = x(N)^\top Q_N x(N)$ according to Proposition 4.1. Then the MPC scheme (4.1) is globally exponentially stabilizing the system (4.4) for any prediction horizon $N > N_g \geq 1$.*

Proof A bounded gust acting on a linear system (4.4), causes a bounded state deviation, for any time $j \in \{1, \dots, N_g\}$. Assumption 4.8 implies that the gust has vanished after the finite time N_g . Thus, by Proposition 4.1, the MPC (4.1) exponentially stabilizes (4.4), i.e.:

$$\forall j \in \{N_g, \dots, N\}, \quad x(k+j) \leq \beta(x(N_g)) \gamma^j, \quad \|\gamma\| < 1. \quad (4.11)$$

Due to the linearity of the model and the boundedness of the gust, this exponential bound can be extended on the states $x(k+j), \forall j \in \{1, \dots, N_g\}$ by enlarging the overshoot constant to some $\beta_g(x(N_g)) > \beta(x(N_g))$:

$$\forall j \in \{1, \dots, N_g\}, \quad x(k+j) \leq \beta_g(x(N_g)) \gamma^j, \quad \|\gamma\| < 1. \quad (4.12)$$

Thus, the MPC scheme (4.1) is globally exponentially stabilizing the system (4.4) in presence of bounded finite-time gusts. \blacksquare

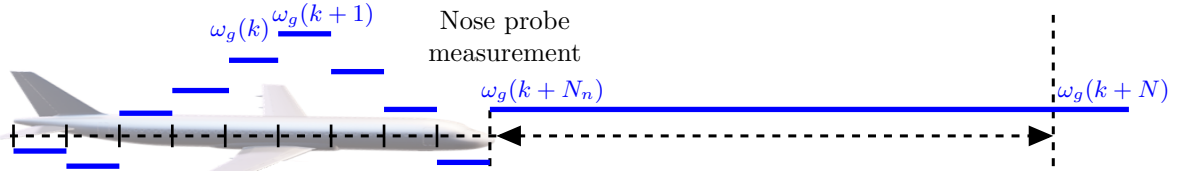
4.2.7 Improving MPC for GLA with Preview Gust Information

MPC allows utilizing the preview of gust information to improve predicting the future state and gust, see Fig. 4.3.

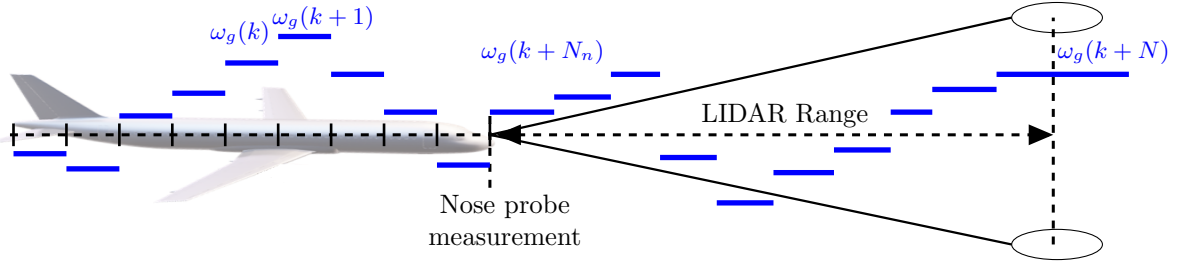
In the first scenario, see Fig. 4.3a, the gust velocity is measured using an onboard AoA probe at the aircraft nose and stored at the current time step k to be used within the MPC formulation (4.1). So the estimated gust velocity sequence $\bar{\omega}_g$ [m/s] at specified positions on the aircraft structure, e.g., CG, tail, or wing-root is:

$$\bar{\omega}_g = [\omega_g(k), \omega_g(k+1), \dots, \omega_g(k+N_n), \omega_g(k+N_n+1), \dots, \omega_g(k+N)],$$

where N_n is the number of time steps in the future that the gust takes to affect these positions. This time depends on the cruise velocity and length of the aircraft. The unmeasured gust terms $\omega_g(k+N_n+1), \dots, \omega_g(k+N)$ are assumed to be constant, i.e., equal zero or the last measured value $\omega_g(k+N_n)$.



(a) Gust measurement via nose probe, and the unmeasured gust is assumed to be constant.



(b) Gust measurement via nose probe and LIDAR sensor, which provide the preview information of the incoming gust further distance ahead of the aircraft.

Figure 4.3: Comparison of gust measurement via nose probe and LIDAR sensor.

In the second scenario, see Fig. 4.3b, LIDAR provides the preview information of the incoming gust certain distance ahead of the aircraft, typically 60-300 m [145, 146]. This improves the GLA performance as the prediction accuracy of MPC adversely affects control performance. According to the LIDAR range and the prediction horizon, the gust information will be available, i.e.,

$$\bar{\omega}_g = [\omega_g(k); \omega_g(k+1), \dots, \omega_g(k+N_n), \omega_g(k+N_n+1), \dots, \omega_g(k+N)].$$

This preview information provides the MPC scheme enough time to counteract dis-

turbances before they arrive at the critical stations, e.g., wing-root. As will be shown, this updated MPC with improved prediction yields a significant load reduction and better usage of the control surfaces. However, the sensor needs to provide an accurate measurement for the gust perturbation to apply the feed-forward load alleviation.

4.3 Implementation

The main objective of this section is to demonstrate the potential for design and real-time implementation of the GLA approach. First, we outline two reference aircraft as case studies with different state, output, and control configurations. We then outline the real-time implementation of the proposed MPC approach.

4.3.1 Aircraft Models Considered for Simulation Studies

We consider two models, the Green Regional Aircraft and the Remos GX aircraft.

The **Green Regional Aircraft** is one of the six Clean-Sky platforms representing a regional transport airplane with 130 seats and T-tail configuration, see Fig. 4.5a. This aircraft has fuselage-mounted engines and is equipped with winglets.

The aircraft was designed for weight reduction, energy, aerodynamics efficiency, and low pollution and noise levels [143]. Two mass configurations were considered, zero fuel weight and maximum take-off weight. Table A.1 lists the aircraft geometry [143].

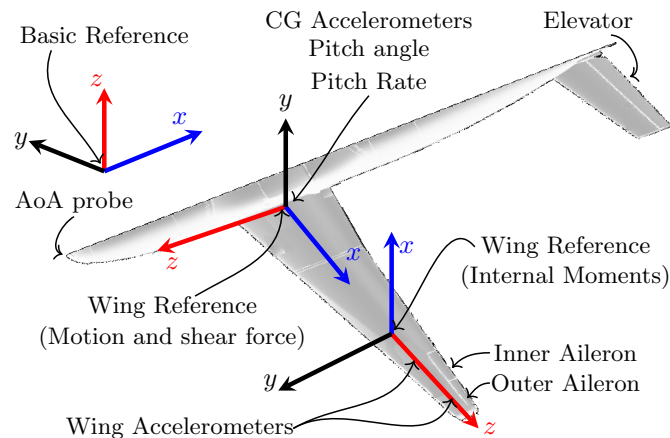


Figure 4.4: Green Regional Aircraft: actuators and sensors configuration and reference systems: fuselage accelerations and displacements are expressed using the basic reference system. Wing reference is used to express the displacement, velocities and accelerations of points on the wing. Second wing reference expresses the internal moments, e.g., shear, bending, and torsional, adapted from [157].


 (a) Green Regional Aircraft¹.


(b) Remos GX Aircraft [158]

Figure 4.5: Reference Aircraft with different state, output and control configurations.

The GLA system was designed based on three control surfaces, e.g., elevator and inboard and outboard aileron, see Fig. 4.4. All the surfaces and motors have positive rotation when the trailing edge is moved downward. The dynamics and limits of the actuators driving the control surfaces are inserted in the numerical model by

$$\delta_s(s) = \frac{\omega_0^2}{s^2 + 2\xi\omega_0s + \omega_0^2}\delta_c(s) - \frac{M_h(s)}{K_\delta} \quad |\delta_s| \leq \delta_m, \quad |\dot{\delta}_s| \leq \dot{\delta}_m, \quad |M_h| \leq M_{hm} \quad (4.13)$$

where δ_s is the actual surface deflection, δ_c is the commanded deflection, M_h is the moment acting on the control surfaces hinges and K_δ represents the actuator stiffness. Two types of saturation were included: saturation limits imposed on the surface deflection angle $|\delta_s| \leq \delta_m = \pm 15^\circ$, and on the deflection rate $|\dot{\delta}_s| \leq \dot{\delta}_m = \pm 50^\circ/\text{s}$, in addition to the torque-dependent saturation on the hinge moment $|M_h| \leq M_{hm} = \pm 10000 [\text{Nm}]$ [141, 157]. The nonlinearities are due to the internal saturation of the actuators and the dependency on the output hinge moments [141, 157].

The nonlinear model is linearized around the steady trimmed condition in the absence of gust to facilitate the control synthesis. The resulting aircraft model has 153 states expressed in three different reference systems, see Fig. 4.4 and Appendix A.1 [141, 143, 157]. The measurement output vector contains 34 variables, e.g., AoA measurement, velocity, and the structural displacements and loads; for more details, see [141, 143, 157]. We investigated the model dynamics, see Fig. A.1, to derive a discrete-time linear model. The frequency of the fastest mode equals 58.4[Hz], so we decided to discretize the model with a fixed time-step of $T_s = 10 \text{ ms}$.

Remos GX, the second case study, is a two-seat ultralight aircraft with fiber composite construction, see Fig. 4.5b. This aircraft has a high-wing design with struts and a front-mounted engine and propeller with three-wheel landing gears [153].

The coupling of the structural and the aerodynamical model is achieved through a surface spline. The simulation model is linearized around a trim point. The model state includes the aircraft velocities $V_B = [V_x \ V_y \ V_z]^\top$, attitudes ϕ, θ, ψ , rotational rates $\Omega = [p, q, r]^\top$, and 3D positions p_n, p_e, p_d in addition to the flexible deformation modes $a_f \in$

¹Source: GLAMOUR Project, <https://home.aero.polimi.it/glamour/>, Accessed: 27.09.2020

\mathbb{R}^{10} , and modal velocities $\dot{a}_f \in \mathbb{R}^{10}$. The control inputs $u = [\delta_{elev}, \delta_{rud}, \delta_{flap}, \delta_{ar}, \delta_{al}]^T$ are elevator, rudder, flap, right, and left ailerons, respectively. The MPC formulation includes the aircraft constraints in (4.1f), which are here the physical limitations and the flight envelope, e.g., safety range, see Table A.2. The output of the state-space model consists of the rigid-body state, vertical load factor, AoA, and the structural displacements and velocities. The strain gauges measure the structural loads (forces and moments) at the specified points. $\omega_g \in \mathbb{R}^{10}$ is the gust velocity vector at ten stations on the aircraft structure [153].

4.3.2 Gust Loads Conditions

For passenger aircraft, civil airworthiness requirements explicitly specify the tolerable aircraft response to both discrete gusts and continuous turbulence [3].

For instance, the EASA airworthiness regulations [66] specify the variance and the power spectral density for stochastic (continuous) turbulence, as well as the time history and intensity for deterministic (discrete) gusts [3]. The continuous turbulences are mainly modeled with two random distributions, e.g., Dryden and von Kármán models, in which the power spectral densities are defined in either the frequency domain or time domain [3, 141, 159]. The Dryden and von Kármán distributions are defined by a characteristic scale wavelength and the root-mean-square turbulence velocity [3, 159]. The power spectral density of the von Kármán gust model is

$$\Phi(\Omega) = \sigma_w^2 \frac{L}{\pi} \frac{1 + \frac{8}{3} (1.339L\Omega)^2}{[1 + (1.339L\Omega)^2]^{11/16}}, \quad (4.14)$$

where σ_w^2 is the velocity variance in the turbulent field and L is the turbulence length scale. This model approximates experimental data well but is quite unpractical to use since it requires an irrational filter. For this reason, a simpler model, the Dryden spectra, is often used in computations:

$$\Phi(\Omega) = \sigma_w^2 \frac{L}{\pi} \frac{1 + 3(\Omega L)^2}{[1 + (\Omega L)^2]^2}. \quad (4.15)$$

Discrete gusts are commonly represented in three forms; step, ramped, and $(1 - \cosine)$ gust profile [3]. The latter one is specified according to the EASA regulations

$$v_g = \frac{V_{gust}}{2} \left[1 - \cos\left(\frac{\pi s}{H}\right) \right], \quad 0 < s \leq 2H, \quad (4.16)$$

where the distance s [m] is the gust penetration, the gust gradient H [m] is the distance over which the gust acts, and V_{gust} [m/s] is the local peak gust velocity, in equivalent

airspeed. The latter is defined as

$$V_{gust} = V_{ref} F_g \left(\frac{H}{350} \right)^{\frac{1}{6}}, \quad (4.17)$$

where V_{ref} is a function of the altitude, while F_g increases linearly from zero at sea level to one at the maximum operating altitude.

4.3.3 Real-Time Solution of MPC Implementation

The real-time implementation of the MPC requires considering many factors, e.g., the maximum computational time, required memory, and numerical algorithms to be implemented on embedded platforms [160]. The required memory depends on the prediction horizon, the system dimension, i.e., the number of inputs, state, and constraints [105]. The increased capability of computer hardware and fast QP techniques have made it possible to apply MPC to fast dynamics problems.

For the real-time implementation of the GLA system, we used the code generation software package μ AO-MPC [114]. μ AO-MPC generates a highly portable library-free C-code for MPC on embedded applications with low computational requirements [114]. The optimization problem (4.1) is cast in the form of a general QP, which is solved online via a QP solver based on an augmented Lagrangian method with Nesterov's gradient method. The constraint equations, gradient vector, and Hessian matrix related to the optimization problem are precalculated in condensed form. This leads to extremely fast implementation with low-memory requirements.

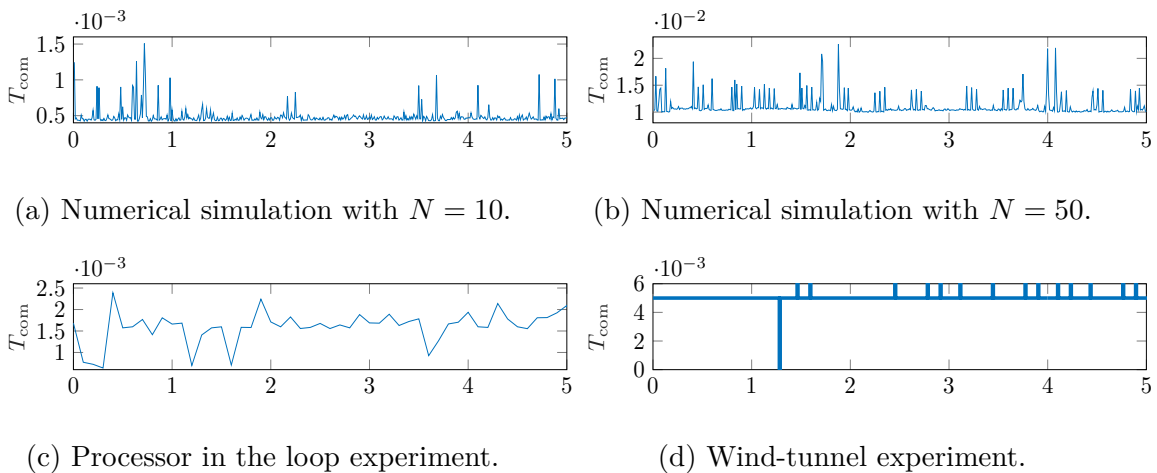


Figure 4.6: μ AO-MPC computational time T_{com} s: increasing the prediction horizon N increases the time-to-solve for calculating the control inputs. Computational time must be smaller than the sampling time of $T_s = 10$ ms.

First, we investigate the timing characteristics of the MPC approach to ensure that

the optimization problem (4.1) is solved online with a reasonable time, i.e., less than the system's sample-time. Fig. 4.6 depicts the corresponding timing characteristics to solve the MPC problem using μ AO-MPC on an Intel[®] Core[™] i7-6700 CPU @ 3.40GHz.

The 10 steps horizon provides adequate performance with a mean time-to-solve of 0.45 [ms] while increasing on prediction horizon (50 steps) provides better results but increases the time-to-solve (10.6 [ms]). It is important to mention that the operating system also performed the nonlinear simulation and computed the control input. This results in some maxima time-to-solve reaching up to 1.51 [ms] and 22.64 [ms] for $N = 10$ and $N = 50$, respectively, see Fig. 4.6. The mean time-to-solve is less than the underlying sample time; there is also enough safety-margin left to perform the state-estimation and accommodate the real-time implementation requirements.

To measure the computation time accurately, we propose in the next section a real-time processor-in-the-loop simulation.

4.3.4 Processor-in-the-Loop Experiments

Practically, conducting the experimental verifications, e.g., wind-tunnel and flight tests, is costly and time-consuming. Therefore, we propose a real-time processor-in-the-loop (PIL) simulation, see Fig. 4.7, to bridge the gap between the numerical simulation and the experimental tests. The proposed PIL framework achieves more realistic simulations to verify the overall control performance. For instance, this framework allows investigating the hardware requirements, e.g., memory usage, and to measure the computation time accurately, see Fig. 4.6c.

In the proposed framework, see Fig. 4.7, two computation modules (onboard hardware and ground station) are integrated with a communication architecture through an Ethernet network via user datagram protocol (UDP). In the ground station module, the primary nonlinear simulation, measurement, and gust generation are performed. The μ AO-MPC code and state estimator are running on the onboard computer. Both computations are performed via two Matlab/Simulink environments. The onboard computer receives the measured output y_m and the gust information ω_g to estimate the full state information \hat{x} . The μ AO-MPC code is implemented as S-function in Simulink to determine the optimal control input u^* . This control signal is delivered to the simulation environment to perform the closed-loop simulation at the next time step. Fig. 4.6c depicts the corresponding time-to-solve of the optimization problem on the onboard computer. As we can see, the computational time is still less than the sampling time. This is very promising to perform the experimental tests.

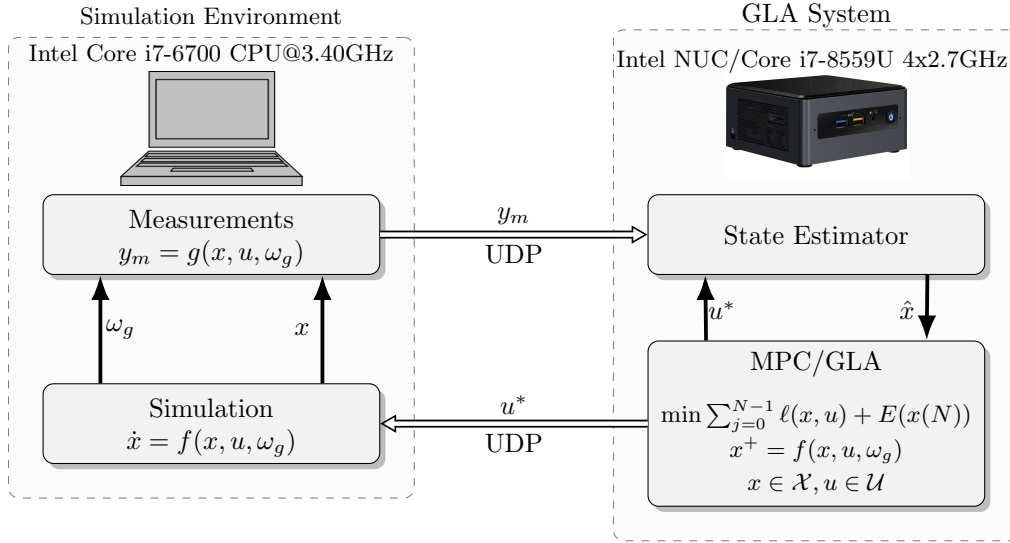


Figure 4.7: PIL schematic diagram with UDP communication protocol. The μ AO-MPC code determines the control input u^* using the state information \hat{x} from the state estimator, which receives the measured output y_m from the main computer, which is computing the main nonlinear simulation

4.3.5 Numerical Simulations Results

A numerical simulation study is used to validate the control performance before the experimental tests. These simulations focus on the discrete vertical gust.

Fig. 4.8a presents the open-loop and closed-loop responses of the first case study, GRA aircraft, passing through a discrete gust ($1 - \cosine$) with and without LIDAR. Fig. 4.8a shows that the MPC/GLA has successfully reduced the load criteria, e.g., wing-root bending moment, by approximately 26%, even 35% when using LIDAR. Also, the wing tail acceleration is decreased by 30% via MPC using the nose probe and 38% by MPC with LIDAR, see Fig. 4.8a. As expected, using LIDAR provides more load reduction and improves the wing fatigue-life significantly because the bending moment does not go through high oscillation.

The performance of the GLA system also depends on the effectiveness and functionality of the control surfaces. The most interesting observation from Fig. 4.8b is that early gust detection (i.e., preview information from LIDAR) provides sufficient lead time to anticipate future loads. This improves the functionality of the control surfaces.

Simultaneously using the actuation surfaces for different functions, e.g., primary control system and alleviating the gust loads, might decrease performance. For instance, using an elevator to alleviate the gust loads can increase the internal stresses in the horizontal tail root. For this reason, different aircraft configurations have been analyzed, e.g., using an elevator or without an elevator. Both configurations achieve

a similar load alleviation performance.

As depicted in Fig. 4.8b, MPC allows using the available actuation configurations while achieving a similar load reduction. In this case, the elevator can be used for the primary flight control system, mainly to regulate the aircraft altitude while the inner and outer ailerons achieve the structural relaxation. The more surprising finding is that the control signals do not reach the maximum deflection limits. Therefore, these constraints are not critical for discrete gust, see Fig. 4.8b.

Fig. 4.9-4.10b present the simulation results for the second case study (Remos-GX airplane) in two cases; using nose probe and LIDAR. MPC-LIDAR utilizes smoother and smaller surface deflection than the case of MPC using the nose probe only. Both MPC approaches stabilize the aircraft at the reference altitude, velocity, and angles with very small rotation. MPC-LIDAR reduces significantly the vertical load factor and the roll rotation. Supplementary video material can be found on <https://www.youtube.com/watch?v=u8dlFdRAQM4>.

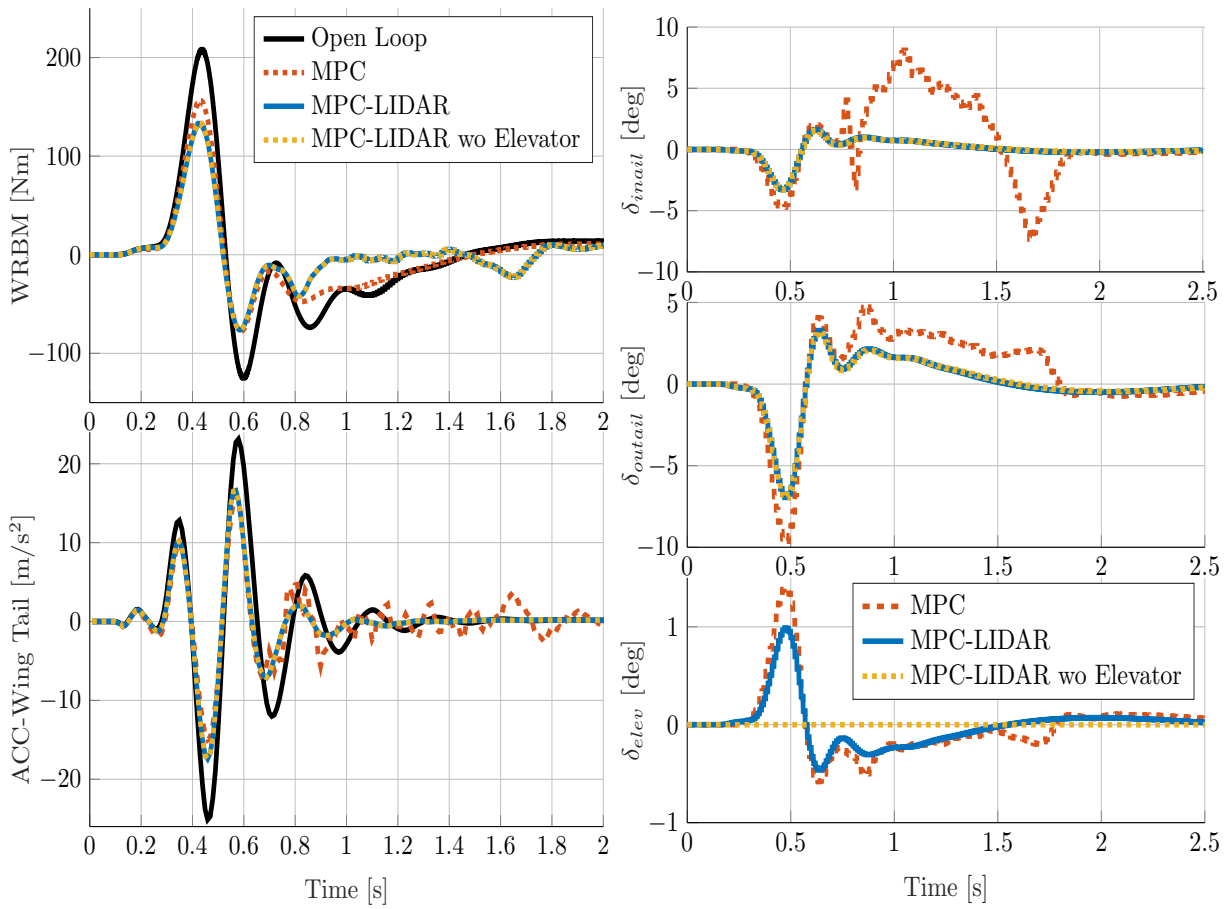
Taken together, the obtained results highlight the applicability of this framework to alleviate the gust loads with less control effort.

4.4 Wind-Tunnel Validation

We validated the proposed control scheme in a series of wind-tunnel (WT) experiments at Politecnico di Milano, see Fig. 4.12 [141, 157]. WT tests are cheaper alternatives to flight tests and allow performing the experiments in a controlled environment. Moreover, the WT tests guarantee repeatability to compare different control strategies [157]. In WT, a left half model of the GRA aircraft, see Fig. 4.12, is placed vertically, so its symmetry plane is parallel to the floor with a support system to guarantee a free motion in plunge and pitch. A weight augmentation system includes an electric motor with force feedback to apply a constant weight force to perform the dynamic response test around the trimmed condition [141, 157].

A suitable scaling process is applied without disregarding critical nonlinear aeroelastic phenomena. The scaling strategies have been mainly performed to two different parameter categories, e.g., dimensional and non-dimensional parameters. The geometry scale has been set to $\lambda_L = 1/6$ because of the maximum size available at the chamber [141, 157]. The time scale was taken unitary $\lambda_T = 1$, to keep unchanged natural frequencies of the structural modes and the bandwidth and rate saturation of the actuators. The density scale was also considered unitary, $\lambda_\rho = 1$, i.e., the air density during the WT tests approximately equals the reference density at sea level [141, 157].

The gust is generated by oscillating gust generator vanes mounted upstream of the WT, see Fig. 4.12, to achieve the specified loading conditions [141, 157]. Since the LIDAR system was not available in the WT, the LIDAR preview information is



(a) Wing-Root bending Moment, Acceleration (b) Control Commands: inner aileron δ_{inail} , outer aileron δ_{outail} , and elevator δ_{elev} .

Figure 4.8: Numerical results for the first case study (GRA aircraft) in three cases: the first simulation uses nose probe, the second one uses LIDAR and elevator, the third is with LIDAR without elevator; MPC-LIDAR utilizes smoother and smaller surface deflection than the case of MPC using the nose probe only. MPC can achieve the same performance without using an elevator.

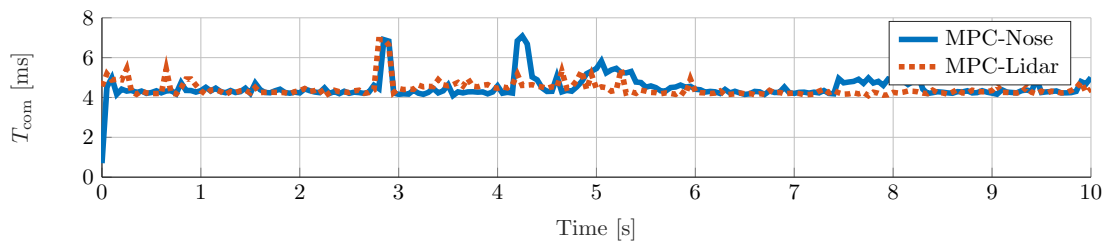
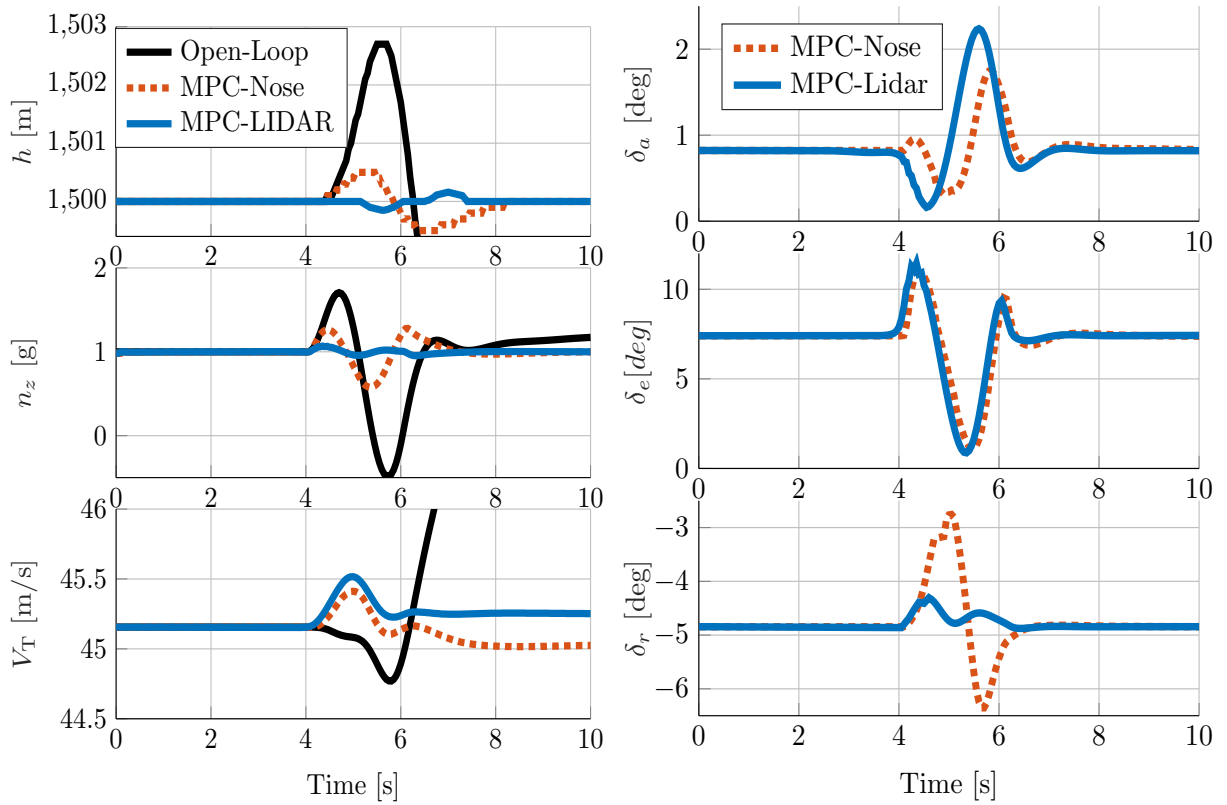


Figure 4.9: Computational time T_{com} is always smaller than the sampling time of $T_s = 50$ ms.



(a) Aircraft altitude h , vertical load factor n_z , (b) Control Commands: aileron δ_a , elevator δ_e , and velocity V_T . and rudder δ_r .

Figure 4.10: Numerical results for the second case study (Remos-GX airplane) in two cases: the first simulation uses nose probe, the second one uses LIDAR; MPC-LIDAR utilizes smoother and smaller surface deflection than the case of MPC using the nose probe only. MPC stabilizes the aircraft at the reference altitude and velocity. MPC-LIDAR reduces significantly the vertical load factor.

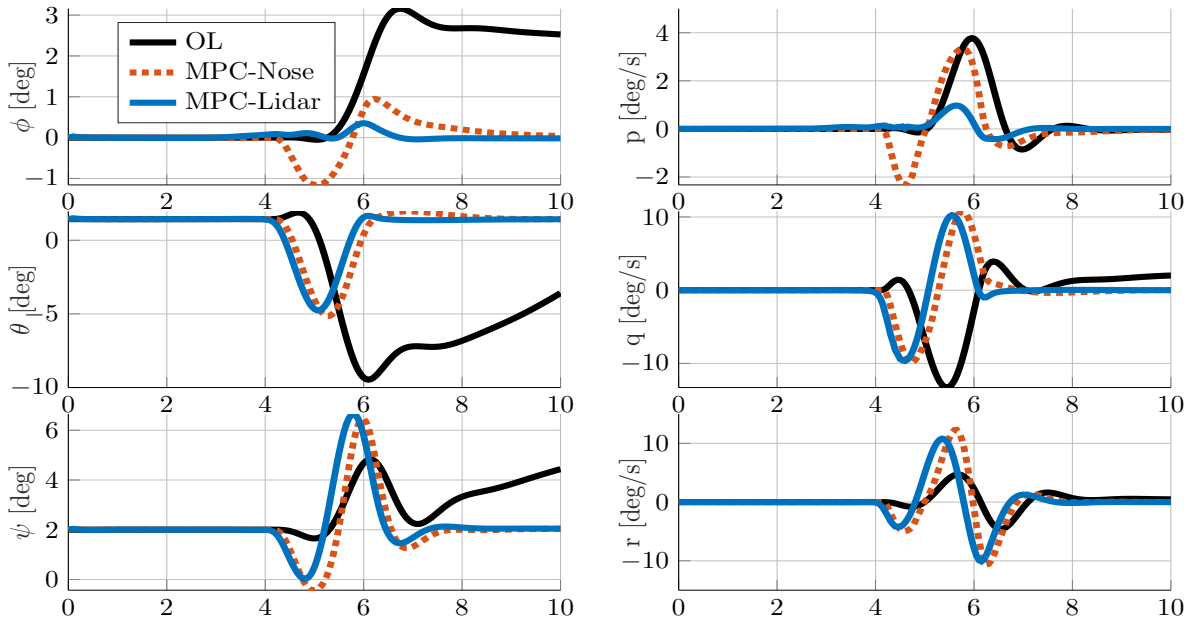


Figure 4.11: Numerical results for the second case study (Remos-GX airplane): Angles: roll ϕ , pitch θ , and yaw ψ . Angular rates: roll r , pitch q , and yaw r . MPC stabilizes the aircraft at the trim angle condition with very small rotation. MPC-LIDAR reduces significantly the roll rotation.

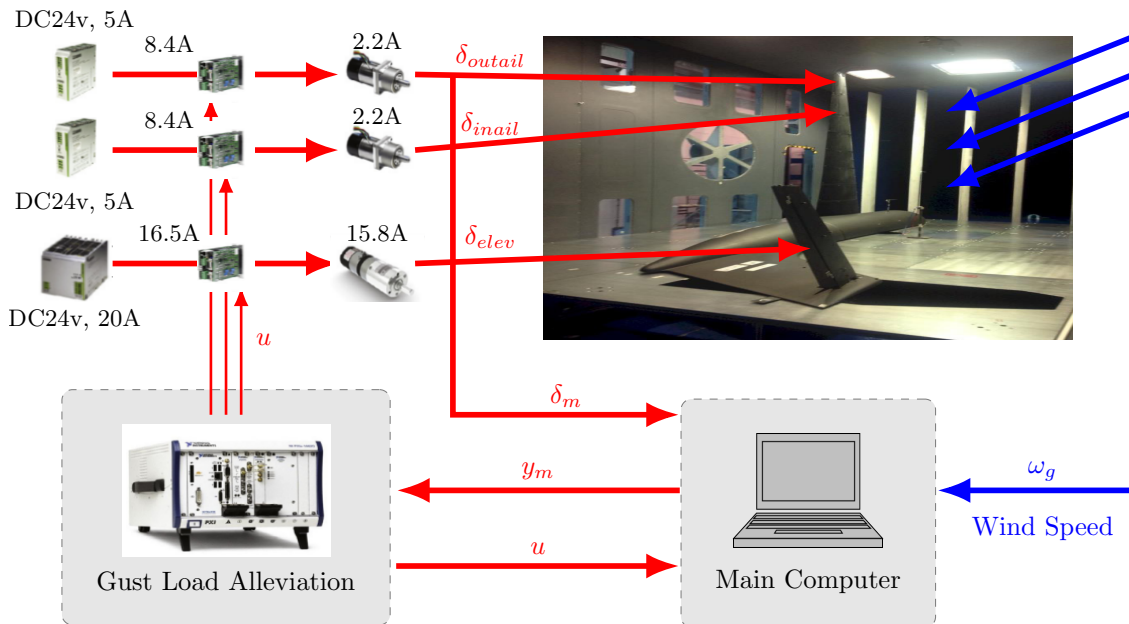


Figure 4.12: Wind Tunnel Experiment Layout: includes one PC, electronics, and four power supplies, which are used to drive the actuation system of the three control surfaces, i.e., two for the inner and outer ailerons and one for the elevator. The gust generator composed of six vanes actuated by electric motors, adapted from [141, 161].

emulated as knowledge of a predetermined time-series of the gust generator.

The WT is equipped with different sensors (c.f. Fig. 4.4, Fig. 4.12), including a set of piezoelectric accelerometers on the wing, the fuselage, and the horizontal tailplane, to measure the vertical accelerations and displacements [141, 157]. The AoA vane on the nose provides both gust and pitch orientation components, i.e., angle and rate. Furthermore, strain gauges are used to measure structural loads and two potentiometers to measure the plunge and pitch motion. The lift load is measured by a dedicated load cell, while the photogrammetric system is used to measure the airframe displacement [141]. The motor position is measured using an embedded encoder inside the motor, while a potentiometer measures the aileron position [141, 157].

4.4.1 Wind-tunnel Results

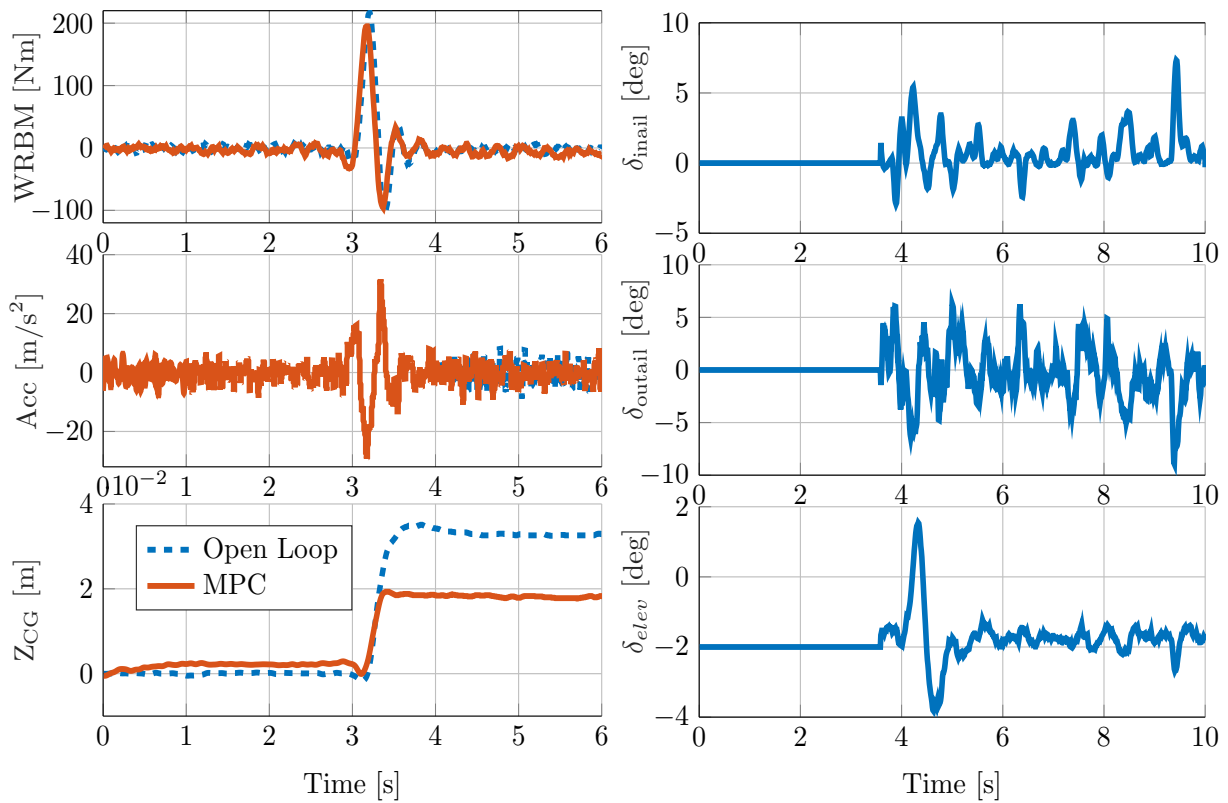
A series of WT tests had been conducted with two flow speeds, $V_\infty = 27.44$ [m/s] and $V_\infty = 34.3$ [m/s], and two gust velocities, $V_G = \pm 2.03$ [m/s]. Table 4.1 summarizes the comparison between the experimental and numerical simulation values in terms of the reduction of the wing-root bending moment (WRBM). The maximum achievable reduction is 35.87% in the numerical simulation and 18.08% in the WT experiments.

Table 4.1: Results for wing root bending moment [N m]

	V_∞ V_G		Numerical			Experimental		
	[m/s]	[m/s]	Open-Loop	Close-Loop	Reduction	Open-Loop	Close-Loop	Reduction
1	27.44	+2.03	207.69	133.21	35.86%	217.59	195.07	10.34%
2	27.44	-2.03	207.69	133.21	35.86%	219.05	179.44	18.08%
3	34.3	+2.03	166.15	106.55	35.87%	184.46	160.72	12.87%
4	34.3	-2.03	166.15	106.55	35.87%	188.99	154.99	17.98%

Fig. 4.13 depicts the experimental results at $V_\infty = 27.44$ [m/s] and upward gust $V_G = 2.03$ [m/s]. The real-time capability of the MPC approach using μ AO-MPC was able to reduce the dynamic loads acting on the wing and other structure positions. Moreover, the proposed MPC/GLA also reduces vertical acceleration, which is related to passenger comfort. The aircraft altitude is also taken into account inside the objective function to guarantee a stable flight, see Fig. 4.13. The potential for load alleviation is very promising, even if the gain is smaller than the values predicted in the simulation.

As expected, the computation time of the μ AO-MPC code using the WT facilities is 6 [ms], see Fig. 4.6d, which is within the same range of the time-to-solve during the numerical simulation, 0.45 [ms], and PIL setup 2.5 [ms]. This gives us an indication of the real-time capability of the μ AO-MPC code to implement MPC/GLA on this limited computational power devices. The mean time-to-solve is less than the underlying sample time; there is also enough a margin left to accommodate state-estimation.



(a) Wing-Root bending Moment, Acceleration at Wing Tail, Altitude. (b) Control Commands: inner aileron δ_{inail} , outer aileron δ_{outail} , and elevator δ_{elev} .

Figure 4.13: Experimental wind-tunnel results.

For safety requirements inside the WT chamber, we consider more weighting on the actuator commands, see Fig. 4.13b, to prevent abrupt control action, even the control rate limitations are taken into account.

These experimental results depend on many factors, e.g., the quality of the numerical model, estimator, and measurements. Better performance might have been achieved by retuning of the weighting matrices in the MPC optimization problem.

4.5 Discussion and Conclusion

The main contributions of this chapter are the design, real-time implementation, and experimental validation of MPC to alleviate the atmospheric loads acting on the passenger aircraft. This work demonstrated how MPC can solve the GLA problem. As shown, MPC allows considering the preview information of the turbulence measurements, e.g., using LIDAR or onboard AoA probe. The effectiveness of the proposed GLA/MPC approach was shown using simulations. The WT experiments demonstrated the real-time capability of the proposed MPC scheme to reduce the loads and to increase the handling qualities, e.g., passenger comfort.

5 Moving Horizon Path Planning

This chapter is organized as follows. Section 5.1 outlines the motivation and provides a background survey of common planning approaches. Section 5.2 outlines the proposed planning algorithm based on MILP formulation to handle non-convex constraints, e.g., obstacle avoidance. This approach is extended to solve the area coverage problem using in Section 5.4. Section 5.5 emphasizes the efficiency and applicability of the proposed approaches through several simulation results, followed by experimental results.

5.1 Introduction and Motivation

Autonomous vehicles are used in a wide range of academic and commercial applications, e.g., for traffic and accident surveillance, or search-and-rescue missions to scan dull, dangerous, or dynamic environments [12–14]. They are furthermore used in case of emergency response and disaster relief, such as fire monitoring and management [12]. Autonomous vehicles, equipped with sensors, have been used to gather scientific data, e.g., for weather forecast, map generation [15], or marine habitat mapping [162]. Moreover, they are used for cleaning and monitoring [15], and for crop management, harvesting, and fertilizers [163]. Autonomous technology improves the efficiency and environmental impact of aviation applications by reducing the operating costs, time, and energy usage [6, 48].

Autonomous vehicles can be classified into ground, underwater, or aerial vehicles [7]. We focus on multi-rotors UAVs, e.g., quadcopter. Contrary to a fixed-wing UAV, multi-rotors UAVs are capable of hovering at a fixed position and vertical takeoff and landing, eliminating the need for a runway. Further advantages of quadcopters are efficient structural design, e.g., mechanical simplicity, robustness, and being easily controllable [6]. One of the control problems in autonomous vehicle operation is the planning of a suitable path. We will develop a moving-horizon path planning approach for task planning and area coverage.

5.1.1 Motion Planning State of the Art

This section provides a brief review of the common planning approaches. For a comprehensive review, the reader is referred to [6, 8, 31, 37, 164]. The planning algorithms are mainly categorized as global or local; optimized or heuristic algorithms; with or without differential constraints; and deliberative or reactive approaches [6, 8, 31, 37, 164].

Global approaches generate a complete path at once. They often assume a static environment [6, 47]. *Local* planning algorithms generate/update the path online in response to environmental dynamics. These algorithms use local sensor information or maps acquired during the mission [6, 47]. These methods are often tailored to be implemented in real-time using low computational power. However, these approaches might get trapped in a situation instead of global objectives. The combination of global and local planning functions ensures safe and effective planning through static or dynamic environments.

Various motion planning techniques have been devised. For instance, *graph-search methods*, e.g., road map and Dijkstra’s algorithm, represent the path as a state sequence, e.g., nodes/waypoints in the configuration space [165]. *Sampling-based approaches* discretize the state and input space, e.g., lattices, which contain a library of steady-states and transient trajectories connecting two steady-states [31, 166]. This simplifies the planning and often only requires low computational time combining a finite number of quantized motion primitives. Plans can be constructed using the motion primitive library by, e.g., *particle swarm optimization* [167] or *greedy search* algorithms [168]. In the same concept, *heuristic search algorithms* have been proposed, e.g., A^* , D^* , and genetic algorithms [6, 169], to determine collision-free but suboptimal trajectories at a low computational cost. One drawback of these approaches is that they typically require a large amount of memory [6].

Potential field approaches produce a feasible path over the so-called collision-free space [6]. The main idea is to construct the potential field as the sum of an attractive force to the destination and a repulsive force to avoid obstacles [170]. These approaches are characterized by a low computational complexity, which is applicable for real-time implementation [171]. However, these techniques can get trapped in local minima and imprecise description of the obstacle’s shape and dimensions [6].

Neural networks have been used to navigate and cover entirely unstructured/clutter environments [169], so they are suitable for map building and learning [15].

For robotic applications, randomized planning approaches have been developed [163]. For instance, *rapidly-exploring random trees* produce an optimal path by a random sampling of the configuration space [6, 8].

Most planning approaches do not directly consider the vehicle dynamics in the planning phase, which can lead to decrease performance or infeasibility. This issue has grown in importance, especially for fast dynamics and in changing environments [8]. We propose a model-based planning algorithm that considers the vehicle dynamics and constraints, e.g., maneuverability, velocity, and acceleration [32–34].

Generally, optimization-based planning techniques have become common for real-time applications, facilitated by recent advancements in computational power and numerical algorithms. These methods determine the optimal path/trajectory by solving constrained OCP taking into account vehicle dynamics and collision avoidance [57].

5.2 Mixed Integer Programming Moving-Horizon Planning

In many applications, it is challenging to represent the operating environment in many applications, see Fig. 5.1, where the vehicle's knowledge might be incomplete/uncertain at the mission start. Typically, new information becomes available as the vehicles enter the operation region, i.e., the situational awareness change [6, 57]. For this reason, we propose to solve a reactive planning problem in a receding horizon scheme, i.e., to generate/update the path online based on the latest information about the environment dynamics to achieve the avoidance requirements, see [32–35]. As a result, the proposed approach reduces the computational complexity by generating only a part of the overall path.

We formulate the planning problem as a MIP that provides a framework for formulating and solving the planning and avoidance problems using discrete decisions and continuous variables, e.g., [32–34, 36, 57–59].

The main idea, see Fig. 5.1, is to design a moving-horizon planning approach to achieve the mission objective and avoiding obstacles. The resulting receding-horizon planning problem is represented as MILP:

$$\min_{x,u,z_d} J(x, u, z_d) \quad (5.1a)$$

$$\text{s.t. } \forall j \in \{0, \dots, N\}$$

$$x(k+j+1|k) = Ax(k+j|k) + Bu(k+j|k), \quad (5.1b)$$

$$x(k+j|k) \in \mathbb{X}(k+j|k) \subset \mathbb{R}^{n_x}, \quad (5.1c)$$

$$u(k+j|k) \in \mathbb{U}(k+j|k) \subset \mathbb{R}^{n_u}, \quad (5.1d)$$

$$r(k+j|k) \notin \mathbb{O}(k+j|k), \quad (5.1e)$$

$$x(k+N|k) \in \mathbb{X}_T. \quad (5.1f)$$

The vehicle dynamics (5.1b) and the constraints (5.1c,5.1d) capture the kinematic and dynamic characteristics. Note that the constraint sets $\mathbb{X}(j)$ and $\mathbb{U}(j)$ can be time-dependent. The constraints (5.1e) represent obstacles, modeled by a set of convex, compact polytopes (denoted by $\mathbb{O}(j)$). The terminal constraint (5.1f) is used to guarantee feasibility.

MIP formulation (5.1) contains continuous variables and discrete decisions. The discrete variables $z_d \in \mathbb{Z}^{n_z}$ are binary values, i.e., $z_d \in \{0, 1\}$. They are used to formulate task assignment problems, waypoint selection, and for collision avoidance [32, 33].

At each time step, the moving-horizon planner generates the optimal path by solving the optimization problem (5.1) online via minimizing the objective function $J(x, u, z)$. Herein the vehicle model (5.1b) is used to predict future behavior over the planning horizon, taking into account the predicted behavior of the environment, i.e., moving obstacles, see Fig. 5.1. Only the first path segment of the generated path is imple-

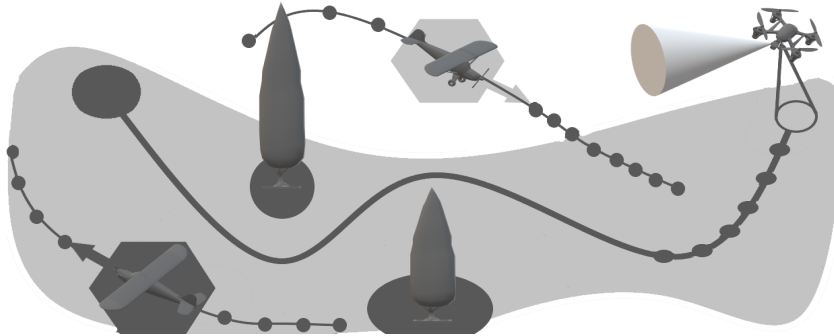


Figure 5.1: Moving horizon planning in a populated environment to achieve the mission objective and avoiding static and moving obstacles.

mented. The optimization process is then repeated for the current state.

However, using a finite-horizon may lead to stability and feasibility problems, which can be handled by designing proper terminal costs and constraints [172]. In the following, we comment on parts of the optimization problem (5.1) and outline possible choices to achieve real-time implementation and improve performance.

5.2.1 Vehicle Dynamics

Contrary to many existing planning strategies, the proposed approach takes the vehicle dynamics into account. They prevent the planner from suggesting the lower-level controller a path that might not be physically feasible. There is always a trade-off between the computational complexity and the model fidelity, i.e., capturing more detailed dynamics. We approximate the vehicle dynamics via a double integrator discrete-time model

$$\begin{bmatrix} r(j+1) \\ v_r(j+1) \end{bmatrix} = \begin{bmatrix} \mathbf{I}_2 & T_p \mathbf{I}_2 \\ \mathbf{O}_2 & \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} r(j) \\ v_r(j) \end{bmatrix} + \begin{bmatrix} \frac{T_p^2}{2} \mathbf{I}_2 \\ T_p \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} a_n \\ a_e \end{bmatrix}. \quad (5.2)$$

Here, we consider for simplicity of presentation only 2D motion planning problems. As a result, the state vector becomes $x = [r \ v_r]^\top$ describing the position $r = [p_n \ p_e]^\top$ and the velocity $v_r = [v_n \ v_e]^\top$. We furthermore use a point mass abstraction of the dynamics, then the inputs become $u = [a_n \ a_e]^\top$, i.e., the acceleration commands. \mathbf{O}_2 and \mathbf{I}_2 represent zero and identity matrices of size 2×2 , respectively.

5.2.2 Vehicle Constraints

The vehicle capabilities are taken into account in form of constraints (5.1c,5.1d). These constraints specify the range of operating conditions, by imposing max/min limits on

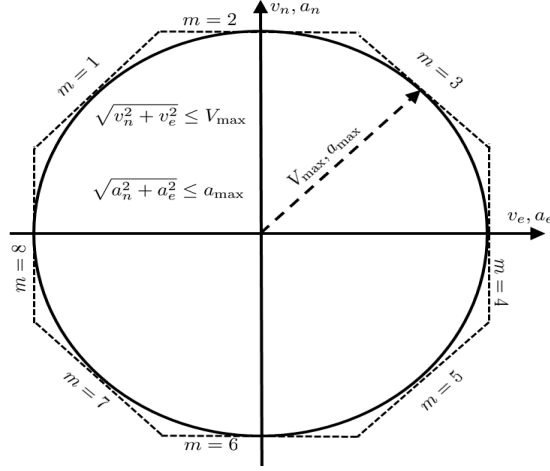


Figure 5.2: Vehicle limitations (velocity V_{\max} and acceleration a_{\max}) are approximated/upper-bounded by M -sided polygons using set of $M = 8$ linear inequalities.

the vehicle velocity and acceleration:

$$\forall j \in \{1, \dots, N\}, \quad \sqrt{v_n^2(j) + v_e^2(j)} \leq V_{\max}, \quad \sqrt{a_n^2(j) + a_e^2(j)} \leq a_{\max}.$$

We reformulate these nonlinear constraints by M -sided polygons, see Fig. 5.2:

$$\forall m \in \{1, \dots, M\}, \forall j \in \{1, \dots, N\}$$

$$v_n(j) \cos\left(\frac{2\pi m}{M}\right) + v_e(j) \sin\left(\frac{2\pi m}{M}\right) \leq V_{\max}, \quad (5.3a)$$

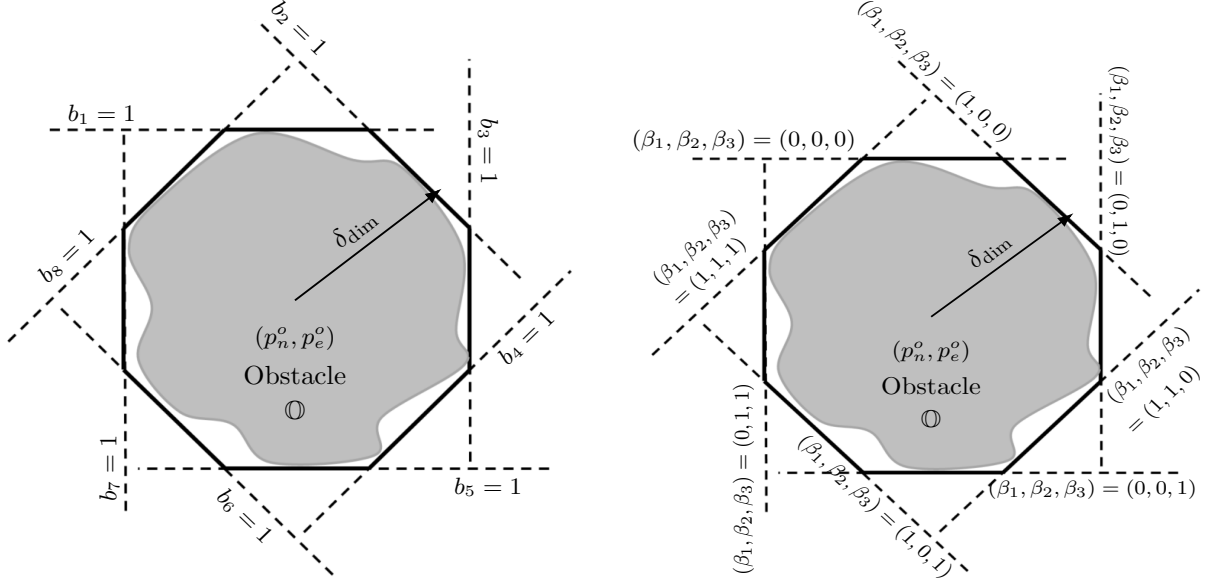
$$a_n(j) \cos\left(\frac{2\pi m}{M}\right) + a_e(j) \sin\left(\frac{2\pi m}{M}\right) \leq a_{\max}. \quad (5.3b)$$

As $M \rightarrow \infty$, we approximate the actual nonlinear constraints more accurately. However, more linear constraints increase the computation complexity [32].

Dynamic constraints on variables such as the maximum acceleration change Δa_{\max} per time step T_p are included to avoid changes in the acceleration, which are physically not possible,

$$\forall j \in \{1, \dots, N\} \quad \begin{aligned} a_n(j) - a_n(j-1) &\leq T_p \Delta a_{\max}, & -a_n(j) + a_n(j-1) &\leq T_p \Delta a_{\max}, \\ a_e(j) - a_e(j-1) &\leq T_p \Delta a_{\max}, & -a_e(j) + a_e(j-1) &\leq T_p \Delta a_{\max}. \end{aligned}$$

Furthermore, *turning rate constraints* are also taken into account in the planning problem, especially in the case of UAV. There is a relation between the maximum velocity and acceleration limits, i.e., $\Omega_{\max} = a_{\max}/v_{\max}$ [36].


 (a) 8 binary variables for $M = 8$ sides.

 (b) $N_\beta = 3$ binary variables for $2^{N_\beta} = 8$ sides.

Figure 5.3: Reformulation of obstacles via binary variables: $N_\beta = 3$ binary variables for $2^{N_\beta} = 8$ polygon sides are used, see Fig 5.3b. Comparing to the formulation, using eight binary variables, see Fig 5.3a.

5.2.3 Collision and Obstacles Avoidance Constraints

MILP formulation overbounded/relaxed the obstacles \odot by polygons introducing binary variables $b_m^o(k)$, see Fig. 5.3 [13, 60, 61, 63, 173]:

$$\forall m \in \{1, \dots, M\}, \forall o \in \{1, \dots, N_o\}, \forall j \in \{1, \dots, N\}$$

$$(p_n(j) - p_n^o(j)) \cos \frac{2\pi m}{M} + (p_e(j) - p_e^o(j)) \sin \frac{2\pi m}{M} \geq \delta_d^o + \delta_{\text{safe}} - M_{\text{big}} b_m^o(j), \quad (5.4a)$$

$$\sum_{m=1}^M b_m^o(j) \leq M - 1, \quad (5.4b)$$

Here N_o is the number of (static/dynamic) obstacles, and δ_d^o represents the vehicle size added to every obstacle with the coordinates (p_n^o, p_e^o) . The binary variables $b_m^o(j) \in \{0, 1\}$ are used to activate or relax the constraints (5.4a) using the so-called “big M” method. Here, M_{big} is a large positive number [32–34, 36]. The constraint (5.4b) ensures that at least one constraint is active, see Fig. 5.3. These constraints, if feasible, ensure a collision-free path, i.e., the vehicle does not touch any obstacle. These constraints can be easily extended to the 3D obstacle shapes [57, 60].

Obstacle Enlargement

We considered a continuous dynamics. Therefore, the generated path in continuous-time might cut through the obstacles corners between two sampling-times. To

avoid this, we enlarge the obstacle dimensions by a safety margin $\delta_{\text{safe}} = 0.5T_p V_{\text{max}} \sin(\pi/4)$ [13, 32, 33, 36, 61]. Note that the safety margin depends on the vehicle dynamics (e.g., V_{max}) and the discretization's sampling time T_p .

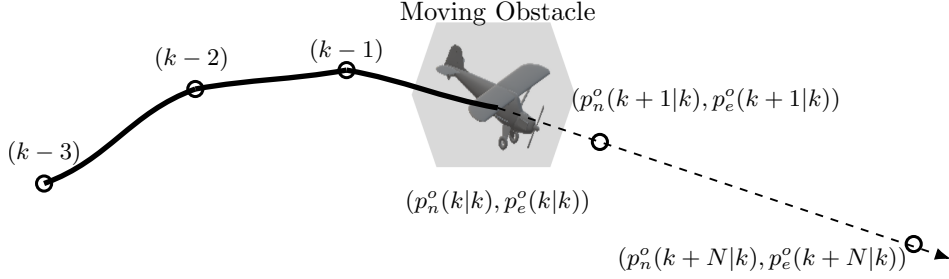


Figure 5.4: Prediction of moving obstacle position.

Representing dynamic obstacles

We consider moving obstacles with a position $(p_e^o(k+j|k), p_n^o(k+j|k))$, predicted over the planning horizon, see Fig. 5.4 by a linear dynamics:

$$\forall o \in \{1, \dots, N_m\}, \forall j \in \{0, \dots, N-1\},$$

$$p_e^o(k+j+1|k) = p_e^o(k+j|k) + T_p v_e^o(k), \quad (5.5a)$$

$$p_n^o(k+j+1|k) = p_n^o(k+j|k) + T_p v_n^o(k). \quad (5.5b)$$

We assume a constant speed $(v_e^o(k), v_n^o(k))$.

Complexity reduction reformulation:

Introducing binary variables $b_m^o(k)$ in (5.4a) leads to an increase in the problem complexity. To reduce the solution time, we propose to reformulate the obstacle avoidance constraints (5.4a) via a suitable representation:

$$\forall m \in \{1, \dots, M\}, \forall o \in \{1, \dots, N_o\}, \forall j \in \{1, \dots, N\}$$

$$(p_n(j) - p_n^o) \cos \frac{2\pi m}{M} + (p_e(j) - p_e^o) \sin \frac{2\pi m}{M} \geq \delta_s - M_{\text{big}} \left(\sigma_0^m + \sum_{s=1}^{N_\beta} \sigma_s^m \beta_s^o(j) \right) \quad (5.6)$$

To do so, we extend the binary encoding [173] by introducing binary variables $(\beta_s^o) \in \{0, 1\}$ with coefficients (σ_0^m, σ_s^m) :

$$\forall s \in \{1, \dots, N_\beta\}, \forall m \in \{1, \dots, M\}$$

$$\sigma_s^m = -(-1)^{\left\lceil \left(\frac{m}{2^{(s-1)}} \right) \right\rceil}, \quad \sigma_0^m = 0.5 \sum_{s=1}^{N_\beta} ((\sigma_s^m)^2 - \sigma_s^m).$$

Here, the operator $\lceil a \rceil$ determines the smallest integer greater than or equal to a .

For example for one obstacle $o = 1$ in Fig. 5.3b, we use $N_\beta = 3$ binary variables for

$2^{N_\beta} = 8$ polygon sides, whereas in [13, 36, 60, 61, 63, 173], eight binary variables are used, see Fig. 5.3a.

Remark *Considering eight obstacles $N_o = 8$, and a prediction horizon $N = 25$, a total reduction in the binary variables of $N_o \times N \times (8 - 3) = 8 \times 25 \times 5 = 1000$ is achieved. This allows reducing the computational time significantly.*

5.2.4 Objective Functions

Basically, any cost function can be considered, allowing formulating different mission objectives. One particular choice is to minimize both the energy consumption and the distance to the target, e.g., by choosing a (quadratic) cost:

$$J_p = \sum_{j=1}^{N-1} W_u \|u(j)\| + W_d \|x_{\text{Target}} - x(N)\|.$$

Here, $D_{\text{Target}} = \|x_{\text{Target}} - x(N)\|$ penalizes the distance to the target point x_{Target} at the end of the planning horizon, while the stage cost minimizes the control input $\|u(j)\| = \sqrt{a_n^2(j) + a_e^2(j)}$, i.e. the energy.

For computational reasons, one can aim to replace the quadratic cost by a linear problem via introducing “slack variables” s_n, s_e , and D_{Target} [13, 32, 33, 63]:

$$J_p = W_u \sum_{j=1}^N (s_n(j) + s_e(j)) + W_d D_{\text{Target}}(N).$$

Here s_n and s_e are upper bounds of the magnitude of a_n and a_e , respectively, given by set of linear constraints $\forall j \in \{1, \dots, N\}$

$$a_n(j) \leq s_n(j), \quad -a_n(j) \leq s_n(j), \quad a_e(j) \leq s_e(j), \quad -a_e(j) \leq s_e(j),$$

while the distance-to-target D_{Target} is given as [13, 33, 63]:

$$\begin{aligned} \forall j \in \{1, \dots, N\}, \quad \forall m \in \{1, \dots, M\}, \\ (p_n(j) - p_n^{\text{Target}}) \cos \frac{2\pi m}{M} + (p_e(j) - p_e^{\text{Target}}) \sin \frac{2\pi m}{M} \leq D_{\text{Target}}(j) + \delta_{\text{th}}. \end{aligned} \quad (5.7)$$

Note that tuning the weights W_u, W_d significantly impacts the control performance. Increasing W_u lays a higher weight on the energy minimization leading to less acceleration and deceleration, i.e., to a more uniform traveling speed with less (sharp) turns. Therefore, it improves the efficiency of the generated path and makes it is easier to be followed. In general, tuning the weights W_u, W_d requires considering many factors, e.g., the region area, number and size of obstacles, and the vehicle capabilities.

5.3 Robust Moving-Horizon planning

In practical applications, the vehicle might deviate from its expected trajectory due to uncertainties and external disturbances. Neglecting the uncertainties is critical, especially in cluttered and dynamic environments resulting in performance degradation and vehicle instability or complete failure [6, 8, 31, 37]. To overcome these challenges, robust planning approaches are needed to obtain safe and plausible references to satisfy constraints and improve performance.

In real missions, autonomous vehicles are often exposed to different uncertainties, e.g., wind disturbances, model mismatch, and measurement noise. Additionally, the surrounding environments might be highly uncertain or subject to limited information due to inadequate sensing capabilities [6, 8, 17, 31, 37]. Therefore, robust planning approaches have been proposed, see, e.g., [6, 8, 31, 37, 174]. A common approach is to treat the uncertainties as a deterministic worst-case by introducing a conservative safety corridor with suitable risk thresholds [6, 62]. We propose robust planning using

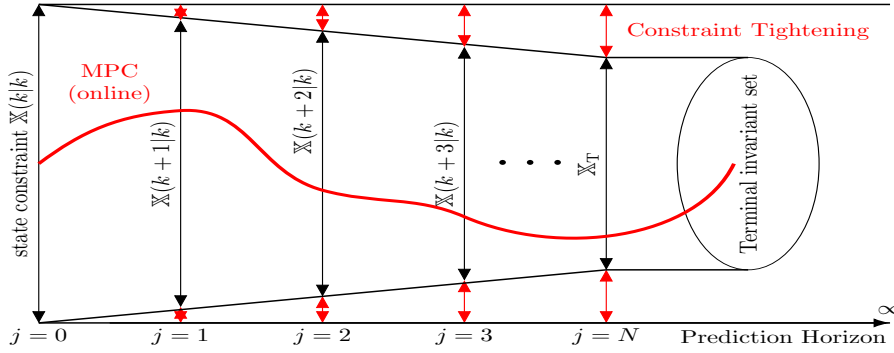


Figure 5.5: Illustration of constraint tightening.

a constraint tightening approach to a guarantee to satisfy the original constraints despite unknown but bounded uncertainties.

The constraint tightening approach is based on solving a nominal open-loop optimization problem. It guarantees robustness by “shrinking” the vehicle constraints sets, i.e., state \mathbb{X} , input \mathbb{U} , and terminal set \mathbb{X}_T , see Fig 5.5. This allows satisfying the original constraints despite uncertainties, which can be rejected using a future feedback correction, e.g., in [57, 83]. As a result, encountered obstacles (5.4b) are enlarged by a safety bound to provide online a collision-free reference.

For simplicity, we focus on linear dynamics subject to additive disturbances:

$$x(k+1) = Ax(k) + Bu(k) + \omega(k), \quad (5.8a)$$

$$x(k) \in \mathbb{X}, \quad u(k) \in \mathbb{U}, \quad (5.8b)$$

$$\omega(k) \in \mathbb{W}. \quad (5.8c)$$

Here $\omega(k)$ is an unknown, but bounded disturbance, i.e., $\mathbb{W} \subset \mathbb{R}^n$ is convex and compact uncertainty set containing the origin in its interior.

We use the Minkowski set sum \oplus and the Pontryagin set difference \ominus . Given two sets $\mathbb{X}, \mathbb{Y} \subset \mathbb{R}^n$, they are defined by:

$$\begin{aligned}\mathbb{X} \oplus \mathbb{Y} &\triangleq \{x + y | x \in \mathbb{X}, y \in \mathbb{Y}\}, \\ \mathbb{X} \ominus \mathbb{Y} &\triangleq \{x | x \oplus \mathbb{Y} \subseteq \mathbb{X}\}.\end{aligned}$$

The effect of the disturbance is counteracted by restricting the constraints (5.3,5.4b) to the worst-case under a disturbance rejection controller K :

$$\mathbb{X}(k + j + 1|k) = \mathbb{X}(k + j|k) \ominus L(j)\mathbb{W}, \quad (5.9a)$$

$$\mathbb{U}(k + j + 1|k) = \mathbb{U}(k + j|k) \ominus K(j)L(j)\mathbb{W}, \quad (5.9b)$$

$$\mathbb{O}(k + j + 1|k) = \mathbb{O}(k + j|k) \oplus L(j)\mathbb{W}, \quad (5.9c)$$

$$\mathbb{X}_T = \mathbb{R}_{CT} \ominus L(N - 1)\mathbb{W}, \quad (5.9d)$$

$L(j)$ denotes a state transition matrix for the closed-loop system under the control law $u(j) = K(j)x(j)$:

$$L(j + 1) = (A + BK(j))L(j), \quad L(0) = I. \quad (5.10)$$

Here, K is a stabilizing feedback law designed using, e.g., an LQR or nilpotent method [59, 83]. This feedback policy is used to determine the margin for constraint tightening in expressions (5.9), with feasibility guarantee (constraints satisfaction), for any admissible uncertainties sequence. The safety set $\mathbb{R}_{CT} \subset \mathbb{X}$ is a robust control invariant admissible set that has the following property:

$$Ax + B\kappa_T(x) + L(N - 1)\omega \in \mathbb{R}_{CT}, \quad (5.11a)$$

$$x \in \mathbb{X} \ominus L(N - 1)\mathbb{W}, \quad (5.11b)$$

$$\kappa_T(x) \in \mathbb{U} \ominus K(N - 1)L(N - 1)\mathbb{W}, \quad (5.11c)$$

$$r \notin \mathbb{O}_i \oplus CL(N - 1)\mathbb{W}. \quad (5.11d)$$

Note that if the state is inside the safety set, the vehicle can remain safe despite the uncertainty, i.e., satisfying the vehicle constraints (e.g., obstacle avoidance) using the terminal control law κ_T .

The resulting robust planning MILP using the constraint tightening is defined as:

$$\begin{aligned} & \min_{x,u} J(x,u) & (5.12a) \\ \text{s.t. } & \forall j \in \{0, \dots, N\} \\ & x(k+j+1|k) = Ax(k+j|k) + Bu(k+j|k), & (5.12b) \\ & x(k+j|k) \in \mathbb{X}(k+j|k), & (5.12c) \\ & u(k+j|k) \in \mathbb{U}(k+j|k), & (5.12d) \\ & r(k+j|k) \notin \mathbb{O}(k+j|k), & (5.12e) \\ & x(N) \in \mathbb{X}_T. & (5.12f) \end{aligned}$$

Using this planning formulation, we can provide constraint satisfaction if the following assumptions hold:

Assumption 5.1 (*Controllability*) The linear system $\{A; B\}$ in (5.8), is controllable.

Assumption 5.2 (*Full state information*) The full state $x(k)$ is available at each time k .

Assumption 5.3 (*Bounded disturbance*) \mathbb{W} in (5.8) is a convex and compact set, which contains the origin in its interior.

Assumption 5.4 (*Terminal set*) There exists an admissible terminal set $\mathbb{X}_T \subset \mathbb{X}$, which is positively invariant under a terminal control law $\kappa_T(x) \in \mathbb{U}$, i.e., $\forall x \in \mathbb{X}_T$ satisfying:

$$Ax + B\kappa_T(x) \in \mathbb{X}_T, \quad (5.13a)$$

$$x \in \mathbb{X} \ominus L(j)\mathbb{W}, \quad (5.13b)$$

$$\kappa_T(x) \in \mathbb{U} \ominus K(j)L(j)\mathbb{W}, \quad (5.13c)$$

$$r \notin \mathbb{O}_i \oplus CL(j)\mathbb{W}. \quad (5.13d)$$

Note that if the vehicle enters the terminal set, the generated path will remain feasible (i.e., obstacle-free) despite the uncertainty, i.e., satisfying the vehicle constraints using the terminal control law κ_T .

Proposition 5.1 (*Feasibility*) Let Assumptions 5.1-5.4 hold. If the moving-horizon planning problem (5.12) is initially feasible at time k_0 , with a feasible solution

$$\begin{aligned} & \{x^*(k_0+1|k_0), x^*(k_0+2|k_0), \dots, x^*(k_0+N|k_0)\} \\ & \{u^*(k_0|k_0), u^*(k_0+1|k_0), \dots, u^*(k_0+N-1|k_0)\} \end{aligned}$$

then for all disturbances $w \in \mathbb{W}$, the robust planning problem (5.12) is feasible at time $k_1 = k_0 + 1$.

Proof To verify the recursive feasibility for the optimization problem (5.12) at $k_1 = k_0 + 1$, we consider the following initial candidate solution based on the previous solution and the terminal control law κ_T :

$$\tilde{u}(k_1 + j|k_1) = u^*(k_0 + j + 1|k_0) + K(j)L(j)w(k_0), \quad j = 1, \dots, N, \quad (5.14a)$$

$$\tilde{u}(k_1 + N|k_1) = \kappa_T(x^*(k_0 + N|k)), \quad (5.14b)$$

$$\tilde{x}(k_1 + j|k_1) = x^*(k_0 + j + 1|k_0) + L(j)w(k_0), \quad j = 1, \dots, N, \quad (5.14c)$$

$$\tilde{x}(k_1 + N + 1|k_1) = A_p x^*(k + N|k) + B\kappa_T(x^*(k_0 + N|k)). \quad (5.14d)$$

One can verify straightforwardly that this initial guess (5.14) is feasible (but suboptimal). Hence, it must satisfy all constraints in (5.12) at time step $k_1 = k_0 + 1$ despite of any disturbance $w \in \mathbb{W}$ using the properties of the terminal set \mathbb{X}_T .

Dynamics constraints:

The previous plan satisfied the vehicle dynamics (5.12b), i.e. implying:

$$\begin{aligned} \forall j \in \{0, \dots, N - 1\}, \\ x^*(k_0 + j + 2|k_0) = Ax^*(k_0 + j + 1|k_0) + Bu^*(k_0 + j + 1|k_0) \end{aligned} \quad (5.15)$$

Substituting on the both sides for x^* and u^* from the candidate solution (5.14):

$$\begin{aligned} & \left(\tilde{x}(k_1 + j + 1|k_1) - L(j + 1)w(k_0) \right) \\ &= A \left(\tilde{x}(k_1 + j|k_1) - L(j)w(k_0) \right) + B \left(\tilde{u}(k_1 + j|k_1) - K(j)L(j)w(k_0) \right) \\ &= A\tilde{x}(k_1 + j|k_1) + B\tilde{u}(k_1 + j|k_1) - (A + BK(j))L(j)w(k_0) \end{aligned}$$

Substituting the state transition matrices L from (5.10), subject to $\forall j \in \{0, \dots, N\}$:

$$\tilde{x}(k_1 + j + 1|k_1) = A\tilde{x}(k_1 + j|k_1) + B\tilde{u}(k_1 + j|k_1) \quad (5.17)$$

we get identical to the dynamics constraint (5.12b) using the candidate solution (5.14).

Initial state constraint:

The vehicle state at time $k_0 + 1$ is found by applying control $u^*(k_0|k_0)$ and disturbance $w(k_0)$ to the vehicle dynamics (5.12b).

$$x^*(k_0 + 1|k_0) = Ax^*(k_0|k_0) + Bu^*(k_0|k_0) + w(k_0). \quad (5.18)$$

Then subtracting from this equation, the certain dynamics (6.5b) for step $j = 0$ at time $k = k_0$:

$$x^*(k_0 + 1|k_0) = Ax^*(k_0|k_0) + Bu^*(k_0|k_0). \quad (5.19)$$

So the new state at time $k_1 = k_0 + 1$ can be expressed as a perturbation from the

planned next state

$$x(k_0 + 1|k_0) = x^*(k_0 + 1|k_0) + w(k_0). \quad (5.20)$$

Substituting $L(0) = I$ into (5.14c) with $j = 0$ gives

$$\tilde{x}(k_1|k_1) = x^*(k_0 + 1|k_0) + w(k_0). \quad (5.21)$$

Then $\tilde{x}(k_1|k_1) = x(k_0 + 1|k_0)$ satisfying the initial state condition.

Terminal constraints (5.12f):

substituting into the candidate solution (5.14c) for $j = N$ gives

$$\tilde{x}(k_1 + N + 1|k_1) = x^*(k_0 + N + 1|k_0) + L(N)w(k_0). \quad (5.22)$$

Feasibility at time k_0 requires

$$x^*(k_0 + N + 1|k_0) \in \mathbb{X}_T. \quad (5.23)$$

according to terminal constraint (5.12f), so

$$\tilde{x}(k_1 + N|k_1) \in \mathbb{R}_{CT}. \quad (5.24)$$

The invariance condition (5.9d) ensures

$$A\tilde{x}(k_1 + N|k_1) + B\kappa_T(\tilde{x}(k_1 + N|k_1)) + L(N)w \in \mathbb{R}_{CT}, \quad \forall w \in \mathbb{W},$$

which from the candidate solution (5.14d) implies $\forall w \in \mathbb{W}$

$$\tilde{x}(k_1 + N + 1|k_1) + L(N)w \in \mathbb{R}_{CT}. \quad (5.25)$$

Using the definition of the Pontryagin difference and the terminal set (5.9d) this shows

$$\tilde{x}(k_1 + N + 1|k_1) \in \mathbb{X}_T. \quad (5.26)$$

which satisfies the terminal constraint (5.12f) for time $k_1 = k_0 + 1$.

State and input constraints (5.12c,5.12d):

Feasibility at time k_0 implies $\forall j \in \{1, \dots, N\}$:

$$x^*(k_0 + j + 1|k_0) \in \mathbb{X}(k_0 + j|k_0), \quad (5.27a)$$

$$u^*(k_0 + j|k_0) \in \mathbb{U}(k_0 + j|k_0). \quad (5.27b)$$

Applying (state and input) constraint sets (5.9), and the definition of the Minkowski

difference $\mathbb{A} \ominus \mathbb{B} = \{a | a + b \in \mathbb{A} \quad \forall b \in \mathbb{B}\}$ in (5.27), with $\forall w(k_0) \in \mathbb{W}$ giving:

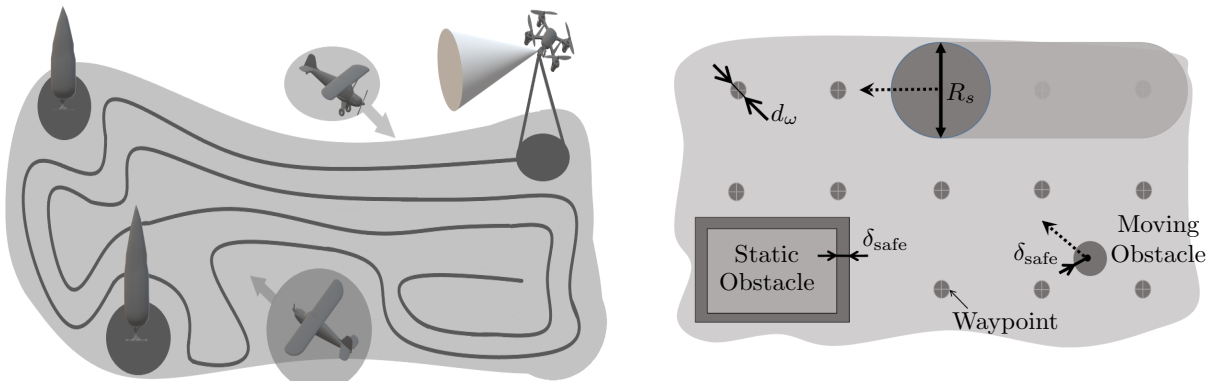
$$\begin{aligned} x^*(k_0 + j + 1 | k_0) &\in \mathbb{X}(k_0 + j + 1 | k_0) \Rightarrow \\ \tilde{x}(k_1 + j | k_1) &= x^*(k_0 + j + 1 | k_0) + L(j)w(k_0) \in \mathbb{X}(k_1 + j | k_1), \\ u^*(k_0 + j | k_0) &\in \mathbb{U}(k_0 + j | k_0) \Rightarrow \\ \tilde{u}(k_1 + j | k_1) &= u^*(k_0 + j + 1 | k_0) + K(j)L(j)w(k_0) \in \mathbb{U}(k_1 + j | k_1). \end{aligned}$$

which satisfies the state and input constraints (5.12c),(5.12d) at time $k_1 = k_0 + 1$.

Also, it follows from (5.24) and the admissibility requirement (5.12c),(5.12d),(5.12e) that the final control step using $\kappa_T(\tilde{x}(k_1 + N | k_1))$ is admissible for state set $\mathbb{X}(N)$ and input $\mathbb{U}(N)$ according to (5.12c),(5.12d),(5.12e).

This ensures that the optimization's feasibility at time $k = k_0$ implies the feasibility at the next time $k_1 = k_0 + 1$, resulting in the robust feasibility. The constraint tightening guarantees the existence of an input sequence such that the constraints are satisfied at the next time step. \blacksquare

Remark (*Online complexity*) *The disturbance feedback policy and tightened constraints are computed offline. Thereby, it does not increase the complexity of the online optimization. The decision space of the online optimization remains the same as nominal MPC, allowing for efficient real-time computation.*



(a) Area Coverage in dynamic environment. (b) Discretized representation of the search area.

Figure 5.6: Area coverage problem: An area should be covered by an autonomous vehicle, which has a sensor range R_s while avoiding static and dynamic obstacles with safety margin δ_{safe} . To do so, we divide the area in cells, each containing a waypoint with a relaxation threshold d_w , allowing deviation from these waypoints [32, 35].

5.4 Planning for Area Coverage

In this section, the planning problem is extended to allow for area coverage [32]. In recent years, area exploration/coverage [11, 174] has received increasing attention,

as it can be used for many applications. Examples are exploration and scanning in agriculture, geological research, structures inspection, forest fire detection, search and rescue missions, up to autonomous cleaning systems and mapping [12–14, 162]. The reader is referred to the comprehensive survey, e.g., [9, 11, 174, 175]. Area coverage planning aims to generate a flyable/moveable path to completely cover the region-of-interest while minimizing a cost, e.g., time and energy. The path must avoid the forbidden regions, e.g., obstacles, see Fig. 5.6a. To approach this problem, we simply decompose the region-of-interest offline into cells/waypoints, see Fig. 5.6b.

5.4.1 Implementation of search area

Practically, motion planning requires an environmental representation to define the space information. *Quantitative mapping* introduces a feasible data structure for every cell using a metric mapping, e.g., occupancy map, meadow map, target probability map, quadtree map, uncertainty map, see [8]. We use a binary representation that is applicable to MILP formulation. Each grid cell has an associated binary value.

To achieve completeness, coverage planning algorithms are often based on decomposition methods, e.g., exact/approximate cellular decomposition, Boustrophedon, triangulation, trapezoidal, Morse function, and Voronoi diagram, see [6, 8, 174–176]. These use computational geometry methods to subdivide the configuration space into a set of smaller uniform or non-uniform regions/polygons termed cells according to the sensor range [163, 169, 175]. Each cell can then be modeled as a node in an adjacency graph, in which the edges represent the boundary between two adjacent cells [170].

We use a discretized representation of the search area, which is decomposed into a simple, non-overlapping discrete map (grid or cells), each containing a waypoint, see Fig. 5.6b. The planner uses a number of waypoints N_p inside this area to reduce the number of variables, thereby the computation time. The number of waypoints and the distance between them is influenced by the area size, the sensor’s range, the desired precision, and the vehicle dynamics [13, 32]. Based on the defined waypoints, the planning algorithm finds a path that covers a maximum number of waypoints online, with minimum overlapping.

5.4.2 Formulation of the Area Coverage Planning Problem

MILP formulation is expanded by discrete decisions “scheduling” waypoints constructing the path exploiting a discrete representation of the environment [32, 33]. The movement between the waypoints is parametrized by continuous decision variables taking simplified dynamics of the autonomous vehicle and moving obstacles into account. In what follows, we will shorten the time-index $(k + j|k)$ to j to simplify the notation. The objective of the planner is to find a plausible path that minimizes the uncovered

area by solving the optimization problem:

$$\min_{x,u,\mathbf{D},\mathbf{C}} J(x(\cdot), u(\cdot), \Phi(\cdot)) \quad (5.28a)$$

$$\text{s.t. } \forall j \in \{0, \dots, N\}, \forall p \in \{1, \dots, N_p\}$$

$$x(j+1) = A(T_p)x(j) + B(T_p)u(j), \quad x(0) = x(k), \quad (5.28b)$$

$$x(j) \in \mathbb{X}(j) \ominus L(j)\mathbb{W}, \quad (5.28c)$$

$$u(j) \in \mathbb{U} \ominus K(j)L(j)\mathbb{W}, \quad (5.28d)$$

$$r(j) \notin \mathbb{O}(j) \oplus CL(j)\mathbb{W}, \quad (5.28e)$$

$$x(N) \in \mathbb{X}_T \ominus L(N)\mathbb{W}, \quad (5.28f)$$

$$\Phi^p(j+1) = \Phi^p(j) - c^p(j), \quad \Phi^p(0) = \Phi^{p^*}(k-1) \quad (5.28g)$$

$$0 \leq \Phi^p(j) \leq 1 \quad 0 \leq c^p(j) \leq 1 \quad c^p(j) \leq d^p(j) \quad (5.28h)$$

$$\|r^p - r(j)\| \leq \delta_{\text{th}} \Rightarrow d^p(j) = 1. \quad (5.28i)$$

Here, J denotes the coverage cost function, T_p is the planning sampling time, N denotes the planning horizon, and N_p is the number of the waypoints. Contrary to many existing strategies, the vehicle dynamics are accounted for by a linear discrete-time model (5.28b). The state and input constraint \mathbb{X} and \mathbb{U} in (eqs. (5.28c) and (5.28d)) represent the vehicle capabilities, e.g., maximum velocity and acceleration. This guarantees the planner to generate a physically feasible, i.e., flyable path. The obstacle avoidance constraints (5.28e) are time-dependent, allowing for moving obstacles. The terminal constraint (5.28f) is a positively invariant set under the control law $\kappa_T(\cdot)$, i.e., if $x \in \mathbb{X}_T$, then $Ax + B\kappa_T(x) \in \mathbb{X}_T$.

$\Phi^p(j)$ is the waypoint status, $d^p(j)$ is a binary decision, and $c^p(j)$ is an auxiliary input for the waypoint p (at position r^p , and time j). If the p -th cell was covered $d^p(j) = 1$, $c^p(j) = 1$, $\Phi^p(j) = 0$, else $d^p(j) = 0$, $c^p(j) = 0$, $\Phi^p(j) = 1$ [32, 58]. At the beginning $k = 0$, $\Phi^p(0) = 1$ means that all waypoints are uncovered.

5.4.3 Implementation of Multiple Waypoint Constraints

The initial state and dynamics of each cell/waypoint is given by (5.28g), i.e., updating the covering map of the environment according to the new sensor measurement. Note that each cell state Φ^p and each auxiliary input c^p are constrained between 0 and 1 by (5.28h). This provides the planner with the capability to compute a path that covers a maximum number of waypoints online. If the distance between the autonomous vehicle $r(k)$ and a waypoint r^p is smaller than a threshold δ_{th} , the binary decision variable $d^p(j)$ is set to one, see (5.28i), which labels the waypoint as being covered via the waypoint dynamics (5.28g). $\mathbf{D} \in \mathbb{R}^{N \times N_p}$ is the decision matrix of binary variables $d^p(j) \in \{0, 1\}$, which indicates whether the waypoint p is covered at time

step j ($d^p(j) = 1$) or not ($d^p(j) = 0$) over the planning horizon:

$$\mathbf{C} = \begin{bmatrix} c^1(j) & \cdots & c^{N_p}(j) \\ c^1(j+1) & \cdots & c^{N_p}(j+1) \\ \vdots & \ddots & \vdots \\ c^1(j+N) & \cdots & c^{N_p}(j+N) \end{bmatrix} \leq \mathbf{D} = \begin{bmatrix} d^1(j) & \cdots & d^{N_p}(j) \\ d^1(j+1) & \cdots & d^{N_p}(j+1) \\ \vdots & \ddots & \vdots \\ d^1(j+N) & \cdots & d^{N_p}(j+N) \end{bmatrix}.$$

Here $\mathbf{C} \in \mathbb{R}^{N \times N_p}$ is the matrix of auxiliary inputs $c^p(j)$, which are introduced as slack variables to allow the autonomous vehicle to visit a waypoint p more than once without any cost. When this happens, the optimizer chooses $d^p(j) = 1$ and $c^p(j) = 0$, so one can guarantee a feasible solution and the waypoint constraint (5.28h) is satisfied, i.e., $\Phi^p \geq 0$. Note that c^p is upper bounded by the binary decision variable d^p via (5.28h), which indicates whether the waypoint p is covered for the first time over the planning horizon at the time step j ($c^p(j) = 1$) or not ($c^p(j) = 0$).

Reformulation for computation reduction:

To reduce the computation time, we use a reformulation of binary decision variables, which leads to decreasing the number of binary variables. We use N_λ binary variables for $2^{N_\lambda} - 1$ waypoints, comparing to the formulation in [32, 58], which uses N_p binary variables d^i for N_p waypoints. c^p is upper bounded by binary decision variables λ_s , i.e., the constraint (5.28h) is modified to:

$$\forall p \in \{1, \dots, N_p\} \quad c^p(j) \leq \gamma_0^p + \sum_{s=1}^{N_\lambda} \gamma_s^p \lambda_s(j), \quad (5.29)$$

where the coefficients (γ_0^p, γ_s^p) are defined by $\forall s \in \{1, \dots, N_\lambda\}, \forall p \in \{1, \dots, N_p\}$:

$$\gamma_s^p = (-1) \left\lceil \left(\frac{p+1}{2^{(s-1)}} \right) \right\rceil, \quad \gamma_0^p = 1 - 0.5 \sum_{s=1}^{N_\lambda} \left((\gamma_s^p)^2 + \gamma_s^p \right).$$

Hereby, $\lceil a \rceil$ determines the least integer greater than or equal to a . The auxiliary decision variables $c^p(k)$ are introduced as slack variables to allow the vehicle to visit a waypoint more than once without decreasing the cost. When this happens, the optimizer chooses the combination

$$\left(\gamma_0^p + \sum_{s=1}^{N_\lambda} \gamma_s^p \lambda_s(j) \right) = 1, \quad \text{and} \quad c^p(j) = 0,$$

so one can guarantee that $\Phi^p \in \{0, 1\}$ is satisfied. For example, let $N_p = 15$, then $N_\lambda = 4$ binary variables are sufficient as for $2^{N_\lambda} - 1 = 15$ waypoints, compared to the formulation used in [32, 58], which uses 15 binary variables for 15 waypoints, see Fig. 5.7. Example: for 15 waypoints, $N_p = 15$, and a prediction horizon $N = 25$, the

total reduction in the binary variables is $N \times (N_p - N_\lambda) = 25 \times 11 = 275$.

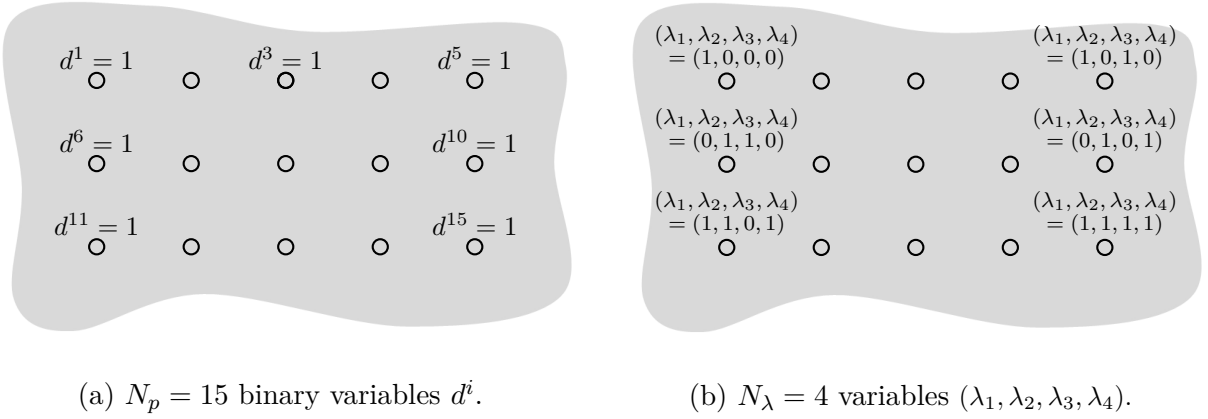


Figure 5.7: Discretized representation of the search area with $N_p = 15$ binary variables $d^p, p \in \{1, \dots, N_p\}$, see Fig.5.7a, while we use $N_\lambda = 4$ binary variables for $2^{N_\lambda} - 1 = 15$ waypoints, see Fig.5.7b.

Visiting a waypoint p is verified via the constraint

$$\|r^p - r(j)\| \leq \delta_{\text{th}} \Leftrightarrow \sqrt{(p_n^p - p_n(j))^2 + (p_e^p - p_e(j))^2} \leq \delta_{\text{th}} \Rightarrow d^p(j) = 1, \quad (5.30)$$

which presents the distance between the vehicle position $r(j)$ at time step j and a waypoint r^p should be smaller than a threshold δ_{th} . The two-norm approximation for estimating distance is very similar to distance-to-target (5.7). We present approximate the waypoints constraint (5.30) as a set of linear constraints representing M -sided polygons [32, 58] using again the big M method:

$$\forall m \in \{1, \dots, M\}, \forall p \in \{1, \dots, N_p\}, \forall j \in \{1, \dots, N\}$$

$$(p_n(j) - p_n^p) \cos \frac{2\pi m}{M} + (p_e(j) - p_e^p) \sin \frac{2\pi m}{M} \leq d_w + M_{\text{big}} \left(\alpha_0^p + \sum_{s=1}^{N_\lambda} \alpha_s^p \lambda_s(j) \right).$$

Here M_{big} is a sufficiently large positive number to relax the constraints when the i -th waypoint is not reached within the prediction horizon. M is the number of sides of the polygon. The coefficients (α_0^p, α_s^p) are defined by $\forall s \in \{1, \dots, N_\lambda\}, \forall p \in \{1, \dots, N_p\}$:

$$\alpha_s^p = -(-1) \left\lceil \left(\frac{p+1}{2^{(s-1)}} \right) \right\rceil, \quad \alpha_0^p = 0.5 \sum_{s=1}^{N_\lambda} \left((\alpha_s^p)^2 - \alpha_s^i \right).$$

Remark *The waypoints constraint (5.30) is a circular region with radius δ_{th} . Setting $\delta_{\text{th}} = 0.5T_p V_{\text{max}}$ leads to generating a path passing through this region without decelerating to obtain a smoother path at lower cost. The waypoints are further apart from each other than $2\delta_{\text{th}}$.*

5.4.4 Control Setting and Objective Function

The area coverage planning can be formulated with respect to different objectives, e.g., time-to-completion, energy/fuel-saving, collision probability [58], the repetition rate [14], and to minimize the number of turns [175].

In this work, the main objective is to minimize the energy consumption and the uncovered area. This is achieved via the following optimization:

$$\min_{x,u,c^p,\lambda_s} J = W_u \sum_{j=1}^N (s_n(j) + s_e(j)) + W_\Phi \sum_{p=1}^{N_p} \Phi^p(N) + W_d D^{\text{Best}}(N). \quad (5.31)$$

Here, s_n, s_e are slack variables. The terminal cost $\Phi_p(N)$ penalizes all uncovered area. The third term $D^{\text{Best}}(N)$ represent the distance-to-go to the best waypoint, which is determined by solving the optimization problem:

$$D^{\text{Best}}(N) = \min_{p \in \mathcal{P}} \|r^p - r(N)\|,$$

where $\mathcal{P} := \{p | \Phi(p) = 1\}$ is the set of uncovered waypoints, see [58].

This section outlined the MPC/MILP approach for area coverage application. If a large-scale area needs a long time to be covered by a single vehicle, multi vehicles might provide a more efficient and robust solution. So the next section describes different MPC-based planning approaches for area coverage using multiple vehicles, e.g., centralized, decentralized, and distributed algorithms.

5.5 Simulation Results

This section presents numerical simulations and experimental studies to verify the feasibility and applicability of the proposed planning approach that yields good performance in a complex environment and also is computationally feasible. In the simulations, the quadcopter is assumed to have a sensor range of $R_s = 50$ m at an altitude of $h = 100$ m and an area to cover of 5×5 circular cells. First, a minimum number of waypoints that cover the area are found and passed to the MILP MPC considering the sensor range R_s and the dimensions of the search area, see [32, 34, 35]. The initial position of the quadcopter is at the map origin where $p_n = 0$ m and $p_e = 0$ m.

Fig. 5.8 shows the overall performance of the area coverage path planning for several scenarios. In each shown scenario, the complete area is covered by an energy minimizing path while avoiding static (Fig. 5.8b) and dynamically moving obstacles (Fig. 5.8c). The distance between the quadcopter and two dynamically moving obstacles while completing the full area coverage is always larger than the predefined safety range of 15 m, see Fig. 5.10. We furthermore analyzed the influence of preview information. To do so, we tested three different ranges of a “radar system”, which can

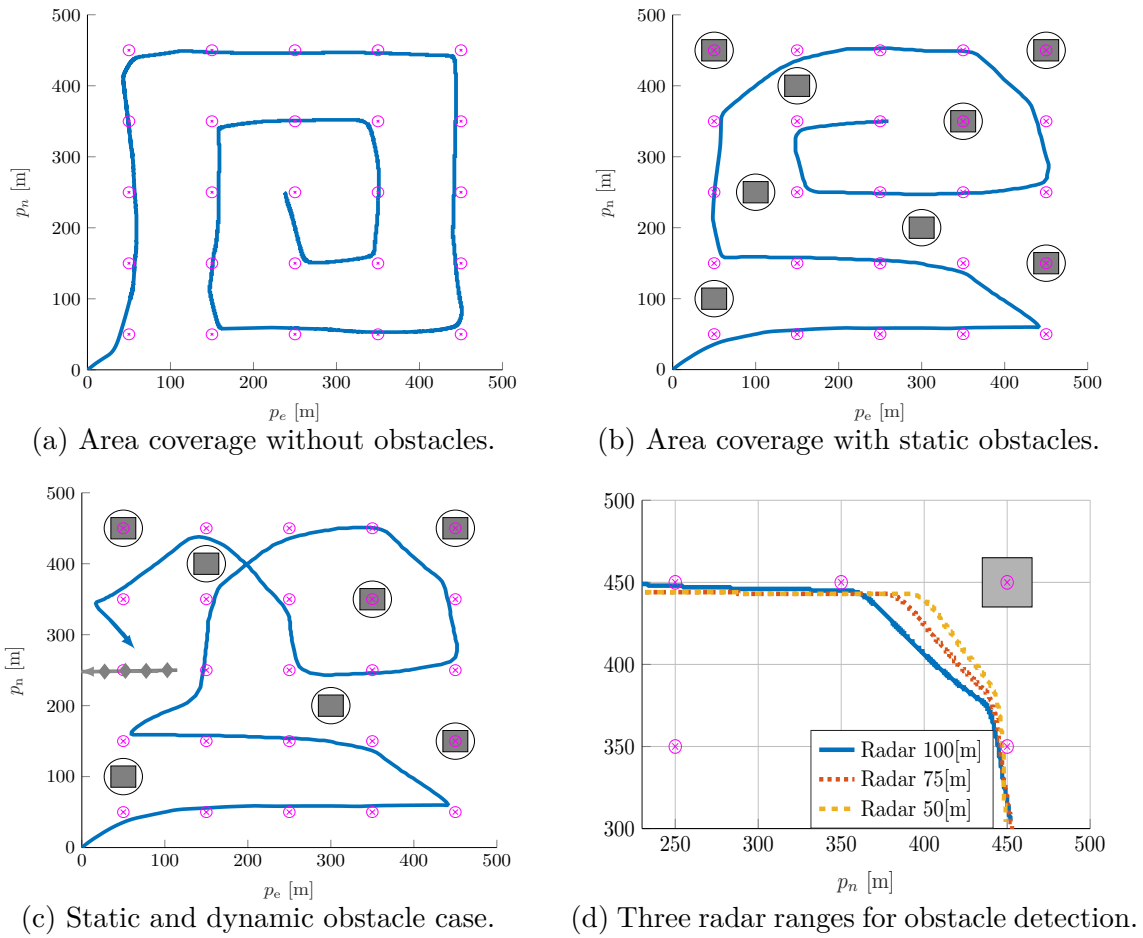


Figure 5.8: Area Coverage results for the quadcopter visiting all way points (magenta circles) while avoiding static obstacles (grey boxes) and dynamic obstacles (grey diamond). Supplementary video material can be found at <https://youtu.be/4GDmBLcjGZs>.

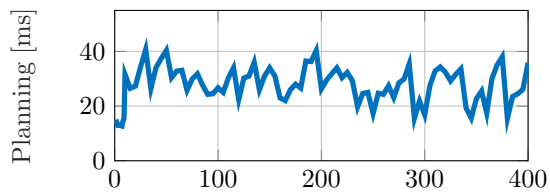


Figure 5.9: Computation time is smaller than planning sampling time $T_d = 1$ s.

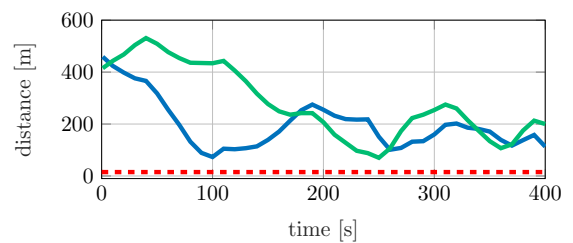


Figure 5.10: Euclidean distance between the quadcopter and two moving obstacles.

be used for obstacle detection, as shown in Fig. 5.8d. The MILP planner re-planned the path considering the detected obstacle earlier for wide range sensors (solid blue line), which leads to reducing the overall time for the complete coverage. Therefore, superior behavior compared to shorted range sensors (dashed lines). However, even shorter-range sensors show safe performance in terms of obstacle avoidance.

It is necessary to solve the planning problem (5.28) online. So the required time to solve both MPC optimization problems should be less than the sampling time (depicted in Fig. 5.9), which could be achieved. The computation times were achieved running both optimizations on an Intel[®] Core[™] i7-6700 CPU @ 3.40GHz desktop CPU. The above results of the proposed planning approach also depend on the underlying coding language and the solver. In this work, the MILP optimization problems have been numerically implemented in YALMIP [115] and solved via Gurobi [116].

These numerical simulations demonstrate that the planning approach can solve complex problems, e.g., area coverage. Note, the linear constraints provide good approximations to the vehicle dynamics and capabilities.

5.5.1 Experimental Validation

The numerical simulations demonstrate the real-time feasibility of MPC/MILP planning approach. We validate and investigate the real-time performance of the proposed planning scheme by experiments. Fig. 5.11 illustrates the implementation setup using Quanser's AVRS system [177]. This system includes a Qdrone, a ground control station, and a four-camera optitrack for motion capture [178]. Table B.1 outlines the quadcopter specifications.

The experimental results (Fig. 5.12-5.15) demonstrate the real-time feasibility of the proposed planning approach. The MPC/MILP planner overcomes the implementation challenges, e.g., limited onboard computation and real-world uncertainties arising from modeling error of the vehicle dynamics, tracking error of the lower-level controller, external disturbance, and sensing noise.

5.6 Discussion and Conclusion

This chapter presented a planning approach for the autonomous vehicle moving through clutter environments. Following a problem description and motivation, this chapter provided a brief survey of the common planning approaches to distinguish between these methods and the moving-horizon approach to clarify the motivation of the research topics presented in this chapter. Then we outlined the principal components of the proposed planning algorithm using MILP formulation. MILP can handle the non-convex constraints, e.g., obstacle avoidance. Contrary to most planning

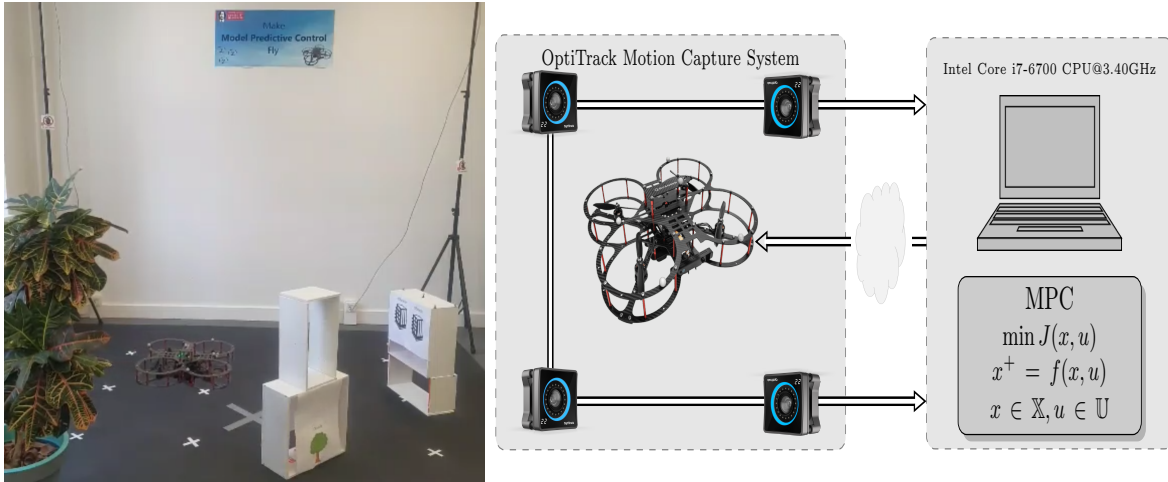


Figure 5.11: Experimental implementation setup using Quanser’s AVRS system, which includes a quadcopter, a ground control station, and a four-camera optitrack flex13 array used for motion capture. Supplementary video material can be found at <https://www.youtube.com/watch?v=1TKaP4NAewU>, <https://youtu.be/wZOhx-XGBc8>.

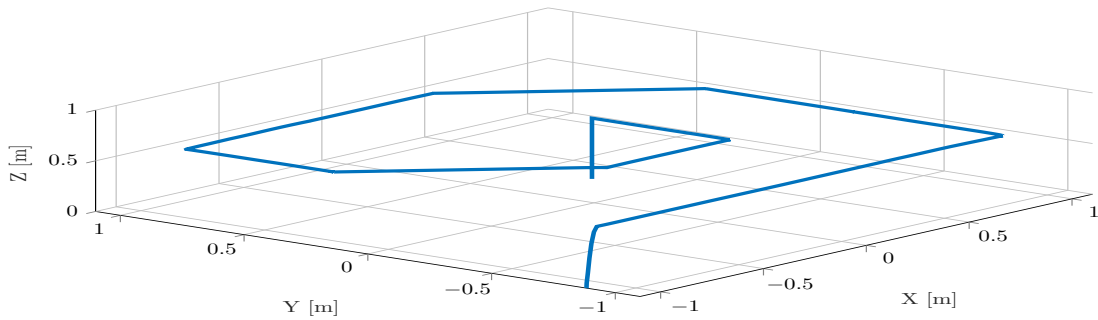


Figure 5.12: Experimental results: 3D quadcopter path covering an area.

approaches, the vehicle dynamics and constraints were taken into account in the optimization problem. Using the tuple encoding allow reducing the solution time by reducing the number of binary variables.

To handle bounded uncertainties, we proposed a robust planning approach using constraint tightening. The main drawback of this robust planning approach consists of conservativeness (i.e., less feasibility regions) introduced to account for the disturbances along the prediction horizon [17]. For this reason, we propose a state-dependent contract in Chapter 6. This contract specifies the capabilities of the lower-level controller, i.e., bounded error between the planned reference and the real movements.

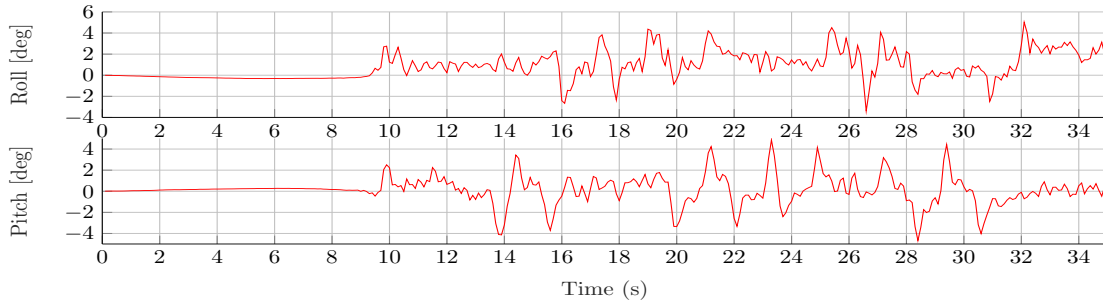


Figure 5.13: Experimental results: roll and pitch angles.

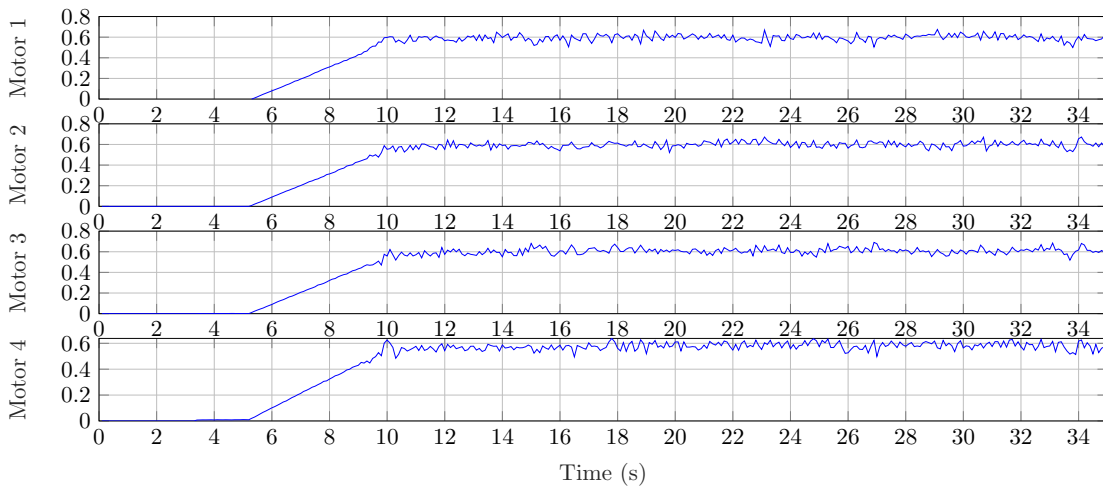


Figure 5.14: Experimental results: the control inputs of the motors.

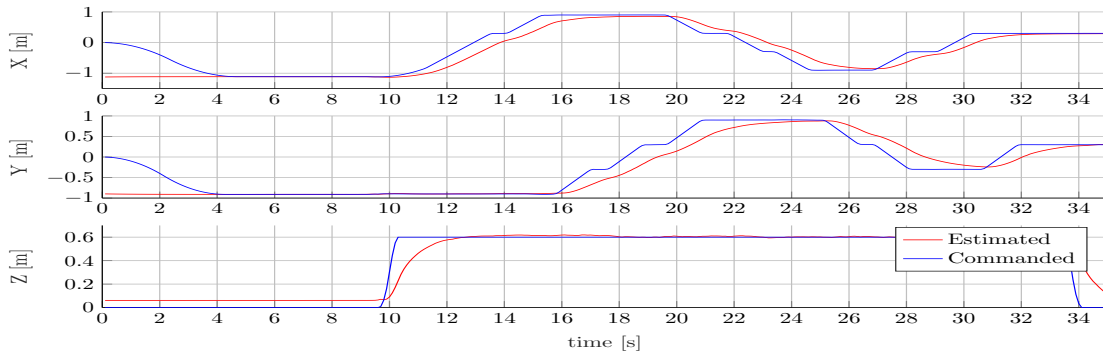


Figure 5.15: Experimental results: quadcopter position in (x,y,z) coordinates.

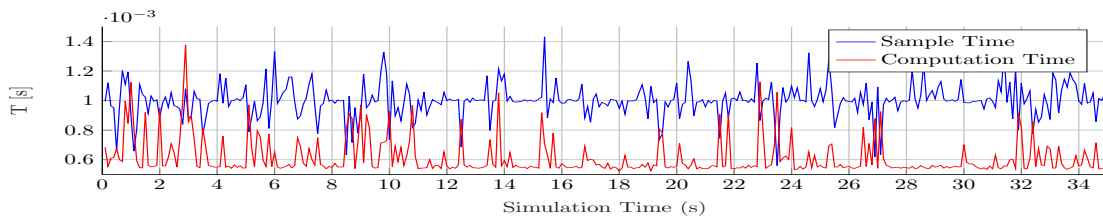


Figure 5.16: Experimental results: Sample time and computation time.

6 Fusing Planning and Path Following Control

This chapter proposes an interface between the planning and path following layers to provide guarantees. The presented moving-horizon planning approach takes the capability of a lower-level linear MPC into account in a hierarchical way.

6.1 Introduction and Motivation

Control and operation of autonomous systems are often decomposed into different decision and control layers in a hierarchy architecture, see Fig. 6.1. In this hierarchy, the planning and control systems are designed and use vehicle models of different complexities [32–35]. While these systems are highly interconnected, they operate on different time-scales, see Fig. 6.1. Due to the time-scale separation, the interaction between guidance and control is often neglected and not fully exploited.

As speed and performance demands increase, the separation between planning and control becomes challenging, leading to possibly unsafe behavior of the autonomous vehicle. The path needs to be more frequently recalculated, leading to loss of time-scale separation. Therefore, efficient and safe operations in highly dynamic environments require tight and coordinated interaction of these levels to achieve maximum performance and to guarantee collision avoidance.

We exploit MPC formulations on both the planning and control layers, which generate plans/controls robust to the vehicle uncertainties and environmental disturbance. The higher-level planner operates on a slow time-scale, taking simplified vehicle dynamics and constraints into account. The optimal planning solution, which avoids collisions, is provided to a lower-level controller. The autonomous vehicle itself is controlled by a lower-level controller with a faster time-scale, which obeys more detailed dynamic and kinematic constraints and follows the provided reference. The design of the controllers takes into account the interconnections in terms of dynamic constraints and reference definitions between the layers by exchanging information on the achievable precision of the lower-level. The planning layer can take this precision in a robust way into account, leading to a safe trajectory.

In reality, the vehicle might deviate from its expected trajectory, see Fig. 6.2, due to the uncertainties, which can cause performance degradation and even loss of vehicle stability [6, 8, 31, 37]. Therefore, robust planning and control approaches are needed to obtain safe and plausible references and control inputs to satisfy constraints and improve performance. Using a single combined planning and control approach can

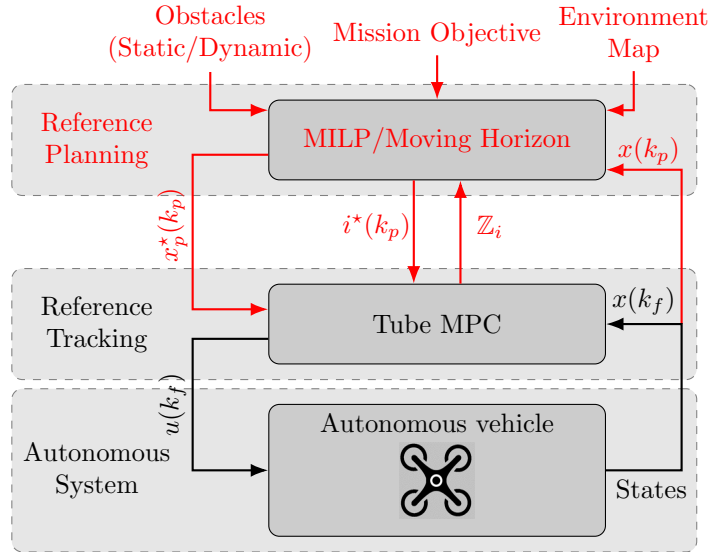


Figure 6.1: Illustration of the hierarchical moving-horizon control strategy and the information exchanged between reference planning and the tracking control layers. The lower-levels (black) utilizes a sampling time T_f (time index k), which is faster than higher-level planner (red) sampling time T_p (time index k_p).

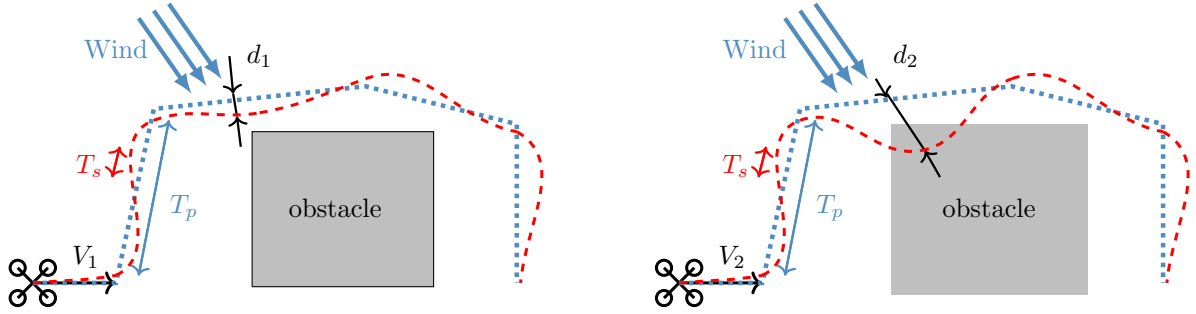
often not be implemented for complexity and computational feasibility reasons, as solving a robust planning control problem is challenging [62].

To address these challenges, we propose a hierarchical robust planning and control framework, see Fig 6.1. The proposed hierarchical structure effectively allows decomposing the complex task into tractable subproblems, allowing each to be pursued independently to reduce the computational burden. This paves the way for the real-time implementation of robust planning and control systems using the often limited onboard computational resources [32–34].

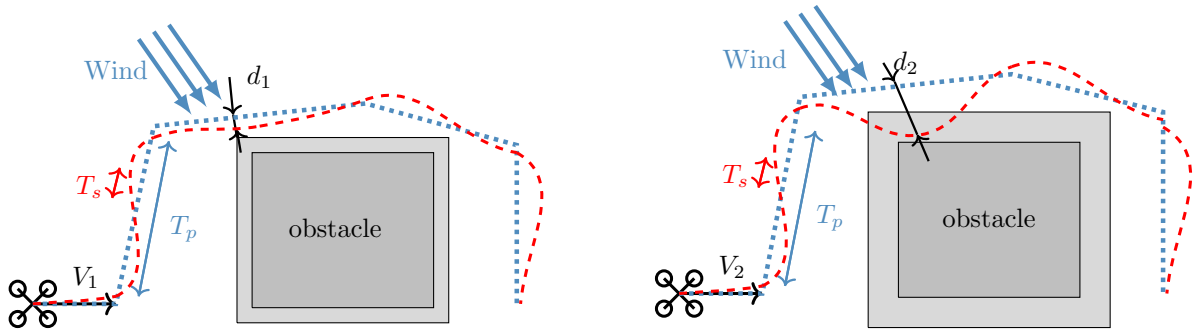
6.2 Robust Planning and Control with Guarantees

The autonomous vehicle might deviate from its expected trajectory, see Fig. 6.2, due to the uncertainties, e.g., model mismatch, tracking error, or measurement noise [17]. Additionally, the surrounding environments might be highly uncertain, e.g., due to wind disturbances [6, 8, 31, 37]. Neglecting the uncertainties can be hazardous, especially in populated/dynamic environments, and can result in performance degradation, complete failure, or obstacle collisions.

To overcome these challenges, robust planning approaches have been proposed to satisfy constraints under uncertainty, see, e.g., [6, 8, 31, 37, 174]. The common approach is to treat the uncertainty propagation as a deterministic worst-case by introducing



(a) Small disturbance influence for slow velocity. (b) Large disturbance influence for fast velocity.



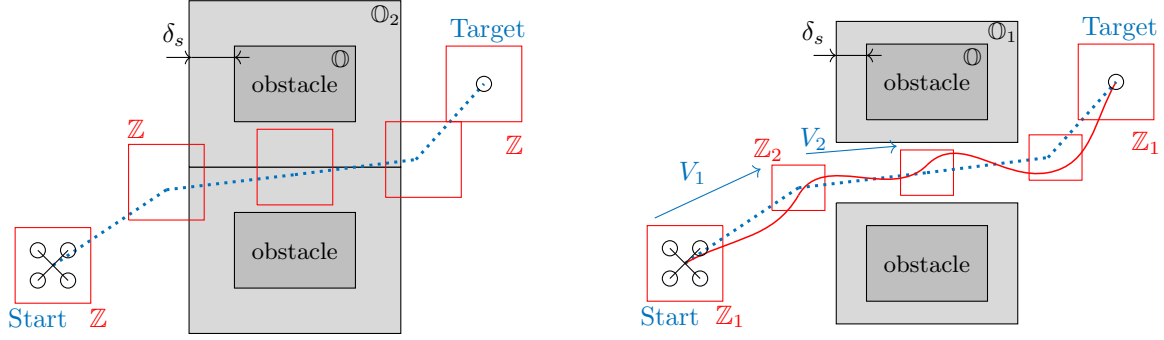
(c) Small obstacle enlargement.

(d) Large obstacle enlargement.

Figure 6.2: The impact of state-dependent uncertainties $d_1 < d_2$ can be smaller for a slow velocity $V_1 < V_2$ (Fig. 6.2a) compared to a fast velocity V_2 (Fig. 6.2b). The planner computes a feasible path (dotted blue line) at the slow time (T_p) but the lower-level controller (red dashed line) is unable to follow this path without violating the constraint (obstacle collision) for high velocity (Fig. 6.2b). To ensure constraint satisfaction, it is necessary to add a obstacle safety margins, which depend on the state, e.g., velocity, see Fig. 6.2c and Fig. 6.2d.

a conservative safety corridor [6, 62]. This approach might lead to a conservative solution for the planning problem if the same safety margin is always used.

We propose a robust hierarchical strategy, see Fig. 6.1. The planning layer and the lower-level controller agree on a “contract”, inspired by [179–182]. This contract maintains guaranteed “consistency” between the layers, thereby robustness and recursive feasibility, i.e., constraints satisfaction (collision avoidance). To do so, the high-level planner uses a constraint tightening approach. Encountered obstacles are enlarged by a safety bound to provide online a collision-free reference. The safety bounds are provided and ensured by the lower-level tube-based MPC [38], i.e., robust constraint satisfaction is guaranteed despite uncertainties by an adequate level of accuracy bounded in a tube, see Fig. 6.3.



- (a) No feasible solution can be found using constant constraint tightening (obstacle enlargement). (b) Feasible solution using an “adaptive” contract formulation, i.e., obstacle enlargement.

Figure 6.3: The planner cannot find a feasible solution in case of a fixed conservative constraint tightening/obstacle enlargement, e.g., Fig. 6.3a. Using an “adaptive” contract formulation, there exists a feasible solution by adjusting the vehicle velocity Fig. 6.3b.

6.2.1 Contract Formulation

For simplicity, we focus on a linear system dynamics subject to bounded disturbances:

$$x(k+1) = A(T_f)x(k) + B(T_f)u(k) + \omega(k), \quad (6.1a)$$

$$x(k) \in \mathbb{X} \subset \mathbb{R}^{n_x}, \quad u(k) \in \mathbb{U} \subset \mathbb{R}^{n_u}. \quad (6.1b)$$

Here $x(k), u(k)$ are the vehicle’s state and control input, while $\omega(k)$ is an unknown, but bounded disturbance. (6.1b) represent the state and control constraints, and the sets \mathbb{X} and \mathbb{U} are convex and compact. The system (6.1a) and lower-level controller operate on a fast time-scale with the sampling time T_f (time index k). The planner operates at a slower sampling time T_p with time index k_p , see Fig. 6.1.

We focus on state-dependent uncertainties $\omega(k)$, which might depend on the vehicle state $x(k)$, e.g., velocity, see Fig. 6.2. These state-dependent disturbances might be due to wind disturbance or due to the controller capability, see [33, 34, 183]. We approximate this dependency by N_ω different operating regions, defining disturbance-bounds.

Assumption 6.1 (*State-dependent disturbance bounds*) *If $x(k) \in \mathbb{X}_i \subseteq \mathbb{X}$, then $\omega(k) \in \mathbb{W}_i$, where \mathbb{X}_i and \mathbb{W}_i are convex, compact polytopes.*

The sets \mathbb{X}_i can overlap, i.e., one can have that $\mathbb{X}_i \cap \mathbb{X}_j \neq \emptyset$ for $i \neq j$.

Remark It is straightforward to generalize the results for both state constraints $x(k) \in \mathbb{X}_i \subseteq \mathbb{X}$ and input constraint $u(k) \in \mathbb{U}_i \subseteq \mathbb{U}$.

The autonomous vehicle is controlled by a lower-level controller, which might be based on a more detailed model (6.1) and constraints (6.1b) to track the reference with guaranteed accuracy. The non-convex constraints (obstacle avoidance) are handled by the planner to achieve an efficient and fast implementation of the lower-level controller. The planner generates a reference to navigate the vehicle, avoiding obstacles, i.e., satisfying (6.1b), for all possible disturbance realizations. To do so, the planner uses a simplified model of the form

$$x_p(k+1) = A_p(T_p)x_p(k) + B_p(T_p)u_p(k). \quad (6.2)$$

We assume that the planning model is obtained from the real model (6.1) using a larger sampling time $T_p = N_f \times T_f$ with time index k_p :

$$A_p(T_p) = A^{N_f}, \quad B_p(T_p) = \sum_{l=0}^{N_f-1} A^l B. \quad (6.3)$$

With respect to the real dynamics (6.1a) and the planning dynamics (6.2) we make the following assumption

Assumption 6.2 (*Controllability*) *The pairs (A, B) and (A_p, B_p) are controllable.*

Definition 6.1 *State-dependent contract: A contract specifies the controller capabilities (uncertainty bound), i.e., guaranteed accuracy for the corresponding operation region.*

According to the contract of Definition 6.1, the lower-level controller guarantees

$$x((k+1)N_f) - x_p(k+1) \in \mathbb{Z}_i, \quad \forall x(k) \in \mathbb{X}_i. \quad (6.4)$$

The contract defines the error between the planned reference and the real movements within this specific bounds. If $x \in \mathbb{X}_i$, i.e., the state is inside the corresponding operation region i , see Assumption 6.1. The contracts are known to both control levels, and they depend on the design of the lower-level controller and the (partly) selectable uncertainty bound. Thereby, the planner can improve the performances by switching between different operation regions, see e.g., Fig. 6.3, exploiting the controller capability in the planning optimization problem. The planner generates online the reference to guide an autonomous vehicle through cluttered and uncertain environments with less conservative constraint tightening (obstacle enlargement), see Fig. 6.3b, contrary to the constant constraint tightening, see Fig. 6.3a. Therefore, the planner, which operates on a slow time-scale T_p , see Fig. 6.1, computes and transmits to the lower-level controller the reference and selects the required maximum discrepancy \mathbb{Z}_i .

6.2.2 Robust Moving-Horizon Planning via MILP

Vehicle safety is typically defined in terms of robust constraint satisfaction, so neglecting the uncertainty in the path planning can result in obstacle collision, see Fig. 6.2.

We extend and improve the moving-horizon planning [32–34], represented in Chapter 5, to take the state-dependent contract (6.2.1) into account. The proposed algorithm modifies the constraints using a constraint tightening approach to guarantee the planning robustness despite uncertainty. This uncertainty is assumed to be unknown and lie in a bounded set \mathbb{Z}_i for each corresponding operating region i . The updated planning problem is represented as:

$$\min_{x_p, u_p, i_p} J_p(\{x_p\}, \{u_p\}) = \sum_{j=k_p}^{k_p+N_p-1} \|u_p(j)\| + \|x_{\text{Target}} - x_p(k_p + N_p)\|, \quad (6.5a)$$

$$\text{s.t. } \forall j \in \{0, \dots, N_p - 1\}$$

$$x_p(k_p + j + 1|k_p) = A_p x_p(k_p + j|k_p) + B_p u_p(k_p + j|k_p), \quad (6.5b)$$

$$x(k_p + N_p) - x_p(k_p|k_p) \in \mathbb{Z}_i, \quad (6.5c)$$

$$x_p(k_p + j|k_p) \in \mathbb{X}_i \ominus \mathbb{Z}_i, \quad (6.5d)$$

$$u_p(k_p + j|k_p) \in \mathbb{U} \ominus K\mathbb{Z}_i, \quad (6.5e)$$

$$C x_p(k_p + j|k_p) \notin \mathbb{O}_i \oplus C\mathbb{Z}_i, \quad (6.5f)$$

$$(x_p(k_p + j|k_p), u_p(k_p + j|k_p)) \in \mathbb{I}_i, \quad (6.5g)$$

$$x_p(k_p + N_p|k_p) \in \mathbb{X}_{T_i}. \quad (6.5h)$$

Here $(k_p + j|k_p)$ denotes the prediction of a planning variable at time $k_p + j$ made at a slower time-scale k_p . The stage cost (6.5a) minimizes the control input $\|u_p(j)\|$, while the terminal cost penalizes the distance to the target point x_{Target} at the end of planning horizon N_p . The constraint (6.5c) defines that the initial state of the autonomous vehicle lies in a region \mathbb{Z}_i guaranteed by the lower-level controller. The sets \mathbb{Z}_i are convex compact polytopes and depend on the lower-level closed-loop tracking accuracy achieved for the selected operation region i . This set is also used to modify the vehicle constraints (6.5d, 6.5e, 6.5f). The inter-sample constraints (6.5g) and the terminal constraint (6.5h) depend on the operation region i . Concerning the inter-sample constraints (6.5g) we make the following assumption to guarantee that the lower-level controller can satisfy the constraints at all times.

Assumption 6.3 (*Inter-sample constraints*) *The set \mathbb{I}_i defines the inter-sample con-*

straints, determined such that $(x_p, u_p) \in \mathbb{I}_i$ implies for $l = 1, \dots, N_f - 1$:

$$A_f^l x_p + \sum_{m=0}^{l-1} A_f^m B_f u_p \in \mathbb{X}_i \ominus \mathbb{Z}_i, \quad (6.6a)$$

$$C(A_f^l x_p + \sum_{m=0}^{l-1} A_f^m B_f u_p) \notin \mathbb{O}_i \oplus C\mathbb{Z}_i. \quad (6.6b)$$

A trivial choice is to choose \mathbb{I}_i directly as (6.6), which might slightly increase the computational effort. However, this is not always necessary, e.g., one can use alternative approaches to enlarge the obstacles, see Section 6.2.3.

For the terminal set \mathbb{X}_{T_i} , we assumed that

Assumption 6.4 (*Terminal sets*) *There exists a terminal control law $\kappa_T(x)$ and terminal sets \mathbb{X}_{T_i} such that if $x \in \mathbb{X}_{T_i}$, then $\forall l = 1, \dots, N_f - 1$*

$$A_p x_p + B_p \kappa_T(x) \in \mathbb{X}_{T_i}, \quad (6.7a)$$

$$x_p \in \mathbb{X} \ominus \mathbb{Z}_i, \quad (6.7b)$$

$$\kappa_T(x_p) \in \mathbb{U} \ominus K\mathbb{Z}_i, \quad (6.7c)$$

$$(x_p, \kappa_T(x_p)) \in \mathbb{I}_i, \quad (6.7d)$$

$$C x_p \notin \mathbb{O}_i \oplus C\mathbb{Z}_i. \quad (6.7e)$$

Note that the terminal set is non-convex due to the obstacle avoidance and (possibly) the inter-sample constraints. Nonetheless, this assumption can be satisfied in many applications. A straight forward choice is to focus on admissible, nominal steady state $x_p = A_p x_p + B_p \kappa_T(x_p)$ for the terminal sets, i.e., points where the autonomous vehicle can stop. The higher-level planner sends the selected operation region i^* and the following inter-sampled reference to the lower-level controller

$$x_{ref}(kN_f + j) = A_f^j x_p^*(k|k) + \sum_{m=0}^{j-1} A_f^{m-1} B_f u_p^*(k|k). \quad (6.8)$$

Proposition 6.1 (*Planning Recursive Feasibility*) *Let Assumptions 6.1- 6.4 hold. If the moving-horizon planning problem (6.5) is feasible at time k , and the lower-level controller guarantees bounded error, i.e., $x(k) - x_{ref}(k) \in \mathbb{Z}_{i^*}$ for the reference (6.8), then planning problem (6.5) is feasible at $k + 1$.*

Proof Let us denote the optimal solution of the planning problem (6.5) as:

$$\{x_p^*(k_p|k_p), \dots, u_p^*(k_p|k_p), \dots, i_p^*\}.$$

To verify the above result consider for the optimization problem at $k_p + 1$ the fol-

lowing initial guess based on the previous solution and the terminal control law κ_T :

$$\begin{aligned} \dot{i}_p &= \dot{i}_p^*, \\ x_p(k_p + j|k_p + 1) &= x_p^*(k_p + j|k_p), \quad j = 1, \dots, N_p, \\ x_p(k_p + N_p + 1|k_p + 1) &= A_p x_p^*(k_p + N_p|k_p) + B_p \kappa_T(x_p^*(k_p + N_p|k_p)), \\ u_p(k_p + j|k_p + 1) &= u_p^*(k_p + j|k_p), \quad j = 1, \dots, N_p - 1, \\ u_p(k_p + N_p|k_p + 1) &= \kappa_T(x_p^*(k_p + N_p|k_p)). \end{aligned}$$

One can verify straightforwardly that this initial guess is feasible (but suboptimal), i.e. that all constraints of (6.5) are feasible at $k_p + 1$ using the properties of the terminal set \mathbb{X}_T and the guarantee on the lower-level control accuracy. ■

Remark (Planning without feedback) *In principle, one can modify the approach such that the initial constraint (6.5c) is only enforced at the begin ($k_p = 0$) and use the equality constraint $x_p^*(k_p + 1|k_p) = x_p(k_p + 1|k_p + 1)$ instead of (6.5c) for $k_p > 0$. This removes the feedback from the plant to the planning, which enables a computationally more efficient planning, but leads to a decreased control performance.*

Remark (Offline computation) *The disturbance feedback policy and tightened constraints are computed offline, this prevents complexity increase in the online optimization. Therefore, the decision space of the online optimization remains similar to nominal MPC, allowing for efficient real-time computation.*

In this setup, the planner computes and transmits to the lower-level controller not only the reference but also selects the required maximum discrepancy due to the choice of \mathbb{Z}_i . The following section describes how the reference planner can improve the performances by switching between different operation regions, see e.g., Fig. 6.3.

6.2.3 Switching between Operating Regions

According to Assumption 6.1, the uncertainty set $\omega \in \mathbb{W}_i$ defines the state-dependent disturbance bounds for every operating region \mathbb{X}_i . For each set, the lower-level controller guarantees the bounds \mathbb{Z}_i on the tracking error according to the contract definition 6.1. Consequently, the planner can exploit, as an additional degree of freedom, scheduling between the operating regions, defined as state constraints sets:

$$\mathbb{X}_i \ominus \mathbb{Z}_i \equiv \{x_p | F_i x_p \leq G_i\}, \quad \forall i \in \{1, \dots, N_\omega\}.$$

Here, N_ω is the number of the operating regions, i.e., velocity ranges. The scheduling between the operating regions is reformulated by exploiting the big M method, see [33]:

$$F_i x_p \leq G_i + M_{\text{big}}(1 - d_i(k_p)), \quad \forall i \in \{1, \dots, N_\omega\}.$$

Here M_{big} is a sufficiently large positive number to relax the constraints when the i -th region is not activated. $d_i(k_p)$ is a binary decision variable used to decide which region is active at time instance k_p . When this happens, the binary decision variable is set to $d_i(k_p) = 1$, which labels the constraint i -th being activated at time k_p . An extra constraint $\sum_{i=1}^{N_\omega} d_i(k_p) = 1$ is imposed to ensure that at least one region is activated at ever time. As a result, the constraint \mathbb{X}_i is tightened, i.e., the obstacle boundaries \mathbb{O}_i are enlarged via the tracking error set \mathbb{Z}_i for each region i .

Non-convex avoidance constraints can be approximated by convex polygons introducing extra binary variables $b_m^i(k_p)$ as in Chapter 5:

$$\forall m \in \{1, \dots, M\}, \forall k \in \{1, \dots, N_p\}, \forall i \in \{1, \dots, N_\omega\}$$

$$\left(p_n(k_p) - p_n^o(k_p) \right) \cos \frac{2\pi m}{M} + \left(p_e(k_p) - p_e^o(k_p) \right) \sin \frac{2\pi m}{M} \geq \delta_{\text{safe}}^i - M_{\text{big}} b_m^i(k_p),$$

An extra constraint $\sum_{m=1}^M b_m^i(k_p) \leq M - d_i(k_p)$ is imposed to ensure that at least one constraint is active for region i -th, i.e., $d_i(k_p) = 1$. δ_{safe}^i is a minimum separation distance between the autonomous vehicle $(p_n(k_p), p_e(k_p))$ and the obstacle position $(p_n^o(k_p), p_e^o(k_p))$ at time step k_p for the region i -th.

Remark *The operating regions are computed offline, while the scheduling is performed online as additional degree of freedom.*

As a result, the planner has an additional degree of freedom to adjust the obstacle enlargement δ_{safe}^i by controlling the vehicle velocity, see Fig. 6.3. The planning approach generates online reference commands on a slow time-scale T_p to guide the autonomous vehicle through cluttered and uncertain environments. To track the generated reference, the following section uses a robust tube-based MPC [38], which operates on a fast time-scale T_f using the real system dynamics (6.1), see Fig. 6.1.

6.2.4 Robust Model Predictive Tracking Control

The proposed tube-based MPC utilizes a nominal prediction dynamics (state z , input v) starting from the current real state

$$z(k+j+1|k) = Az(k+j|k) + Bv(k+j|k), \quad (6.9a)$$

$$z(k|k) = x(k), \quad (6.9b)$$

for predicting the effect of future disturbances $w(k+j)$, which is taken into account using the fictitious, auxiliary control law

$$u(k+j|k) = v(k+j|k) + K(x(k+j) - z(k+j|k)), \quad (6.10)$$

where K is a control gain, such that $A + BK$ is Schur stable. Note that one does not need to use only one control gain, see [184]. The difference

$$e(k + j|k) = x(k + j) - z(k + j|k)$$

between the predictions made using the prediction dynamics (6.9) and the real system dynamics (6.1) can be bounded in form of sets with the above auxiliary control law.

In detail, if the system is in the i -th operation region, then $e(k + j|k) \in \mathbb{E}_i(j)$ where

$$\mathbb{E}_i(j + 1) = (A + BK)\mathbb{E}_i(j) \oplus \mathbb{W}_i, \quad \mathbb{E}_i(0) = \{0\}. \quad (6.11)$$

As $j \rightarrow \infty$ the sets size \mathbb{E}_i increases, but is bounded by $\mathbb{E}_i(j) \subseteq \mathbb{Z}_i$, where \mathbb{Z}_i is the (minimum) robust positive invariant set:

$$\mathbb{Z}_i \supseteq (A + BK)\mathbb{Z}_i \oplus \mathbb{W}_i. \quad (6.12)$$

The lower-level MPC predicts until the next planning instant, so the horizon shrinks between planing instants and is increased at the next planing instant again to length N_f . In detail, if k is a multiple of N_f , then $L_k = N_f$. Otherwise, L_k is chosen such that $L_k < N_f$ and $k + L_k$ is a multiple of N_f .

The lower-level MPC predicts a (nominal) state and input trajectory

$$\begin{aligned} \mathbf{z}(k) &= \{z(k|k), \dots, z(k + L_k|k)\}, \\ \mathbf{v}(k) &= \{v(k|k), \dots, v(k + L_k - 1|k)\}, \end{aligned}$$

which are consistent with the nominal dynamics (6.9) and the following constraints

$$z(k + j|k) \in \mathbb{X}_i \ominus \mathbb{E}_i(j), \quad (6.13a)$$

$$v(k + j|k) \in \mathbb{U} \ominus K\mathbb{E}_i(j), \quad (6.13b)$$

$$Cz(k + j|k) \in \{Cx_{ref}(k + j) \oplus C(\mathbb{Z}_i \ominus \mathbb{E}_i(j))\}, \quad (6.13c)$$

$$z(k + L_k|k) \in \{x_{ref}(k + L_k) \oplus (\mathbb{Z}_i \ominus \mathbb{E}_i(L_k))\}. \quad (6.13d)$$

Note that the (convex) state and input constraints (6.1b) are directly included in these constraints. In contrast the non-convex obstacle avoidance constraints are considered by requiring that the lower-level controller enforces the guaranteed accuracy on the vehicle position and the terminal state constraint, which results in convex constraints.

In summary, the lower-level MPC solves the following optimization problem:

$$\begin{aligned} \min_{\mathbf{z}, \mathbf{v}} J_f(\mathbf{z}(k), \mathbf{v}(k)) = & \sum_{j=k}^{k+L_k-1} \|x_{ref}(j) - z(j|k)\|_Q^2 + \|v(j|k)\|_R^2 \\ & + \|x_{ref}(k+L_k) - z(k+L_k|k)\|_P^2, \end{aligned} \quad (6.14a)$$

$$\begin{aligned} \text{s.t. } \forall j \in \{0, \dots, L_k - 1\} \\ z(k|k) = x(k), \end{aligned} \quad (6.14b)$$

$$z(k+j+1|k) = Az(k+j|k) + Bv(k+j|k), \quad (6.14c)$$

$$z(k+j|k) \in \mathbb{X}_i \ominus \mathbb{E}_i(j), \quad (6.14d)$$

$$v(k+j|k) \in \mathbb{U} \ominus K\mathbb{E}_i(j), \quad (6.14e)$$

$$Cz(k+j|k) \in \{Cx_{ref}(k+j) \oplus C(\mathbb{Z}_i \ominus \mathbb{E}_i(j))\}, \quad (6.14f)$$

$$z(k+L_k|k) \in \{x_{ref}(k+L_k) \oplus (\mathbb{Z}_i \ominus \mathbb{E}_i(L_k))\}. \quad (6.14g)$$

Here, $Q \in \mathbb{R}^{n_x \times n_x}$, $Q_N \in \mathbb{R}^{n_x \times n_x}$, and $R \in \mathbb{R}^{n_u \times n_u}$ represent the positive definite weighting matrix for the state and the inputs respectively, used to penalizes the deviation error from the reference x_{ref} and the control input. The optimization problem is a convex quadratic program and can be solved efficiently, even on computationally limited hardware. Note that the optimization problem (6.14) is based on the current state and the reference and the operation region determined by the higher-level planner. For the closed-loop, we have the following properties.

Proposition 6.2 (*Constraint satisfaction*) *Let Assumptions 6.1, 6.3 hold. If the lower-level MPC problem (6.14) is feasible, then the constraints $x_p(k) \in \mathbb{X}_i \subseteq \mathbb{X}$ and $u_p(k) \in \mathbb{U}$ are satisfied and the obstacles are avoided, i.e., (6.5f) holds.*

Proof From the constraints (6.9), (6.10) and (6.13) we have that $z^*(k|k) = x_p(k)$, as $v^*(k|k) = u_p(k)$, $Cz^*(k|k) = Cx_p(k)$. Together with $\mathbb{E}_i(0) = \{0\}$ and Assumption 6.1 this implies that the vehicle constraint (6.1b) is satisfied. Moreover, $x_p(k) \in x_{ref}(k) \oplus \mathbb{Z}_i$, which implies together with the avoidance constraint (6.5f) and the inter-sample constraints (6.6) and that $Cx_p(k) \notin \text{int}(\mathbb{O}_i)$, i.e. (6.5f) holds. \blacksquare

Proposition 6.3 (*Recursive feasibility of the hierarchical control scheme*) *Let Assumptions 6.1- 6.4 hold. If the planning problem (6.5) is feasible at $k = 0$, then for the closed-loop system consisting of the higher-level moving-horizon planner (6.5), the lower-level controller (6.14), and the plant dynamics (6.1), the optimization problems (6.5), (6.14) are feasible for any $k > 0$.*

Proof The proof has three parts: first we show that feasibility of the planning problem (6.5) implies feasibility of the lower-level controller (6.14), second that feasibility of the lower-level controller (6.14) implies feasibility of the lower-level controller (6.14) (if $k+1$ is not a multiple of N_f) or the planning problem (6.5) (otherwise).

If the higher-level planning problem (6.5) is feasible at k , then using the lower-level input trajectory

$$v(k+j|k) = u_p^*(k|k) + K((A+BK)^j x_p(k) - x_{ref}(j))$$

where $j = 0, \dots, N_f - 1$ results in a state trajectory satisfying all constraints due to the constraint tightening utilized in the upper and lower-level optimization problems.

If $k+1$ is not a multiple of N_f , i.e. no planing takes place and the horizon $L(k)$ shrinks, then due to the design of the set $\mathbb{E}_i(j)$ at $k+1$ a feasible nominal state trajectory $\mathbf{z}(k+1)$ and a nominal input trajectory $\mathbf{v}(k+1)$, satisfying the constraints (6.9) and (6.13), exists:

$$\begin{aligned} z(k+j|k+1) &= z^*(k+j|k) + (A+BK)^{j-1}w(k), \\ v(k+j|k+1) &= v^*(k+j|k) + K(A+BK)^{j-1}w(k). \end{aligned}$$

If $k+1$ is a multiple of N_f , then the feasibility of the lower-level controller (6.14) at k implies that $x_p(k) - x_{ref}(k) \in \mathbb{Z}_{i^*}$, which together with Proposition 6.1 implies that the planning problem (6.5) is feasible. ■

Remark We assumed that the available constraints $\{\mathbb{X}_i, \mathbb{Z}_i\}$ are fixed over a certain time at the initial time. In principle, they can also be changed/adapted, e.g., due to changing weather conditions for UAVs. This will be addressed in the future research.

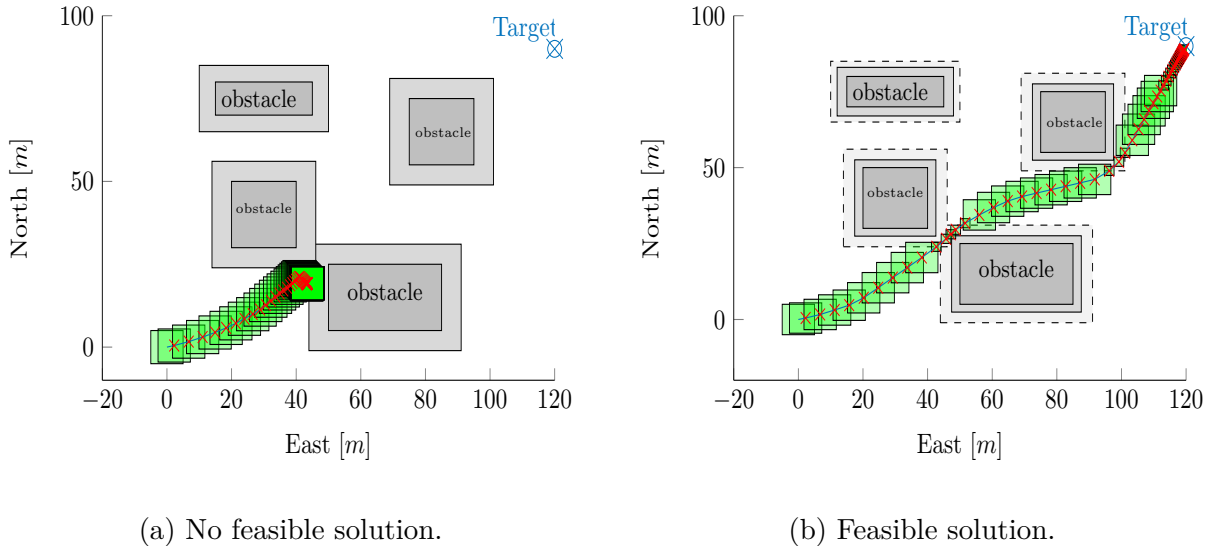


Figure 6.4: Adaptive contracts achieves less conservative results with enlarged feasible region (Fig. 6.4b) compared to the case of constant worst-case uncertainty (Fig. 6.4a).

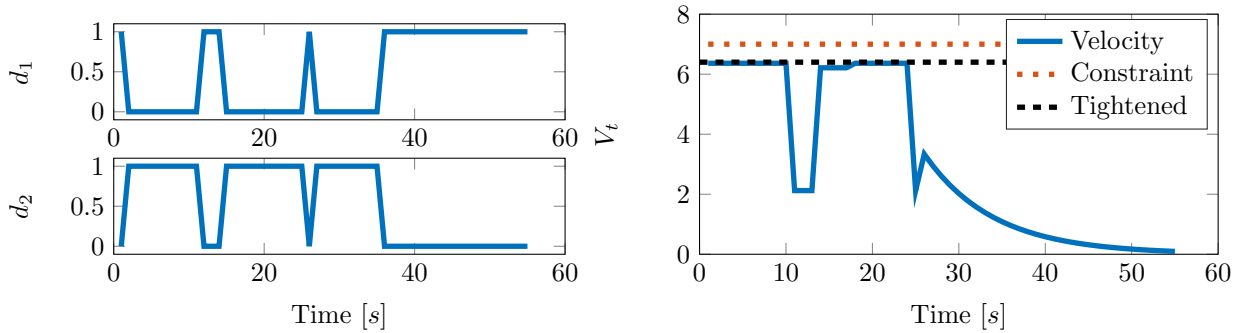


Figure 6.5: The reference planner selects different contracts decides which state region is activated (i.e., adjusts the vehicle velocity V_t) via the decision variables d_1 and d_2 .

6.2.5 Simulation Results

To illustrate the efficiency of the proposed robust hierarchical MPC, we consider a linear vehicle model. For the sampling time of the planner, we use $T_p = 1s$ and for the tracking control $T_f = 0.1s$, i.e., $N_f = 10$, while the planning horizon is set to $N_p = 15$.

As depicted in Fig. 6.4a, there is no feasible solution using conservative constraint tightening (Obstacle enlargement), even if there might exist a feasible one, due to the short planning horizon. Nevertheless, a long horizon might require a considerable computational time, which it is often unfeasible for onboard implementation. In the “adaptive” contract case, the reference planner can find a feasible solution with less conservative constraint tightening by switching between the contract, see Fig. 6.4b. The contract choice enables the planner to find a feasible solution by adjusting the vehicle velocity V_t to operate at the state region with smaller uncertainty bound.

The reference planner decides which state region is activated via the decision variables d_1 and d_2 , see Fig. 6.5. As we can see, the vehicle is moving at fast speed up to the time 11s when it is close to the obstacle. At this time, the planner activates the state region with less uncertainty bound. Thereby, the planner finds a feasible solution and then accelerates again after passing through the obstacles.

Both robust MPC formulations were formulated with YALMIP [115] and solved via Gurobi [116], while the tube MPC is implemented using the MPT toolbox [119].

In this section, the proposed hierarchical framework guarantees robustness and recursive feasibility, i.e., it maintains a guarantee of “consistency” between the different layers. Even though the path planner takes a simplified dynamic model of the system into account, this is often too simple for good performance.

6.3 Discussion and Conclusion

We have proposed in this chapter a combined moving-horizon planner and robust model predictive controller and shown the applicability of these planning and control approaches in simulation.

In the hierarchical strategy, the upper and lower-levels exploit “contract” (guaranteed precision levels). To ensure compatibility between these levels and to guarantee safety, we do facilitate the recursive feasibility of the hierarchical controller by suitable constraints. The reference planning is formulated as an MILP considering constraints tightening. The constraints tightening –achievable performance– is calculated at the lower-level controller, based on the capabilities of the autonomous vehicle. Utilizing for the planning a different controller with different precision provides significant advantages, e.g., allowing for less conservative results with the enlarged feasible region compared to assuming a constant worst-case uncertainty. This decomposition of the control problem reduces the computational cost, enables real-time implementation for robust control in an autonomous vehicle. Simulation results demonstrate the efficiency of the proposed hierarchical approach.

7 Conclusion and Future Perspectives

In this work, we proposed to use MPC to solve different problems, e.g., load alleviation, planning, and control. We investigate both theoretical aspects and real-time implementation, such as controller structure, feasibility, and stability.

First, we considered MPC for load alleviation for passenger aircraft. The MPC scheme's advantages are the ability to consider the aircraft dynamics and constraints to minimize a cost function, e.g., loads reduction. Furthermore, MPC allows using the preview information, e.g., from LIDAR. As shown, the anticipation of the coming gust improves the load alleviation performance. Two different aircraft were used to demonstrate the proposed approach's effectiveness through numerical simulations subject to different gust conditions. Furthermore, this work provided one of the first wind-tunnel investigations to validate the real-time capability and effectiveness of MPC-based GLA. As shown, MPC has the potential to alleviate the gust loads actively, which allows designing lighter aircraft, thereby reducing the fuel costs.

The second part considered the planning and control problem of autonomous vehicles moving through uncertain/dynamic environments. The main developments of this part are the inclusion of robustness in a hierarchical control framework.

Contrary to most planning approaches, the vehicle dynamics and constraints are taken into account in the optimization problem. We proposed to use MILP to solve non-convex optimization problems, e.g., obstacle avoidance. We also proposed a new approach to reduce the number of binary variables used to represent the obstacle and waypoint constraints. This reduced the solution time and enabled online planning on embedded platforms.

Based on this approach, the constraint tightening was used to guarantee constraint satisfaction despite uncertainties. Obstacles were enlarged by a safety margin to ensure safe collision avoidance. The planning approach was extended to area coverage. The efficiency and real-time feasibility of the proposed approaches are emphasized through numerical simulations, followed by hardware experiments using a quadcopter flying in a cluttered environment.

In Chapter 6, we considered the interaction between the planning and control layers. To ensure compatibility and constraint satisfaction, we proposed that the moving-horizon planner considered the low-level controller capabilities to switch between different controllers in a predictive and optimized way, exploiting the contract idea. Simulations demonstrated that contracts allow improving the planning applicability while maintaining recursive feasibility.

The effectiveness and robustness of the proposed planning and control strategies were validated via simulation results. As shown, the decomposition of the control problem reduced computational cost, enabled real-time implementation for robust planning and control. While the planning and control methodologies developed in this thesis can be applied to many vehicle types, we focused on quadcopter scenarios operating in cluttered/uncertain environments. Experimental results and flight-test demonstrated that the proposed algorithms successfully overcome the implementation challenges, e.g., limited onboard computation and different uncertainties, e.g., external disturbance and sensing noise. The experimental work was helpful in identifying real-world problems not captured in numerical simulations.

In summary, this thesis explored mainly designing and real-time implementation of MPC for load alleviation, planning, and control that deal with different types of environmental uncertainties and taking the different control layers into account.

7.1 Directions for Future Research

Stimulated by the findings, many future research directions can be derived. The following challenges stand out as further research areas.

1- Reducing the computational burden: Despite the achieved results, real-time MPC implementations are still challenging in aerospace applications, especially for nonlinear (high-dimension) systems, e.g., flexible aircraft. They require solving nonlinear programming problems, which are in general non-convex, with a large number of variables. Approximation on MIP formulation allows using LMPC to reduce the computational time. However, this has some drawbacks, e.g., linearization, are only valid on a limited range and difficult to obtain, e.g., in the case of flexible aircraft. One future direction might be to integrate MPC with dynamic feedback linearization, i.e., an algebraic transformation of nonlinear dynamics to a linear model with new control inputs. Another direction is to apply multi-parametric quadratic programming or approximation functions, e.g., hinging hyperplanes or artificial neural networks.

2- Extending the idea of contracts to nonlinear systems using, e.g., ellipsoidal tube NMPC [185] would enable reducing conservativeness in the case of nonlinear systems. In this case, the lower controller online sends the tube parameterization upwards. Therefore, the planner can predict a possible uncertainty evaluation over the planning horizon.

3- Integrate MPC with a machine learning scheme to improve the prediction by including, for example, the obstacle velocity.

4- Extend the hierarchical MPC structure to include more layers with faster sampling to achieve better performance.

5- Considering communications issues, e.g., variable time-delay or data-losses,

to maintain the connectivity between the multi-vehicles.

6- Substantiate the usefulness of the planning and control approaches in a real outdoor environment with different tasks, e.g., searching and tracking a moving object. This requires further studies to reduce the computational effort without impacting the planning and control performance. For instance, the main hardware issue is to process the sensor information about the environment, which are currently often too computationally intensive to be implemented on embedded systems.

7- Real flight test experiments; Although the WT experiment provides a cheap method to investigate the real-time performance of the proposed control scheme, the full aircraft dynamics, i.e., six DoF, are not considered. During the Inflight project activities, we are preparing to validate the proposed MPC/GLA system on the Remos GX aircraft via real flight tests. The interaction between the MPC/GLA and the aircraft systems, e.g., the navigation system and the primary flight control system, is depicted in Fig. 7.1.

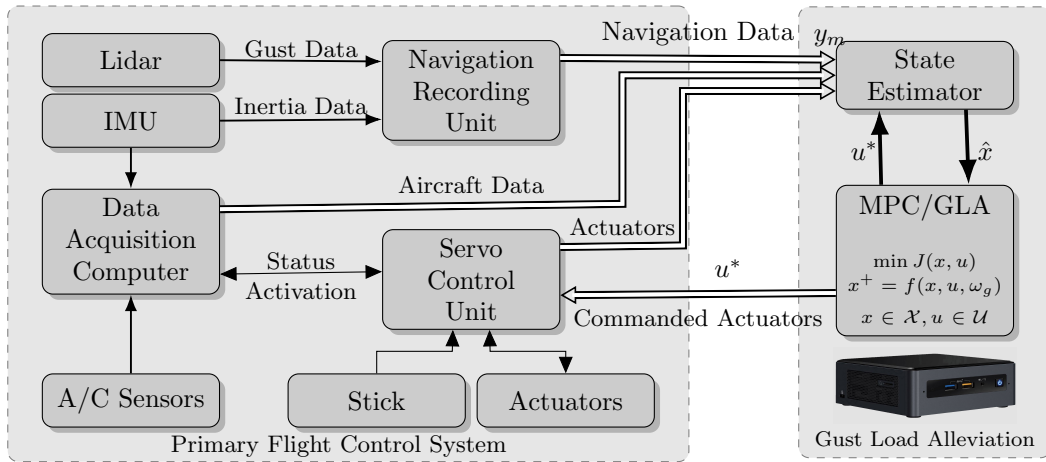


Figure 7.1: Schematic diagram of flight control system for the Inflight Project. μ AO-MPC code determines the control input u^* using the state information \hat{x} from the state estimator, which receives the measured output y_m from the main data acquisition computer, navigation recording, and servo control units, adapted from [186].

A Gust Load Alleviation

A.1 Green Regional Aircraft States

The Green Regional Aircraft model includes 153 states, which are expressed in three different reference systems, see Fig. 4.4 [141, 143, 157]:

- 20 aerodynamic states and 38 structural states.
- actuator dynamics (6 states) and low pass filter for control surface deflections (3 states).
- discrete gust shaping filter (4 states).
- IMU (4 states), 9 accelerometers (63 states), and 3 delay filters (15 states).

A.2 Aircraft Parameters

Property	Green Regional Aircraft	Remos GX Aircraft	Unit
Mass	52,266	600	kg
Length	40.69	6.48	m
Height	7.88	2.28	m
Wingspan	34.96	9.32	m
Wing root chord	5.45	1.23	m

Table A.1: Aircraft geometric properties [143, 187].

Variable	Symbol	Min	Max	Unit
True Airspeed	V_t	22.6	69.45	m/s
Angle of Attack	α	-10	30	$^\circ$
Side Slip Angle	β	-20	20	$^\circ$
Vertical load factor	n_z	0	4	g
Bank angle	ϕ	-180	180	$^\circ$
Pitch angle	θ	-90	90	$^\circ$
Yaw angle	ψ	0	360	$^\circ$
Bank & pitch & yaw rates	p, q, r	-100	100	$^\circ/\text{s}$
Elevator	δ_{elev}	-30	30	$^\circ$
Rudder	δ_{rud}	-28	28	$^\circ$
Ailerons	δ_{ar}, δ_{al}	-21	12	$^\circ$

Table A.2: Remos GX aircraft limitations [186].

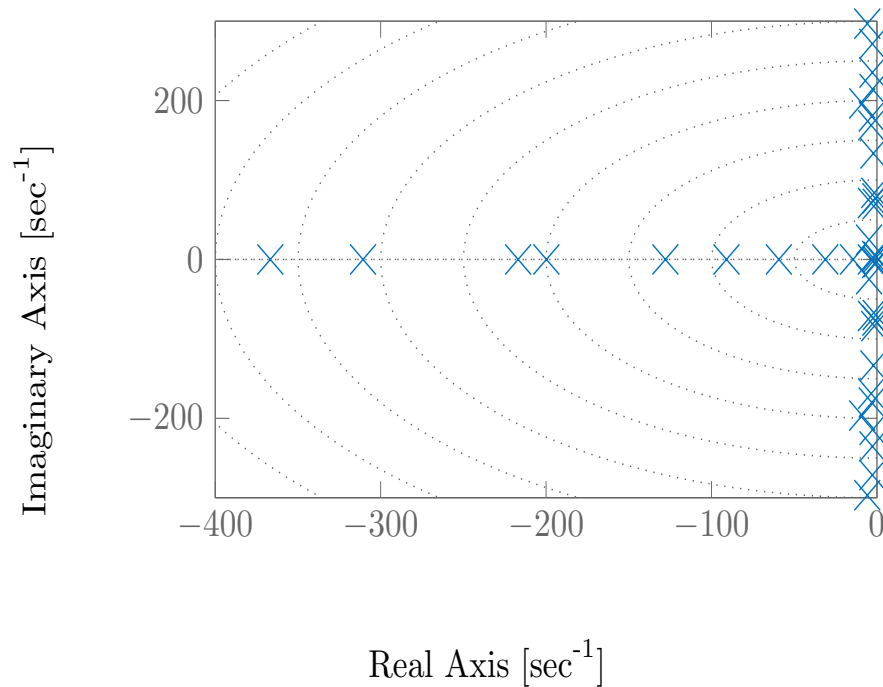


Figure A.1: Pole-Zero Map of the model dynamics.

B Quadcopter UAV Case-Study

Dimensios	$40 \times 40 \times 15$ cm
Weight	1000 g
Max Payload	300 g
Onboard Computer	Intel Aero Compute Board – Intel Atom x7-Z8750 Quad-core 64-bit 2.56 GH z processor 4 GB LPDDR3-1600 RAM
Cameras Intel RealSense Omnivision	640×480 @ 60 FPS or 1080p @ 30FPS VGA 640×480 @ 120 FPS

Table B.1: quadcopter specifications [177].

Bibliography

- [1] M. Kopf, E. Bullinger, H. G. Giessler, S. Adden, and R. Findeisen. Model predictive control for aircraft load alleviation: Opportunities and challenges. In *American Control Conference*, pages 2417–2424. IEEE, 2018.
- [2] J. H. Hansen, M. Duan, I. Kolmanovsky, and C.s E. Cesnik. Control allocation for maneuver and gust load alleviation of flexible aircraft. In *AIAA Scitech Forum*, page 1186, 2020.
- [3] C. D. Regan and C. V. Jutte. Survey of applications of active control technology for gust alleviation and new challenges for lighter-weight aircraft. Technical report, NASA TM-2012-216008, 2012.
- [4] R. Cavallaro and L. Demasi. Challenges, ideas and innovations of joined-wing configurations: a concept from the past, an opportunity for the future. *Progress in Aerospace Sciences*, 2016.
- [5] G. Cai, K. Lum, B. M. Chen, and T. H. Lee. A brief overview on miniature fixed-wing unmanned aerial vehicles. In *ICCA*, pages 285–290. IEEE, 2010.
- [6] F. Kendoul. Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics*, 29(2):315–378, 2012.
- [7] M. A. Kamel, A. T. Hafez, and X. Yu. A review on motion control of unmanned ground and aerial vehicles based on model predictive control techniques. *Engineering Science and Military Technologies*, 2(1):1939–1944, 2018. doi: 10.21608/ejmtc.2017.1059.1050.
- [8] C. Goerzen, Z. Kong, and B. Mettler. A survey of motion planning algorithms from the perspective of autonomous UAV guidance. *Journal of Intelligent and Robotic Systems*, 57(1-4):65, 2010.
- [9] P. Toth and D. Vigo. *Vehicle routing: problems, methods, and applications*. SIAM, 2014.
- [10] Anqi Xu, Chatavut Viriyasuthee, and Ioannis Rekleitis. Efficient complete coverage of a known arbitrary environment with applications to aerial operations. *Autonomous Robots*, 36(4):365–381, 2014.
- [11] R. Bormann, F. Jordan, J. Hampp, and M. Hägele. Indoor coverage path planning: survey, implementation, analysis. In *International Conference on Robotics and Automation*, pages 1718–1725. IEEE, 2018.
- [12] M. Mirzaei, F. Sharifi, B. W. Gordon, C. A. Rabbath, and Y. Zhang. Cooperative

- multi-vehicle search and coverage problem in uncertain environments. In *Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 4140–4145. IEEE, 2011.
- [13] E. J. Forsmo. Optimal path planning for unmanned aerial systems. Master’s thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology, 2012.
- [14] Z. Cai, S. Li, Y. Gan, R. Zhang, and Q. Zhang. Research on complete coverage path planning algorithms based on A* algorithms. *The Open Cybernetics Systemics Journal*, pages 418–426, 2014.
- [15] C. Luo and S. X. Yang. A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments. *IEEE Transactions on Neural Networks*, 19(7):1279–1298, 2008.
- [16] A. K. Jeyaraj. *A Model-Based Systems Engineering Approach for Efficient System Architecture Representation in Conceptual Design: A Case Study for Flight Control Systems*. PhD thesis, Concordia University, 2019.
- [17] U. Eren, A. Prach, B. B. Koçer, S. V. Raković, E. Kayacan, and B. Açıkmeşe. Model predictive control in aerospace systems: Current state and opportunities. *Journal of Guidance, Control, and Dynamics*, 40(7):1541–1566, 2017.
- [18] D. Mayne. Nonlinear model predictive control: Challenges and opportunities. In *Nonlinear model predictive control*, pages 23–44. Springer, 2000.
- [19] R. Findeisen and F. Allgöwer. An introduction to nonlinear model predictive control. *21st Benelux Meeting on Systems and Control*, 11:119–141, 2002.
- [20] R. Findeisen. *Nonlinear model predictive control: a sampled data feedback perspective*. PhD thesis, University of Stuttgart, 2006.
- [21] J. B. Rawlings and D. Q. Mayne. *Model predictive control: Theory and design*. Nob Hill Pub., 2009.
- [22] E. F. Camacho and C. B. Alba. *Model predictive control*. Springer Science & Business Media, 2013.
- [23] D. Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.
- [24] J. B. Rawlings, D. Q. Mayne, and M. Diehl. *Model predictive control: theory, computation, and design*, volume 2. Nob Hill Publishing, Madison, WI, 2017.
- [25] L. Grüne and J. Pannek. Nonlinear model predictive control. In *Nonlinear Model Predictive Control*, pages 45–69. Springer, 2017.
- [26] R. Findeisen, L. Imsland, F. Allgöwer, and B. A. Foss. State and output feedback nonlinear model predictive control: An overview. *European journal of control*, 9(2-3):190–206, 2003.

-
- [27] J. H. Lee. Model predictive control: Review of the three decades of development. *International Journal of Control, Automation and Systems*, 9(3):415, 2011.
- [28] S. Di Cairano and I. Kolmanovsky. Automotive applications of model predictive control. In *Handbook of Model Predictive Control*, pages 493–527. Springer, 2019.
- [29] L. Imsland, R. Findeisen, E. Bullinger, F. Allgöwer, and B. A. Foss. A note on stability, robustness and performance of output feedback nonlinear model predictive control. *Journal of Process Control*, 13(7):633–644, 2003.
- [30] S. Di Cairano and I. Kolmanovsky. Real-time optimization and model predictive control for aerospace and automotive applications. In *American Control Conference (ACC)*, pages 2392–2409. IEEE, 2018.
- [31] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.
- [32] M. Ibrahim, J. Matschek, B. Morabito, and R. Findeisen. Hierarchical model predictive control for autonomous vehicle area coverage. *IFAC-PapersOnLine*, 52(12):79–84, 2019.
- [33] M. Ibrahim, M. Kögel, C. Kallies, and R. Findeisen. Contract-based hierarchical model predictive control and planning for autonomous vehicle. In *21st IFAC World Congress*, Berlin, Germany, 2020.
- [34] M. Ibrahim, C. Kallies, and R. Findeisen. Learning-supported approximated optimal control for autonomous vehicles in the presence of state dependent uncertainties. In *19th European Control Conference*, Saint Petersburg, Russia, 2020.
- [35] M. Ibrahim, J. Matschek, B. Morabito, and R. Findeisen. Improved area covering in dynamic environments by nonlinear model predictive path following control. *IFAC-PapersOnLine*, 52(15):418–423, 2019.
- [36] K. F. Culligan. Online trajectory planning for UAVs using mixed integer linear programming. Master’s thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 2006.
- [37] N. Dadkhah and B. Mettler. Survey of motion planning literature in the presence of uncertainty: Considerations for UAV guidance. *Journal of Intelligent & Robotic Systems*, 65(1-4):233–246, 2012.
- [38] D. Q. Mayne, M. M. Seron, and S. V. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2): 219–224, 2005.
- [39] Directorate General for Mobility and Transport and Directorate General for Research and Innovation. Flightpath 2050: Europe’s vision for aviation. Technical report, European Commission, 2012.
- [40] NASA. Technology Roadmaps, TA 15: Aeronautics. Technical report, NASA, 2015.

- [41] A. Abbas, J. D. Vicente, and E. Valero. Aerodynamic technologies to improve aircraft performance. *Aerospace Science & Technology*, 28(1):100–132, 2013.
- [42] G. Carrier, O. Atinault, S. Dequ, J.L. Hantrais-Gervois, C. Liauzun, B. Paluch, A. M. Rodde, and C. Toussaint. Investigation of a strut-braced wing configuration for future commercial transport. In *28th Congress of the International Council of the Aeronautical Sciences*. ICAS Bonn, 2012.
- [43] N. Paletta. *Maneuver load controls, analysis and design for flexible aircraft*. PhD thesis, Università degli Studi di Napoli Federico II, 2011.
- [44] B. Imperatore and L. Vecchione. A flexible wing unmanned aerial research system. Technical report, SAE Technical Paper, 2009.
- [45] G. Frulla. Aeroelastic behaviour of a solar-powered high-altitude long endurance unmanned air vehicle (HALE-UAV) slender wing. *the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 218(3):179–188, 2004.
- [46] W. Su and C. Cesnik. Dynamic response of highly flexible flying wings. *AIAA Journal*, 49(2):324–339, 2011.
- [47] R. W. Beard and T. W. McLain. *Small unmanned aircraft: Theory and practice*. Princeton University Press, 2012.
- [48] A. T. Hafez. *Design and implementation of modern control algorithms for unmanned aerial vehicles*. PhD thesis, Queen’s University (Canada), 2014.
- [49] F. Afonso, J. Vale, É. Oliveira, F. Lau, and A. Suleman. A review on non-linear aeroelasticity of high aspect-ratio wings. *Progress in Aerospace Sciences*, 89: 40–57, 2017.
- [50] T. Noll, J. Brown, M. Perez-Davis, S. Ishmael, G. Tiffany, and M. Gaier. Investigation of the Helios prototype aircraft Mishap Volume I: Mishap Report. Technical report, NASA, 2004. URL https://www.nasa.gov/pdf/64317main_helios.pdf.
- [51] S. Haghghat, J. Martins, and H Liu. Aeroservoelastic design optimization of a flexible wing. *Journal of Aircraft*, 49(2):432–443, 2012.
- [52] L. Cary and J. Coyne. ICAO unmanned aircraft systems (UAS), circular 328. *UVS International, Blyenburgh & Co*, 2012:112–115, 2011.
- [53] E. Garone, S. Di Cairano, and I. Kolmanovsky. Reference and command governors for systems with constraints: A survey on theory and applications. *Automatica*, 75:306–328, 2017.
- [54] C. Kallies, M. Ibrahim, and R. Findeisen. Fallback approximated constrained optimal output feedback control under variable parameters. In *Portuguese Conference on Automatic Control*, pages 404–414. Springer, 2020.
- [55] N. van Duijkeren. *Online Motion Control in Virtual Corridors-for Fast Robotic Systems*. PhD thesis, KU Leuven, 2019.

-
- [56] U. Rosolia, S. De Bruyne, and A. G. Alleyne. Autonomous vehicle control: A nonconvex approach for obstacle avoidance. *IEEE Transactions on Control Systems Technology*, 25(2):469–484, 2016.
- [57] Y. Kuwata. *Trajectory planning for unmanned vehicles using robust receding horizon control*. PhD thesis, Massachusetts Institute of Technology, 2007.
- [58] P. Trodden and A. Richards. Multi-vehicle cooperative search using distributed model predictive control. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, page 7138, 2008.
- [59] A. Richards and J. P. How. Robust variable horizon model predictive control for vehicle maneuvering. *International Journal of Robust and Nonlinear Control*, 16(7):333–351, 2006.
- [60] T. Schouwenaars. *Safe trajectory planning of autonomous vehicles*. PhD thesis, Massachusetts Institute of Technology, 2006.
- [61] K. Culligan, M. Valenti, Y. Kuwata, and J. P. How. Three-dimensional flight experiments using on-line mixed-integer linear programming trajectory optimization. In *American Control Conference*, pages 5322–5327. IEEE, 2007.
- [62] S. S. Ponda. *Robust distributed planning strategies for autonomous multi-agent teams*. PhD thesis, Massachusetts Institute of Technology, 2012.
- [63] D. Mellinger and V. Kushleyev, A. and Kumar. Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams. In *International Conference on Robotics and Automation*, pages 477–483. IEEE, 2012.
- [64] B. L. Stevens, F. L. Lewis, and E. N. Johnson. *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons, 2015.
- [65] D. Simon. *Fighter Aircraft Maneuver Limiting Using MPC: Theory and Application*, volume 1881. Linköping University Electronic Press, 2017.
- [66] European Aviation Safety Agency. Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes CS-25, Amendment 19, Annex to ED Decision. *May*, 2017.
- [67] H-G Giessler, M. Kopf, P. Varutti, T. Faulwasser, and R. Findeisen. Model predictive control for gust load alleviation. *IFAC Proceedings Volumes*, 45(17): 27–32, 2012. ISSN 02734508.
- [68] P. Panyakeow. *Uncertainty Management for Aerial Vehicles: Coordination, Deconfliction, and Disturbance Rejection*. PhD thesis, Uni. of Washington, 2015.
- [69] R. G. Cook. *Robust Control of High Altitude Long Endurance Unmanned Aerial Vehicles using Novel Lift Effectors*. PhD thesis, Imperial College London, 2012.
- [70] A. Mannarino. *Nonlinear aeroservoelasticity: reduced order modeling and active control*. PhD thesis, Politecnico di Milano, Italy, 2016.

- [71] R. Rajamani. *Vehicle Dynamics and Control*. Springer Science & Business Media, 2011.
- [72] C. R. Cutler and B. L. Ramaker. Dynamic matrix control— a computer control algorithm. In *Joint Automatic Control Conference*, 17(72),1980.
- [73] J. Richalet, A. Rault, J. L. Testud, and J. Papon. Model predictive heuristic control. *Automatica*, 14(5):413–428, 1978.
- [74] D. W. Clarke, C. Mohtadi, and P. S. Tuffs. Generalized predictive control—part I. the basic algorithm. *Automatica*, 23(2):137–148, 1987.
- [75] B. Alrifaae. *Networked Model Predictive Control for Vehicle Collision Avoidance: Networked Model-based Predictive Control for Collision Avoidance of Vehicles*. PhD thesis, RWTH Aachen University, 2017.
- [76] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin. Provably safe and robust learning-based model predictive control. *Automatica*, 49(5):1216–1226, 2013.
- [77] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.
- [78] M. Kopf, H-G. Giessler, P. Varutti, T. Faulwasser, and R. Findeisen. On the effect of enforcing stability in model predictive control for gust load alleviation. In *American Control Conference*. IEEE, 2015.
- [79] A. Jadbabaie, J. Yu, and J. Hauser. Unconstrained receding-horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 46(5):776–783, 2001.
- [80] P. Scokaert, D. Q. Mayne, and J. B. Rawlings. Suboptimal model predictive control (feasibility implies stability). *IEEE Transactions on Automatic Control*, 44(3):648–654, 1999.
- [81] S. L. de Oliveira Kothare and M. Morari. Contractive model predictive control for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 45(6):1053–1071, 2000.
- [82] G. Pannocchia, J. B. Rawlings, and S. J. Wright. Conditions under which sub-optimal nonlinear MPC is inherently robust. *Systems & Control Letters*, 60(9): 747–755, 2011.
- [83] B. B. D. Luders. *Robust trajectory planning for unmanned aerial vehicles in uncertain environments*. PhD thesis, Massachusetts Institute of Technology, 2008.
- [84] S. V. Raković. Invention of prediction structures and categorization of robust MPC syntheses. *IFAC Proceedings Volumes*, 45(17):245–273, 2012.
- [85] F. A. Cuzzola, J. C. Geromel, and M. Morari. An improved approach for constrained robust model predictive control. *Automatica*, 38(7):1183–1189, 2002.
- [86] J. Löfberg. *Minimax approaches to robust model predictive control*. PhD thesis,

- Linköping University, 2003.
- [87] W. Langson, I. Chrysochoos, S. V. Raković, and D. Q. Mayne. Robust model predictive control using tubes. *Automatica*, 40(1):125–133, 2004.
- [88] A. G. Richards. *Robust constrained model predictive control*. PhD thesis, Massachusetts Institute of Technology, 2005.
- [89] R. Soloperto, M. A. Müller, S. Trimpe, and F. Allgöwer. Learning-based robust model predictive control with state-dependent uncertainty. *IFAC-PapersOnLine*, 51(20):442–447, 2018.
- [90] G. Pin, D. M. Raimondo, L. Magni, and T. Parisini. Robust model predictive control of nonlinear systems with bounded and state-dependent uncertainties. *IEEE Transactions on automatic control*, 54(7):1681–1687, 2009.
- [91] D. Q. Mayne, E. C. Kerrigan, E. J. Van Wyk, and P. Falugi. Tube-based robust nonlinear model predictive control. *International Journal of Robust and Nonlinear Control*, 21(11):1341–1353, 2011.
- [92] F. A. C. C. Fontes, S. V. Raković, and I. Kolmanovsky. Rigid tube model predictive control for linear sampled-data systems. *IFAC-PapersOnLine*, 50(1):9840–9845, 2017.
- [93] J. Yan and R. R. Bitmead. Incorporating state estimation into model predictive control and its application to network traffic control. *Automatica*, 41:595–604, 2005.
- [94] M. Ono and B. C. Williams. Decentralized chance-constrained finite-horizon optimal control for multi-agent systems. In *49th Conference on Decision and Control (CDC)*, pages 138–145. IEEE, 2010.
- [95] A. Mesbah, I. Kolmanovsky, and S. Di Cairano. Stochastic model predictive control. In *Handbook of Model Predictive Control*, pages 75–97. Springer, 2019.
- [96] J. Rogers. Stochastic model predictive control for guided projectiles under impact area constraints. *Journal of Dynamic Systems, Measurement, and Control*, 137(3), 2015.
- [97] S. V. Rakovic, A. R. Teel, D. Q. Mayne, and A. Astolfi. Simple robust control invariant tubes for some classes of nonlinear discrete time systems. In *45th Conference Decision Control*, page 6397–6402. IEEE, 2006.
- [98] Z. Wan and M. V. Kothare. Robust output feedback model predictive control using off-line linear matrix inequalities. *Journal of Process Control*, 12(7):763–774, 2002.
- [99] H. X. Araújo, A. G. S. Conceição, G. H. C. Oliveira, and J. Pitanga. Model predictive control based on LMIs applied to an omni-directional mobile robot. *IFAC Proceedings Volumes*, 44(1):8171–8176, 2011.
- [100] Y. Kuwata, A. Richards, and J. How. Robust receding horizon control using

- generalized constraint tightening. In *American Control Conference*, pages 4482–4487. IEEE, 2007.
- [101] G. Cao, E. M-K. Lai, and F. Alam. Gaussian process model predictive control of an unmanned quadrotor. *Journal of Intelligent & Robotic Systems*, 88(1):147–162, 2017.
- [102] A. Jadlovská and Š. Jajčičšin. Predictive control algorithms verification on the laboratory helicopter model. *Acta Polytechnica Hungarica*, 9(4):221–245, 2012.
- [103] F. Malchow and O. Sawodny. Model based feedforward control of an industrial glass feeder. *Control Engineering Practice*, 20(1):62–68, 2012.
- [104] M. Roozbehani, A. Megretski, and E. Feron. Optimization of Lyapunov invariants in verification of software systems. *IEEE Transactions on Automatic Control*, 58(3):696–711, 2013.
- [105] P. Zometa, M. Kögel, T. Faulwasser, and R. Findeisen. Implementation aspects of model predictive control for embedded systems. In *American Control Conference*, pages 1205–1210. IEEE, 2012.
- [106] G. Frison, D. K. M.e Kufoalor, L. Imsland, and J. B. Jørgensen. Efficient implementation of solvers for linear model predictive control on embedded devices. In *Conference on Control Applications*. 1954–1959, IEEE, 2014.
- [107] Y. Wang and S. Boyd. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2):267–278, 2009.
- [108] J. Nocedal and S. J. Wright. Sequential quadratic programming. *Numerical optimization*, pages 529–562, 2006.
- [109] R. Quirynen. *Numerical simulation methods for embedded optimization*. PhD thesis, KU Leuven, 2017.
- [110] I. Maurović, M. Baotić, and I. Petrović. Explicit model predictive control for trajectory tracking with mobile robots. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 712–717, 2011.
- [111] M. Vukov, A. Domahidi, H. J. Ferreau, M. Morari, and M. Diehl. Auto-generated algorithms for nonlinear model predictive control on long and on short horizons. In *52nd IEEE Conference on Decision and Control*, pages 5113–5118, 2013.
- [112] H. G. Bock, M. Diehl, E. Kostina, and J. P. Schlöder. Constrained optimal feedback control of systems governed by large differential algebraic equations. In *Real-Time PDE-constrained optimization*, pages 3–24. SIAM, 2007.
- [113] J. V. Frasch, L. Wirsching, S. Sager, and H. G. Bock. Mixed—level iteration schemes for nonlinear model predictive control. *IFAC Proceedings Volumes*, 45(17):138–144, 2012.
- [114] P. Zometa, M. Kögel, and R. Findeisen. μ AO-MPC: a free code generation tool for embedded real-time linear model predictive control. In *American Control*

- Conference*, pages 5320–5325. IEEE, 2013.
- [115] J. Lofberg. YALMIP: A toolbox for modeling and optimization in matlab. In *Computer Aided Control Systems Design*, pages 284–289. IEEE, 2004.
- [116] Gurobi Optimization. Gurobi optimizer 8.1. URL <http://www.gurobi.com>, 2019.
- [117] H. A. Åsgård. Combining methods of mathematical optimization and artificial intelligence for autonomous UAV mission planning and execution. Master’s thesis, Norwegian University of Science and Technology, 2017.
- [118] D. Bertsimas and R. Weismantel. *Optimization over integers*, volume 13. Dynamic Ideas, Belmont, 2005.
- [119] M. Kvasnica, P. Grieder, M. Baotić, and M. Morari. Multi-parametric toolbox (MPT). In *International Workshop on Hybrid Systems: Computation and Control*, pages 448–462. Springer, 2004.
- [120] B. Houska, H. J. Ferreau, and M. Diehl. ACADO toolkit—an open-source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011.
- [121] B. Houska, H. J. Ferreau, and M. Diehl. An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. *Automatica*, 47(10):2279–2285, 2011.
- [122] R. Verschueren, G. Frison, D. Kouzoupis, N. van Duijkeren, A. Zanelli, R. Quirynen, and M. Diehl. Towards a modular software package for embedded optimization. *IFAC-PapersOnLine*, 51(20):374–380, 2018.
- [123] J. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.
- [124] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):363–396, 2005.
- [125] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.
- [126] A. Kelman, S. Vichik, and F. Borrelli. BLOM: The Berkeley library for optimization modeling and nonlinear model predictive control. *Berkeley, CA*, <http://www.mpclab.net/Trac>, 2012.
- [127] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363, 2014.
- [128] G. Frison, H. H. B. Sørensen, B. Dammann, and J. B. Jørgensen. High-

- performance small-scale solvers for linear model predictive control. In *European Control Conference*, pages 128–133. IEEE, 2014.
- [129] G. Frison, D. Kouzoupis, T. Sartor, A. Zanelli, and M. Diehl. BLASFEO: Basic linear algebra subroutines for embedded optimization. *ACM Transactions on Mathematical Software (TOMS)*, 44(4):42, 2018.
- [130] A. Bemporad, M. Morari, and N. L. Ricker. Model predictive control toolbox user’s guide. *The Mathworks*, 2010.
- [131] J. Currie and D. I. Wilson. Lightweight model predictive control intended for embedded applications. *IFAC Proceedings Vol.*, 43(5):278–283, 2010.
- [132] F. Ullmann. FiOrdOs: A Matlab toolbox for C-code generation for first order methods. Master’s thesis, ETH Zürich, Switzerland, 2011.
- [133] S. Rivero, A. Battocchio, and G. Ferrari-Trecate. PnPMPC toolbox. *Identification and Control of Dynamic Systems Laboratory, Univ. of Pavia*, 2013.
- [134] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1, 2014.
- [135] A. Domahidi, E. Chu, and S. Boyd. ECOS: An SOCP solver for embedded systems. In *European Control Conference*. 3071–3076, IEEE, 2013.
- [136] A. Domahidi and J. Jerez. FORCES professional. *embotech GmbH* (<http://embotech.com/FORCES-Pro>), 2014.
- [137] P. Falugi, E. Kerrigan, and E. Van Wyk. Imperial college london optimal control software user guide (ICLOCS). *Department of Electrical and Electronic Engineering, Imperial College London, London, England, UK*, 2010.
- [138] P. E. Rutquist and M. M. Edvall. PROPT-MATLAB Optimal Control Software, Tomlab Optimization. *Inc., Pullman, WA*, page 65, 2010.
- [139] P. Kühn, J. Ferreau, J. Albersmeyer, C. Kirches, L. Wirsching, S. Sager, A. Potschka, G. Schulz, M. Diehl, and D. B. Leineweber. Muscod-II users manual. *University of Heidelberg*, 2007.
- [140] Z. K. Nagy. OptCon—an efficient tool for rapid prototyping of nonlinear model predictive control applications. In *AIChE Annual Meeting, 16–21*, 2008.
- [141] S. Ricci, A. De Gaspari, L. Riccobene, and F. Fonte. Design and wind tunnel test validation of gust load alleviation systems. In *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 1818, 2017.
- [142] J. J. Ryan and J. T. Bosworth. Current and future research in active control of lightweight and flexible structures using the X-56 aircraft. In *52nd Aerospace Sciences Meeting*, 2014.
- [143] F. Fonte, S. Ricci, and P. Mantegazza. Gust load alleviation for a regional aircraft through a static output feedback. *Journal of Aircraft*, 52(5):1559–1574, 2015.

-
- [144] Y. Bi, C. Xie, C. An, and C. Yang. Gust load alleviation wind tunnel tests of a large-aspect-ratio flexible wing with piezoelectric control. *Chinese Journal of Aeronautics*, 2017.
- [145] H-G. Giessler, M. Kopf, T. Faulwasser, P. Varutti, and R. Findeisen. Gust load alleviation based on model predictive control. In *International Forum on Aeroelasticity and Structural Dynamics*, 2013.
- [146] N. Fezans and H. D. Joos. Combined feedback and lidar-based feedforward active load alleviation. In *AIAA Atmospheric Flight Mechanics Conference*, 2017.
- [147] H. Li, Y. Zhao, and H. Hu. Adaptive maneuver load alleviation via recurrent neural networks. *J. of Guidance, Control, and Dynamics*, 40(7):1824–1831, 2017.
- [148] R. G. Cook, R. Palacios, and P. Goulart. Robust gust alleviation and stabilization of very flexible aircraft. *AIAA journal*, 2013.
- [149] M. Dillsaver, C. Cesnik, and I. Kolmanovsky. Gust load alleviation control for very flexible aircraft. In *AIAA Atmospheric Flight Mechanics Conference*, page 6368, 2011.
- [150] M. de Virgilio Pereira, I. Kolmanovsky, C. E. Cesnik, and F. Vetrano. Model predictive control architectures for maneuver load alleviation in very flexible aircraft. In *AIAA Scitech 2019 Forum*, page 1591, 2019.
- [151] M. de Virgilio Pereira, I. Kolmanovsky, and C. E. S. Cesnik. Model predictive control with constraint aggregation applied to conventional and very flexible aircraft. In *58th Conference on Decision and Control (CDC)*, pages 431–437. IEEE, 2019.
- [152] Y. Wang, A. Wynn, and R. Palacios. Model-predictive control of flexible aircraft dynamics using nonlinear reduced-order models. In *57th AIAA/ASCE/AHS/ASC Structures, Structural dynamics, and Materials Conference*, page 0711, 2016.
- [153] W. Gropengießer. Mathematische Flugzeugmodelle und deren Beschreibung. Technical Report, IBK Innovation GmbH & Co. KG, 2019.
- [154] Martin R Waszak and David K Schmidt. Flight dynamics of aeroelastic vehicles. *Journal of Aircraft*, 25(6):563–571, 1988.
- [155] Zvi Artstein. Stability, observability and invariance. *Journal of differential equations*, 44(2):224–248, 1982.
- [156] A. Ran and R. Vreugdenhil. Existence and comparison theorems for algebraic riccati equations for continuous-and discrete-time systems. *Linear Algebra and its Applications*, 99:63–83, 1988.
- [157] F. Fonte. *Design and validation of active gust load alleviation systems for aircraft*. PhD thesis, Politecnico di Milano, Italy, 2018.
- [158] W. Gropengießer. INFLIGHT: Review Meeting - HAP 4: Flugversuchsträger

- und -versuche. IBK Innovation GmbH & Co. KG, 2019.
- [159] M. Dillsaver, C. E. Cesnik, and I. Kolmanovsky. Trajectory control of very flexible aircraft with gust disturbance. In *AIAA Atmospheric Flight Mechanics (AFM) Conference*, page 4745, 2013.
- [160] I. Necoara, L. Ferranti, and T. Keviczky. An adaptive constraint tightening approach to linear model predictive control based on approximation algorithms for optimization. *Optimal Control Applications and Methods*, 36(5):648–666, 2015.
- [161] A. De Gaspari, A. Mannarino, and P. Mantegazza. A dual loop strategy for the design of a control surface actuation system with nonlinear limitations. *Mechanical Systems and Signal Processing*, 2017.
- [162] E. Galceran and M. Carreras. Coverage path planning for marine habitat mapping. In *Oceans*, pages 1–8. IEEE, 2012.
- [163] S. W. Moon and D. H. C. Shim. Study on path planning algorithms for unmanned agricultural helicopters in complex environment. *International Journal of Aeronautical and Space Sciences*, 10(2):1–11, 2009.
- [164] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1):33–55, 2016.
- [165] J. Bohren, T. Foote, J. Keller, A. Kushleyev, D. Lee, A. Stewart, P. Vernaza, J. Derenick, J. Spletzer, and B. Satterfield. Little Ben: The Ben Franklin racing team’s entry in the 2007 DARPA urban challenge. *Journal of Field Robotics*, 25(9):598–614, 2008.
- [166] A. Liniger and J. Lygeros. A viability approach for fast recursive feasible finite horizon path planning of autonomous RC cars. In *18th International Conference on Hybrid Systems: Computation and Control*, pages 1–10. ACM, 2015.
- [167] J. Karimi and S. H. Pourtakdoust. Optimal maneuver-based motion planning over terrain and threats using a dynamic hybrid PSO algorithm. *Aerospace Science and Technology*, 26(1):60–71, 2013.
- [168] D. J. Grymin, C. B. Neas, and M. Farhood. A hierarchical approach for primitive-based motion planning and control of autonomous vehicles. *Robotics and Autonomous Systems*, 62(2):214–228, 2014.
- [169] T. K. Lee, S. H. Baek, S. Y. Oh, and Y. H. Choi. Complete coverage algorithm based on linked smooth spiral paths for mobile robots. In *11th International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 609–614. IEEE, 2010.
- [170] T. M. Driscoll. Complete coverage path planning in an agricultural environment. Master’s thesis, Digital Repository@ Iowa State University, 2011.

-
- [171] G. Atinc. *Safe and reliable dynamic coverage control for multi-agent systems*. PhD thesis, University of Illinois, Urbana, Illinois, 2014.
- [172] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.
- [173] S. C. Pinto and R. J. M. Afonso. Risk constrained navigation using MILP-MPC formulation. *IFAC-PapersOnLine*, 50(1):3586–3591, 2017.
- [174] E. Galceran and M. Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous systems*, 61(12):1258–1276, 2013.
- [175] H. Choset. Coverage for robotics—a survey of recent results. *Annals of mathematics and artificial intelligence*, 31(1-4):113–126, 2001.
- [176] A. Yazici, G. Kirlik, O. Parlaktuna, and A. Sipahioglu. A dynamic path planning approach for multirobot sensor-based coverage considering energy constraints. *IEEE Transaction on Cybernetics*, 44(3):305–314, 2013.
- [177] Quanser. Autonomous Vehicles Research Studio, 2020. <https://www.quanser.com/products/qdrone/>. Accessed: 30.09.2020.
- [178] OptiTrack Motion Capture, 2020. <https://www.optitrack.com/>. Accessed: 30.09.2020.
- [179] T. Bähge, M. Kögel, S. Di Cairano, and R. Findeisen. Contract-based predictive control for modularity in hierarchical systems. *IFAC-PapersOnLine*, 51(20):499–504, 2018.
- [180] S. Lucia, M. Kögel, P. Zometa, D. E. Quevedo, and R. Findeisen. Predictive control, embedded cyberphysical systems and systems of systems—a perspective. *Annual Reviews in Control*, 41:193–207, 2016.
- [181] S. Lucia, M. Kögel, and R. Findeisen. Contract-based predictive control of distributed systems with plug and play capabilities. *IFAC-PapersOnLine*, 48(23):205–211, 2015.
- [182] S. Blasi, M. Kögel, and R. Findeisen. Distributed model predictive control using cooperative contract options. *IFAC-PapersOnLine*, 51(20):448–454, 2018.
- [183] C. Kallies, M. Ibrahim, and R. Findeisen. Approximated explicit infinite horizon constraint optimal control for systems with parametric uncertainties. In *IFAC World Congress*, Berlin, Germany, 2020.
- [184] M. Kögel and R. Findeisen. Fusing multiple time varying tubes for robust MPC. In *IFAC World Congress*, Berlin, Germany, 2020.
- [185] H. Hu, X. Feng, R. Quirynen, M. E. Villanueva, and B. Houska. Real-time tube MPC applied to a 10-state quadrotor model. In *American Control Conference (ACC)*, pages 3135–3140. IEEE, 2018.
- [186] J. Troelsen. Inflight: Interface Control Document. Technical report, messWERK,

2020.

- [187] R. Wittkopp. Flugexperimentelle Untersuchungen zum Widerstand und softwareunterstützte Widerstandsreduzierung am Ultraleichtflugzeug REMOS GX. Hochschule für Angewandte Wissenschaften Hamburg, 2010. Diplomarbeit.