LA COLLANA DELLA SCUOLA DI ALTA FORMAZIONE DOTTORALE ACCOGLIE LE MIGLIORI TESI DI DOTTORATO DELL'UNIVERSITÀ DEGLI STUDI DI BERGAMO, INSIGNITE DELLA DIGNITÀ DI STAMPA E SOTTOPOSTE A PROCEDURA DI *BLIND PEER REVIEW*.

This book presents advanced model predictive control methods for nonlinear systems, with a focus on robustness, changing setpoints, and data-driven modeling. After introducing the main theoretical foundations of nonlinear MPC, it develops new control schemes for perturbed systems, including a robust tracking MPC and a contraction-based MPC for regulation when standard terminal ingredients are difficult to compute. The book also explores Gaussian Process-based MPC and concludes with a biomedical application to blood glucose regulation in subjects with type 1 diabetes.

MARCO POLVER received his PhD in Control Engineering from the University of Bergamo (37^{th} cycle) in 2025. His doctoral research focused on model predictive control for nonlinear systems, with particular attention to robustness, data-driven approaches, and biomedical applications. Since 2025, he has been working on control algorithms for robotic systems, including model predictive control and invariant extended Kalman filtering for autonomous wheeled loaders, optimization-based whole-body control of articulated robots, and guidance, navigation, and control methods for quadcopters.

ISBN: 978-88-97253-51-8
DOI: 10.13122/978-88-97253-51-8

Marco Polver

ROBUST NMPC FOR NONLINEAR SYSTEMS

96

Collana della Scuola di Alta Formazione Dottorale
- 96 -

Marco Polver

ROBUST MPC FOR NONLINEAR SYSTEMS

UNIVERSITÀ DEGLI STUDI DI BERGAMO

2026

Collana della Scuola di Alta Formazione Dottorale

- 96 -

Collana della Scuola di Alta Formazione Dottorale

Diretta da Paolo Cesaretti

Ogni volume è sottoposto a *blind peer review*.

ISSN: 2611-9927

Sito web: https://aisberg.unibg.it/handle/10446/130100

Marco Polver

ROBUST MPC FORMULATIONS
for perturbed constrained nonlinear systems

Università degli Studi di Bergamo

2026

Robust MPC formulations for perturbed constrained nonlinear systems.
/ Marco Polver. – Bergamo :
Università degli Studi di Bergamo, 2026.
(Collana della Scuola di Alta Formazione Dottorale; 96)

ISBN: 978-88-97253-51-8
DOI: 10.13122/978-88-97253-51-8

Progetto grafico: Servizi Editoriali – Università degli Studi di Bergamo

via Salvecchio, 19
24129 Bergamo
Cod. Fiscale 80004350163
P. IVA 01612800167

https://aisberg.unibg.it/handle/10446/330905

List of Acronyms

AID automatic insulin delivery. 161, 162, 163, 180, 182

AP Artificial Pancreas. 161, 163, 166, 180

ARE algebraic Riccati equation. 14

BG blood glucose. 161, 162, 163, 166, 167

CGM continuous glucose monitor. 163, 166, 167

CLF control Lyapunov function. 10, 15, 19, 20, 23, 30, 33, 34, 42, 45, 57, 73, 74, 93, 99, 114, 122, 125, 179, 189

CSTR continuous stirred-tank reactor. 131, 149, 150, 156, 199

CVGA Control-Variability Grid Analysis. 174, 175, 197

FDA U.S. Food and Drug Administration. 3

GAS globally asymptotically stable. 94

GP Gaussian Processes. 2, 3, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 144, 151, 152, 153, 156, 161, 162, 163, 164, 165, 167, 168, 171, 172, 175, 176, 180, 181, 182, 196, 199

GP-MPC Gaussian Process-based Model Predictive Control. 130, 131, 138, 139, 142, 144, 145, 146, 147, 148, 149, 150, 151, 154, 155, 156, 157, 164, 176, 180, 181, 196, 197, 199

GPR Gaussian Process regression. 130, 133, 135, 163, 164, 165, 169, 176, 181

i.i.d. independent and identically distributed. 133, 137, 149, 166

IFAC International Federation of Automatic Control. 1

IMS information management system. 22

IS insulin sensitivity. 163, 167

ISpS input-to-state practically stable. 101, 108, 109, 111, 147, 149, 190

ISS input-to-state stable. 43, 45, 62, 64, 65, 67, 70, 72, 93, 109, 190

IT information technology. 22

LMI linear matrix inequality. 13, 15, 74, 75, 76

LPV Linear Parameter-Varying. 74

LTI Linear Time-Invariant. 75, 130

LTV Linear Time-Varying. 74, 75, 76, 85

MPC Model Predictive Control. 2, 3, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 41, 43, 44, 45, 46, 49, 50, 51, 52, 53, 54, 55, 56, 59, 60, 61, 62, 74, 75, 80, 82, 83, 87, 93, 95, 96, 97, 98, 99, 100, 107, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 122, 123, 124, 125, 126, 129, 130, 131, 136, 138, 139, 140, 142, 144, 146, 147, 149, 154, 155, 156, 161, 162, 163, 164, 170, 171, 173, 175, 177, 179, 180, 181, 182, 186, 192, 195, 196, 197, 199

NARX Nonlinear AutoRegressive eXogenous model. 166

NN neural networks. 130, 135, 163

PDF probability density function. 131, 164, 165, 191

PID Proportional-Integral-Derivative. 22, 163

PLC Programmable Logic Controllers. 22

QIH-MPC quasi-infinite horizon Model Predictive Control. 13, 16, 17, 18, 19, 20, 32, 93, 114, 118

RPI robust positively invariant set. 42, 55, 98, 99, 114, 116, 124, 144, 145, 179, 187, 190

RTO real-time optimizer. 22

SE-ARD squared exponential with automatic relevance determination. 133, 134, 135, 151, 165, 167, 196

SoD subset of data. 135, 168

T1DM Type 1 Diabetes Mellitus. 3, 161, 162, 163, 166, 167, 169, 175, 176, 180, 181

TEC-MPC terminal equality constraint Model Predictive Control. 16, 17, 18, 23, 93, 114

Table of Contents

Introduction

Motivation

Control has accompanied human beings throughout their entire history. Perhaps control has not always been considered a discipline, but its concepts have influenced science and the entire body of knowledge for millennia. It is claimed, for example, that before 2000 B.C. the control of irrigation systems was a well known art in Mesopotamia [43]. Later, Romans unwittingly used concepts from control theory in the design of their aqueducts, with the aim of keeping the water level constant. In his monograph "Politics", Chapter 3, Book 1, Aristotle writes:

"...if every instrument could accomplish its own work, obeying or anticipating the will of others...if the shuttle weaved and the pick touched the lyre without a hand to guide them, chief workmen would not need servants, nor masters slaves."

This last sentence not only explicates what the goal of control theory is, i.e. automatizing processes, but also provides a political clarification of this goal, namely letting all human beings gain in liberty, freedom, and quality of life. Many years later, in 1788, James Watt equipped his steam engine with a mechanical feedback governor that is by many considered to be the first known proportional controller to ever be implemented. The operation of governors was first theoretically described by James Clerk Maxwell, with his earliest writing dating back to 1868 [43]. Nonetheless, it was only in the 1920s and 1930s that control theory gained interest, coming to be an important research field during World War II. The war itself highlighted that many mathematical models employed for control reasons were not accurate enough. Humans had long known that nearly all physical systems exhibit nonlinearities, but despite this, classical control theory (often concerning linear systems) had until then been considered sufficient for the design of effective controllers. Consequently, the interest in nonlinear control rose, leading many to consider 1960 as the beginning year of the modern era of nonlinear control [11, 60]. In the same year, the first International Federation of Automatic Control (IFAC) congress was held in Moscow. Also in 1960 Kalman and Beltram published [62], where Lyapunov's analysis method of dynamic system stability, which dates back to the XIXth century, was first employed in the control context, for both linear and nonlinear systems.

We could go on listing achievements in the field of nonlinear system control over the past decades, but we will simply state that nonlinear control has become progressively more important over time, as the complexity of the systems to be controlled has increased, making it necessary to consider nonlinearities and model uncertainties to ensure the effectiveness of the applied control techniques.

The characteristics that a control technique should ideally have to control a complex nonlinear system include:

- It should be able to effectively control the system even with little information about its dynamics.
- It should ensure closed-loop stability, also when the system is perturbed.
- It should act safely, i.e. the closed-loop system under the employed control law should verify (possible) safety constraints.
- The operation of the system should be guaranteed also in case of setpoint changes.
- The controller itself should be simple to design and implement.

The last is obviously a utopian description of what a modern control technique should be like. Nonetheless, control theorists are striving to create new solutions that possess as many of the features mentioned in the last list as possible.
Among the control methods that the scientific and industrial communities find most promising there is Model Predictive Control (MPC) [22], a feedback optimization-based control method that employs a model of the system under control to make predictions about its state evolution and computes a control sequence that minimizes a performance index while ensuring that the closed-loop system verifies input and state constraints. Further details about MPC can be found in Chapter 1. However, MPC only complies with some of the points mentioned above. As a matter of fact, having a prediction model of the system (and hopefully an accurate one) requires having a lot of information about the system itself. Moreover, even though the idea behind MPC is intuitive, its design and implementation can become troublesome. Indeed, the good operation of an MPC depends on some stability-related ingredients, called *terminal ingredients*, whose computation is challenging for some families of nonlinear systems, consequently limiting the set of systems that can be controlled by an MPC in practice. Last but not least, robustness to perturbations and the ability of MPC to enable the closed-loop system to track changing setpoints are issues that are often addressed separately in the case of nonlinear systems, although it would be desirable to find a unique solution to both issues.
This book addresses some of the aforementioned issues. A robust MPC for tracking changing setpoints for perturbed nonlinear systems is first presented, followed by an MPC for regulation to a fixed setpoint that enables the control of perturbed nonlinear systems for which the design of a standard MPC would be challenging, if not impossible. Finally, the problem of knowing a mathematical model of the system is also addressed, providing a novel MPC based on Gaussian Processes (GP), which just requires the availability of data about the operation of the plant under control. Consistently with the latter scenario,

how predictive controllers based on GPs can be used for the control of biomedical systems is also shown, giving an example of the control of the glucose levels in diabetic subjects.
Finally yet importantly, this book provides an intuitive yet theoretically sound introduction to the usage of MPC for the control of nonlinear systems. Such an introduction can prove useful in itself and may also facilitate the understanding of the predictive controllers that constitute the theoretical and applicative contributions. For this reason, readers are advised to read the document from beginning to end, following the chosen chapter order, as it will ease the overall understanding of the manuscript.

Contributions and outline

The remainder of the book is organized as follows:

- **Chapter 1** reviews several families of MPC schemes for nonlinear systems. In detail, both the problems of regulating a system to a fixed setpoint and tracking changing setpoints are addressed. Finally, an overview of robust MPC schemes for perturbed nonlinear systems is provided.

- **Part I** covers the theoretical contributions.

  – **Chapter 2** presents a robust MPC for tracking changing setpoints for perturbed nonlinear systems, proves its stabilizing capabilities and shows its practical effectiveness on two simulated case studies.

  – **Chapter 3** describes a robust contraction-based MPC for the regulation of perturbed nonlinear systems for which the computation of standard terminal ingredients is difficult, proving also its stabilizing capabilities. The method is finally validated in simulation on two different systems.

  – **Chapter 4** introduces a robust contractive MPC for the regulation of perturbed nonlinear systems that are modeled by means of GPs, showing under what conditions such method inherits the stabilizing capabilities of the contraction-based MPC presented in Chapter 3 and validating the controller on a simulated case study.

- **Part II** covers the applicative contributions.

  – **Chapter 5** presents a GP-based zone MPC for the regulation of the subcutaneous glucose level of Type 1 Diabetes Mellitus (T1DM) subjects, validating the proposed controller on the in-silico subjects provided by the FDA-approved UVA/Padova simulator.

- **Chapter 6** draws some conclusions.
- **Appendices A-C** review the mathematical background of stability analysis, list the main properties of Gaussian distributions and describe the uncertain quadruple-tank system that is used to test the control performances of some of the controllers presented in this book.

Chapter 1. Model Predictive Control

MPC is an advanced optimization-based control method that has gained increasing prominence in both academia and industry. Its operation is as follows: considering a discrete-time system, at each time instant (i) a finite-horizon open-loop optimal control problem is solved, (ii) the first element of the optimal control sequence is applied to the system, and (iii) the loop is eventually closed by reading the new state of the system and solving a new finite-horizon open-loop optimal control problem starting from the new system state, in a receding-horizon fashion. The recent success of MPC over standard control techniques, e.g. PID controllers, can be attributed to the following reasons:

1. *The underlying idea is intuitive*. Indeed, the usage of a mathematical model of the system to make predictions over a finite time horizon, the choice of a control sequence that minimizes a performance-related objective function, and the closing of the loop by applying just the first part of the optimal control sequence, followed by the repetition of the same process starting from the new system state, remind us of the way human beings often take decisions. Chess players, for example, usually decide their moves by making predictions about the outcome they can get over a finite time horizon, and repeat the same decision-making process every time the opponent makes a new move, aware of the fact that the previously conceived sequence of moves might be suboptimal in the new chessboard configuration. The same happens when humans drive a car: they optimize the sequence of actions to be taken within a finite time to achieve a goal, and keep on updating such sequence based on the changing environmental conditions.

2. *It can control a wide variety of systems*. As far as a mathematical model of the system to be controlled is available, systems with relevant time delays, non-minimum phase systems, linear and nonlinear systems, as well as coupled multivariable systems can be effectively controlled by MPC.

3. *It can handle constraints*. Constraints are a fundamental part of many control problems, as they represent, among others, actuator saturation levels, limits on the control action gradient, safety conditions to be met, or bounds within which the employed system model is mathematically sound. As the MPC operation is based on the repeated solution of an optimization problem, constraints on the system states, inputs and outputs can be simply added to the optimal control problem to be solved at each time instant.

4. *It benefits from the availability of any information.* Differently from other control techniques, the knowledge of measurable disturbances, future reference values and possible bounded uncertainties can be leveraged to improve the control performances and ensure the closed-loop verification of the constraints.

The first articles to treat the main concepts about MPC date back to the 1960s, e.g. [82] and [117], which respectively deal with linear and nonlinear systems, but it was only in the late 1970s that MPC started to become popular in control engineering [22, 55]. In particular, MPC started to gain interest in the area of process control, whose slowness was compatible with the time it took to the technological resources of the time to solve optimization problems like those inherent in MPC. The academia became interested in the stability properties of MPC only later, with the first works investigating the stability properties of linear and nonlinear MPC dating back to the 1980s and early 1990s [18, 24, 47].
Today's stability theory of MPC is mainly based on the employment of the so-called *terminal ingredients*, namely ingredients of the optimal control problem related to the state predicted at the end of the prediction horizon, i.e. the terminal state, that are employed to guarantee closed-loop stability. The MPC schemes employing stability-related terminal ingredients can be mostly divided in two groups:

- MPC schemes with terminal equality constraint, namely a constraint that imposes the terminal state to be equal to the reference;
- MPC schemes whose objective function comprises a particular component related only to the terminal state, i.e. the terminal cost function, and an inequality constraint that imposes the terminal state to belong to a region with particular properties, i.e. the terminal region.

These concepts trace back to the 1990s. Indeed, comprehensive studies about the stability of MPC schemes employing terminal ingredients consistent with those of today trace back to those years, e.g. [103], which investigates the stability of MPC schemes with terminal equality constraint for continuous-time systems, or [25, 32], which treat the stability of schemes that employ a terminal cost function and a terminal inequality constraint. During the 1990s and the 2000s, schemes with either no terminal ingredients or reduced terminal ingredients were also presented, e.g. [6, 54, 85].
Up to this point, MPC schemes have been listed that aim at solving regulation problems, namely control problems in which the goal is to stabilize the system state at an equilibrium x_s. Nevertheless, this is not the only kind of control problem that can be solved by employing an MPC. In fact, at least five contexts can be identified, among many others, in which MPC has been successfully used:

- *Regulation*, that has already been described.

- *Tracking of changing setpoints*, namely the regulation to equilibrium setpoints that can abruptly change over time.

- *Path following*, which is the problem of following a predefined and known path.

- *Trajectory tracking*, that is similar to path following, but imposes time constraints, i.e. the path to follow is time-parameterized.

- *Zone control*, namely the regulation to a zone, when a fixed setpoint is not available or meaningful.

Given that this book mainly treats robust nonlinear MPC schemes for regulation and for tracking changing setpoints, the remainder of this chapter only investigates the stability theory related to such formulations, eventually providing also insight into robust MPC schemes, not covered so far.

1.1 MPC for regulation

We consider a nonlinear time-invariant discrete-time system described by the equation

$$x(k+1) = f\left(x(k), u(k)\right), \tag{1.1}$$

where $x \in \mathbb{R}^n$ is the system state, here considered to be known and accessible, and $u \in \mathbb{R}^m$ is the system input. The system state and input are subject to hard constraints, namely

$$x(k) \in \mathcal{X} \subset \mathbb{R}^n, u(k) \in \mathcal{U} \subset \mathbb{R}^m, \forall k \geq 0. \tag{1.2}$$

In regulation problems, the aim is to steer the system state to an equilibrium $x_s = f\left(x_s, u_s\right)$. Nevertheless, without loss of generality, the remainder of the current section considers the case in which the reference is the origin, which requires the equilibrium $0 = f(0,0)$ to exist. In the following, we investigate how to design different MPC formulations to accomplish such goal and present all the assumptions under which each scheme is able to ensure closed-loop stability. First, a generic formulation of the optimization problem that has to be solved at each instant $k \geq 0$ is provided, here denoted as $P_{N_p}(x(k))$:

$$\min_{\mathbf{u}(k)} V_{N_p}\left(x(k), \mathbf{u}(k)\right) \coloneqq \sum_{j=0}^{N_p-1} \ell\left(\hat{x}(j|k), u(j|k)\right) + V_f\left(\hat{x}(N_p|k)\right) \tag{1.3a}$$

$$\text{subject to } \hat{x}(0|k) = x(k), \tag{1.3b}$$

$$\hat{x}(j|k) = f\left(\hat{x}(j-1|k), u(j-1|k)\right), \forall j \in [1, N_p], \tag{1.3c}$$

$$\hat{x}(j|k) \in \mathcal{X}, \forall j \in [0, N_p], \tag{1.3d}$$

$$u(j|k) \in \mathcal{U}, \forall j \in [0, N_p - 1], \tag{1.3e}$$

$$\hat{x}(N_p|k) \in \mathcal{X}_f. \tag{1.3f}$$

In detail:

- $\mathbf{u}(k) \coloneqq \left\{u(0|k), \ldots, u(N_p-1|k)\right\}$ is the control sequence to be optimized;
- $\hat{x}(j|k) \coloneqq \phi\left(j; x(k), \mathbf{u}(k)\right)$;
- $N_p \in \mathbb{N} \setminus \{0\}$ is the prediction horizon;
- (1.3a) is the cost function that represents the control goal, and its main ingredients are the stage cost function $\ell(\cdot,\cdot)$ and the (possible) terminal cost function $V_f(\cdot)$;
- (1.3b) imposes the initial condition $\hat{x}(0|k)$ to be equal to the current state $x(k)$;
- (1.3c) imposes the predictions $\hat{x}(j|k)$ to be consistent with the system dynamics, defined by the function $f(\cdot,\cdot)$;
- (1.3d) and (1.3e) impose the state predictions $\hat{x}(j|k)$ and the chosen control actions $u(j|k)$ to verify the state and input constraints;
- (1.3f) is the stability-related terminal constraint that can be imposed on the terminal state $\hat{x}(N_p|k)$.

The optimal control sequence resulting from the solution of $P_{N_p}(x(k))$ is hereafter denoted as $\mathbf{u}^*(x(k))$, or $\mathbf{u}^*(k)$ for brevity, where $\mathbf{u}^*(k) = \left\{u^*(0|k), \ldots, u^*(N_p-1|k)\right\}$, while the optimal value of the cost function given by $\mathbf{u}^*(k)$ is $V^*_{N_p}(x(k)) \coloneqq V_{N_p}(x(k), \mathbf{u}^*(k))$. The control law implied by this MPC scheme is defined as $\kappa_{N_p}(x) \coloneqq u^*(0|k)$.
In the remainder of this section, we discuss under what conditions four different regulation MPC schemes can guarantee closed-loop stability. In detail, we analyze (i) the MPC for regulation with

terminal inequality constraint and terminal cost function, (ii) the MPC for regulation with terminal equality constraint, (iii) the MPC for regulation with terminal cost function and without terminal constraint, and finally (iv) the MPC for regulation without terminal ingredients.

Before delving into the details of all these schemes, we first provide some basic assumptions that are needed to ensure that the optimal control problem (1.3) has a solution.

Assumption 1.1: Continuity of the system dynamics and the cost function

The functions $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$, $\ell : \mathcal{X} \times \mathcal{U} \to \mathbb{R}_{\geq 0}$ *and* $V_f : \mathcal{X}_f \to \mathbb{R}_{\geq 0}$ *are continuous. Moreover,* $f(0,0) = 0$, $\ell(0,0) = 0$ *and* $V_f(0) = 0$.

Assumption 1.2: Properties of the constraint sets

The sets $\mathcal{X}$ *and* $\mathcal{U}$ *are closed sets that contain the origin.*

Assumptions 1.1 and 1.2 are necessary but not sufficient conditions for the feasibility of $P_{N_p}(x(k))$. Missing conditions are provided in the next subsections, as they change depending on the MPC formulation.

1.1.1 MPC for regulation with terminal inequality constraint and terminal cost function

Before analyzing how the inclusion of a terminal inequality constraint and a terminal cost function can ensure closed-loop stability, let us briefly explain the method that is usually employed in the literature to prove the stability of a closed-loop system under a generic MPC scheme. First, given the receding horizon principle, that is solving an optimal control problem, applying the first element of the resulting control sequence, and repeating the loop starting from the new state of the system, it becomes clear that a critical point of this process resides in each beginning of a new cycle. Indeed, before ensuring that the closed-loop system state stabilizes at the desired equilibrium, we would like to be sure that, starting from an initial condition $x(0) \in \mathcal{X}$ that makes $P_{N_p}(x(0))$ feasible, $P_{N_p}(x(k))$ remains feasible $\forall k > 0$. This property is called *recursive feasibility* and is a fundamental property that has to be guaranteed by design.

Secondly, given that standard stability analysis methods, like the analysis of the eigenvalues of the closed-loop system, cannot be exploited in the analysis of the stability of a closed-loop system under an MPC, the MPC literature usually recurs to Lyapunov stability theory, whose details can be found in the Appendix A.4. From a practical point of view, what we do is proving that the optimal cost $V^*_{N_p}(\cdot)$ is a Lyapunov function, namely that there exist two $\mathcal{K}_\infty$ (or $\mathcal{K}$) functions $\alpha_1(\cdot)$ and $\alpha_2(\cdot)$ and a continuous, positive definite ($\mathcal{PD}$) function $\alpha_3(\cdot)$ such that, for any state $x \in \mathcal{X}$ that makes $P_{N_p}(x)$

feasible, verify the following conditions:

$$V^*_{N_p}(x) \geq \alpha_1(\|x\|)$$
$$V^*_{N_p}(x) \leq \alpha_2(\|x\|)$$
$$V^*_{N_p}(f(x, \kappa_{N_p}(x))) - V^*_{N_p}(x) \leq -\alpha_3(\|x\|).$$

The function families $\mathcal{K}_\infty$ and $\mathcal{K}$ (together with $\mathcal{KL}$ and $\mathcal{PD}$) are comparison function families. As these are widely used in the proofs of stability of MPC, their properties are provided in the Appendix A.1.

In the following, we therefore provide the conditions under which the inclusion of a terminal inequality constraint and a terminal cost function in the MPC optimal control problem ensure recursive feasibility and that $V^*_{N_p}(\cdot)$ is a Lyapunov function.

The next assumptions are needed to ensure that the function $V^*_{N_p}(\cdot)$ is a Lyapunov function.

Assumption 1.3: Terminal ingredient properties

The terminal constraint set $\mathcal{X}_f$ is compact and contains the origin. Moreover, there exists a locally stabilizing control law $\kappa_f(\cdot)$ such that $\kappa_f(x) \in \mathcal{U}$ $\forall x \in \mathcal{X}_f$ and

$$f\left(x, \kappa_f(x)\right) \in \mathcal{X}_f,$$

$$V_f\left(f\left(x, \kappa_f(x)\right)\right) - V_f\left(x\right) \leq -\ell\left(x, \kappa_f(x)\right).$$

Assumption 1.4: Stage cost function and terminal cost function bounds

There exist $\mathcal{K}_\infty$ functions $\alpha_\ell(\cdot)$ and $\alpha_f(\cdot)$ that verify the following conditions:

$$\ell(x, u) \geq \alpha_\ell(\|x\|), \forall (x, u) \in \mathcal{X} \times \mathcal{U},$$

$$V_f\left(x\right) \leq \alpha_f(\|x\|), \forall x \in \mathcal{X}_f.$$

Assumption 1.4 poses non-restrictive conditions on the choice of the stage cost function $\ell(\cdot, \cdot)$ and the terminal cost function $V_f(\cdot)$. Indeed, the majority of the MPC implementations employs quadratic stage and terminal cost functions. The compactness of $\mathcal{X}_f$ required by Assumption 1.3, together with Assumptions 1.1 and 1.2, ensures that $P_{N_p}\left(x(k)\right)$ has a solution (we refer to [126, Proposition 2.4] for further details). However, the other conditions posed by Assumption 1.3 are harder to verify, as they require $\mathcal{X}_f$ to be a positively invariant set for the closed-loop system under the local control law $\kappa_f(\cdot)$ and $V_f(\cdot)$ to be a control Lyapunov function (CLF) in $\mathcal{X}_f$. While these conditions are easily verified for controllable linear systems, designing such ingredients can become challenging, if not impossible, for some nonlinear systems.

We now introduce the theorem that explicates how Assumptions 1.2-1.4 ensure recursive feasibility and that the origin is asymptotically stable for the closed-loop system under an MPC with terminal inequality constraint and terminal cost function.

Theorem 1.1: Stability of the MPC with terminal inequality constraint and terminal cost function

Under Assumptions 1.1-1.4, the MPC for regulation with terminal inequality constraint and terminal cost function is recursively feasible and the origin is asymptotically stable for the closed-loop system.

Proof. We need to prove that $V_{N_p}^*(\cdot)$ is a Lyapunov function. In order to do so, we first prove that $V_{N_p}^*(\cdot)$ has a lower bound $\alpha_1(\cdot)$ and an upper bound $\alpha_2(\cdot)$, and then prove the descent property of $V_{N_p}^*(\cdot)$ and the recursive feasibility of this MPC scheme.

Lower and upper bounds for $V_{N_p}^*(\cdot)$. The lower bound for $V_{N_p}^*(\cdot)$ is immediately found. Indeed, defining the region of feasible states $\mathcal{X}_{N_p} \subseteq \mathcal{X}$, i.e. the region of attraction of the MPC, we have that $V_{N_p}^*(x) \geq \ell(x, u)$ for any $x \in \mathcal{X}_{N_p}$. Moreover, Assumption 1.4 ensures that a function $\alpha_\ell(\cdot) \in \mathcal{K}_\infty$ exists, such that $\ell(x, u) \geq \alpha_\ell(\|x\|)$ for any $x \in \mathcal{X}_{N_p}$. Therefore, $V_{N_p}^*(x) \geq \alpha_\ell(\|x\|)$ $\forall x \in \mathcal{X}_{N_p}$. The proof of the existence of an upper bound $\alpha_2(\cdot) \in \mathcal{K}_\infty$ is provided in [126, Propositions 2.15-2.18]. In particular, Proposition 2.18 proves that, for any $x \in \mathcal{X}_f$, $V_{N_p}^*(x) \leq V_f(x)$, while Proposition 2.16 proves that such local bound implies the existence of a function $\alpha_2(\cdot) \in \mathcal{K}_\infty$ such that $V_{N_p}^*(x) \leq \alpha_2(\|x\|)$ $\forall x \in \mathcal{X}_{N_p}$.

Descent property of $V_{N_p}^*(\cdot)$ and recursive feasibility. Consider a feasible state $x(k) \in \mathcal{X}_{N_p}$ and the optimal control sequence $\mathbf{u}^*(k) = \left\{u^*(0|k), \ldots, u^*(N_p - 1|k)\right\}$ resulting from the solution of $P_{N_p}(x(k))$. We define the resulting optimal state sequence as

$$\hat{\mathbf{x}}^*(k) := \left\{\hat{x}^*(0|k), \hat{x}^*(1|k), \ldots, \hat{x}^*(N_p|k)\right\}.$$

Note that $\hat{x}^*(0|k) = x(k)$ and that $x(k+1) = f\left(x(k), \kappa_{N_p}(x(k))\right) = \hat{x}^*(1|k)$. We now need to show that $P_{N_p}(x(k+1))$ is feasible and that there exists $\alpha_3(\cdot) \in \mathcal{PD}$ such that $V_{N_p}^*(x(k+1)) - V_{N_p}^*(x(k)) \leq -\alpha_3(\|x\|)$. In order to do so, given that $V_{N_p}^*(x(k+1))$ cannot be computed before actually solving $P_{N_p}(x(k+1))$, we choose a feasible control sequence

$$\mathbf{u}^+(k+1) := \left\{u^*(1|k), \ldots, u^*(N_p - 1|k), \kappa_f\left(\hat{x}^*(N_p|k)\right)\right\}$$

as candidate solution to $P_{N_p}(x(k+1))$, and exploit it to prove recursive feasibility and the desired descent property.

Recursive feasibility is easily proved by noting that the candidate state sequence $\hat{\mathbf{x}}^+(k+1)$ is given by

$$\hat{\mathbf{x}}^+(k+1) = \left\{\hat{x}^*(1|k), \ldots, \hat{x}^*(N_p|k), f\left(\hat{x}^*(N_p|k), \kappa_f\left(\hat{x}^*(N_p|k)\right)\right)\right\}.$$

Therefore, given that $\hat{x}^*(1|k), \ldots, \hat{x}^*(N_p|k)$ come from the solution of $P_{N_p}(x(k))$, they verify the state constraints. Moreover, provided that $\hat{x}^*(N_p|k) \in \mathcal{X}_f$, and given that the local control law $\kappa_f(\cdot)$ is feasible in $\mathcal{X}_f$, we have that $f(\hat{x}^*(N_p|k), \kappa_f\left(\hat{x}^*(N_p|k)\right)) \in \mathcal{X}_f$ due to Assumption 1.3, consequently proving that all the constraints of $P_{N_p}(x(k+1))$ are verified.

We now prove that there exists a function $\alpha_3(\cdot) \in \mathcal{PD}$ such that $V_{N_p}(x(k+1), \mathbf{u}^+(k+1)) - V^*_{N_p}(x(k)) \leq -\alpha_3(\|x\|)$. We start by explicating the terms that compose such subtraction, i.e.

$$\begin{aligned}
V_{N_p}(x(k+1), &\mathbf{u}^+(k+1)) - V^*_{N_p}(x(k)) = \sum_{j=1}^{N_p-1} \ell\left(\hat{x}^*(j|k), u^*(j|k)\right) \\
&+ \ell\left(\hat{x}^*(N_p|k), \kappa_f\left(\hat{x}^*(N_p|k)\right)\right) + V_f\left(f\left(\hat{x}^*(N_p|k), \kappa_f\left(\hat{x}^*(N_p|k)\right)\right)\right) \\
&- \sum_{j=0}^{N_p-1} \ell\left(\hat{x}^*(j|k), u^*(j|k)\right) - V_f\left(\hat{x}^*(N_p|k)\right) \\
&= \ell\left(\hat{x}^*(N_p|k), \kappa_f\left(\hat{x}^*(N_p|k)\right)\right) + V_f\left(f\left(\hat{x}^*(N_p|k), \kappa_f\left(\hat{x}^*(N_p|k)\right)\right)\right) \\
&- \ell\left(\hat{x}^*(0|k), u^*(0|k)\right) - V_f\left(\hat{x}^*(N_p|k)\right).
\end{aligned}$$

Provided that $\hat{x}^*(0|k) = x(k)$ and that

$$V_f\left(f\left(\hat{x}^*(N_p|k), \kappa_f\left(\hat{x}^*(N_p|k)\right)\right)\right) - V_f\left(\hat{x}^*(N_p|k)\right) \leq -\ell\left(\hat{x}^*(N_p|k), \kappa_f\left(\hat{x}^*(N_p|k)\right)\right)$$

due to Assumption 1.3, we finally get that

$$V_{N_p}\left(x(k+1), \mathbf{u}^+(k+1)\right) - V^*_{N_p}(x(k)) \leq -\ell\left(x(k), u^*(0|k)\right).$$

Since $\ell\left(x(k), u^*(0|k)\right) \geq \alpha_\ell(\|x(k)\|)$ thanks to Assumption 1.4, the last inequality becomes

$$V_{N_p}\left(x(k+1), \mathbf{u}^+(k+1)\right) - V^*_{N_p}(x(k)) \leq -\alpha_\ell(\|x(k)\|).$$

Due to optimality, we have that $V^*_{N_p}(x(k+1)) \leq V_{N_p}(x(k+1), \mathbf{u}^+(k+1))$, leading to

$$V^*_{N_p}(x(k+1)) - V^*_{N_p}(x(k)) \leq -\alpha_\ell(\|x(k)\|),$$

which confirms that $V^*_{N_p}(\cdot)$ is a Lyapunov function defined in $\mathcal{X}_{N_p}$. As affirmed in the Appendix A.4, the existence of a Lyapunov function defined in $\mathcal{X}_{N_p}$ is a sufficient condition for the asymptotic stability of the origin for the closed-loop system. ■

The MPC scheme presented in this subsection is often referred to as *quasi-infinite horizon Model Predictive Control (QIH-MPC)* [55], as the employment of an approximation of the infinite horizon optimal value function $V_\infty(\cdot)$ as terminal cost function $V_f(\cdot)$ allows one to use the MPC with terminal inequality constraint and terminal cost function as an approximation to the infinite horizon constrained optimal controller.

It is important to note that the feasibility of a QIH-MPC requires the terminal set $\mathcal{X}_f$ to be reachable in N_p steps from the current state $x(k)$. However, if such terminal set is small or the system is particularly complex, such condition might require to employ a large prediction horizon N_p, thus increasing the computational burden of the MPC. In such cases, a possible solution is that of replacing the optimal control problem (1.3) with the following one:

$$
\begin{aligned}
\min_{\mathbf{u}(k)} V_{N_p}\left(x(k), \mathbf{u}(k)\right) \coloneqq & \sum_{j=0}^{N_c-1} \ell\left(\hat{x}(j|k), u(j|k)\right) + \sum_{j=N_c}^{N_p-1} \ell(\hat{x}(j|k), \kappa_f(\hat{x}(j|k))) \\
& + V_f\left(\hat{x}(N_p|k)\right)
\end{aligned} \tag{1.7a}
$$

$$
\text{subject to } \hat{x}(0|k) = x(k), \tag{1.7b}
$$

$$
\hat{x}(j|k) = f\left(\hat{x}(j-1|k), u(j-1|k)\right), \forall j \in [1, N_c], \tag{1.7c}
$$

$$
\hat{x}(j|k) = f\left(\hat{x}(j-1|k), \kappa_f(\hat{x}(j-1|k))\right), \forall j \in [N_c+1, N_p], \tag{1.7d}
$$

$$
\hat{x}(j|k) \in \mathcal{X}, \forall j \in [0, N_p], \tag{1.7e}
$$

$$
u(j|k) \in \mathcal{U}, \forall j \in [0, N_p - 1], \tag{1.7f}
$$

$$
\kappa_f(\hat{x}(j|k)) \in \mathcal{U}, \forall j \in [N_c, N_p - 1], \tag{1.7g}
$$

$$
\hat{x}(N_p|k) \in \mathcal{X}_f. \tag{1.7h}
$$

In this formulation, a *control horizon* $N_c < N_p$ is considered, therefore optimizing the cost function value with respect to the shorter control sequence $\mathbf{u}(k) \coloneqq \{u(0|k), \ldots, u(N_c-1|k)\}$ and relying on the control law $\kappa_f(\cdot)$ to make the predictions within the window $[N_c, N_p]$ and reach the terminal set $\mathcal{X}_f$ after N_p steps. If a control law $\kappa_f(\cdot)$ verifying (1.7g) by design can be computed, the optimal control problem formulated in (1.7) can enlarge the domain of attraction of the MPC without prohibitively increasing its computational burden, as proved in [95].

Nevertheless, for both the formulations (1.3) and (1.7), the computation of a terminal cost function $V_f(\cdot)$, a terminal region $\mathcal{X}_f$ and a local control law $\kappa_f(\cdot)$ that verify Assumption 1.3 can be challenging for some systems. Indeed, the local control law $\kappa_f(\cdot)$ is often chosen to be a linear controller designed to make the linearization of the original system at the origin asymptotically stable, while the terminal cost function is chosen as a quadratic function whose weight matrix comes from the solution of a linear matrix inequality (LMI) or is the solution to the algebraic Riccati equation (ARE), still based on a local linearization of the system. However, if the (A, B) couple obtained from the local linearization of the system is not fully controllable and the uncontrollable states of the linearization are not asymptotically stable, such techniques for computing the terminal ingredients cannot be employed. In those cases, under some further assumptions, MPC schemes with terminal equality constraint can be a valid alternative.

Remark 1.1
Examples of algorithms for the computation of the aforementioned terminal ingredients can be found in [25] for continuous-time systems and in [79] for discrete-time systems. In particular, a simplified formulation of the method from [79] is given next.

Computation of the terminal ingredients of an MPC for the regulation of nonlinear systems

In the following, we assume to deal with a discrete-time time-invariant nonlinear system described by the state equation $x(k+1) = f(x(k), u(k))$ and that is subject to the following state and input constraints:

$$x(k) \in \mathcal{X},\ u(k) \in \mathcal{U}, \quad \forall k \geq 0.$$

Without loss of generality, we consider the problem of controlling the system state to the origin.

In standard MPC formulations, recursive feasibility and closed-loop stability are ensured by imposing that the state predicted at the end of the prediction horizon belongs to an invariant set $\mathcal{X}_f \subseteq \mathcal{X}$, where a local feasible and stabilizing control law $u = \kappa_f(x)$ is supposed to exist [126]. Hereafter, the dynamics of the system under $\kappa_f(x)$ is referred to as $x(k+1) = f_{cl}(x) \coloneqq f\left(x, \kappa_f(x)\right)$. Moreover, a terminal cost function $V_f(x)$ is added to the cost function of the optimization problem to be solved, which must be positive definite and such that

$$V_f\left(f_{cl}(x)\right) - V_f(x) \leq -\ell(x, \kappa_f(x)),\ \forall x \in \mathcal{X}_f, \tag{1.8}$$

where $\ell(x, u)$ is the stage cost function employed in the MPC.

As often done in the literature, we assume to have a quadratic stage cost function $\ell(x, u) = x^\top Q x + u^\top R u$, to employ a quadratic terminal cost function $V_f(x) = x^\top P x$, with P symmetric and positive definite, and a local linear controller $\kappa_f(x) = Kx$. We also assume to build the terminal set $\mathcal{X}_f$ as a sublevel set of the terminal cost function $V_f(x)$, i.e. $\mathcal{X}_f \coloneqq \{x : \ x^\top P x \leq \rho\}$, for some $\rho > 0$.

Remark 1.2

In MPC schemes for the regulation of nonlinear systems that employ both a terminal cost function and a terminal inequality constraint, it is common to employ sublevel sets of the terminal cost function $V_f(\cdot)$ as terminal sets. Indeed, given that $V_f(\cdot)$ is required to be a CLF, any of its sublevel sets $\{x : V_f(x) \leq \rho\}$, with $\rho > 0$, is a positively invariant set. Therefore, it is sufficient to find the maximum value of ρ for which the control law $\kappa_f(\cdot)$ is feasible for all $x \in \{x : V_f(x) \leq \rho\}$.

Assuming to have a controllable couple (A, B) given by the linearization of the system at the origin, and defining the linearization error as $e(x) = f_{cl}(x) - A_K x$, where $A_K = A + BK$, the condition (1.8) rewrites as

$$(A_K x + e(x))^\top P (A_K x + e(x)) - x^\top P x \leq -x^\top \left(Q + K^\top P K\right) x,$$

or, equivalently,

$$x^\top \left(A_K^\top P A_K - P + Q + K^\top P K\right) x + e(x)^\top P e(x) + 2x^\top A_K^\top P e(x) \leq 0, \tag{1.9}$$

which must hold true for all $x \in \mathcal{X}_f$.

Assuming that the contribution of the linearization error is bounded, as done in [79], i.e.

$$e(x)^\top P e(x) + 2x^\top A_K^\top P e(x) \leq \kappa x^\top P x, \ \forall x \in \mathcal{X}_f, \tag{1.10}$$

for some $\kappa \in (0, 1)$, condition (1.9) can be replaced by the matrix inequality

$$A_K^\top P A_K - (1-\kappa) P + Q + K^\top P K \leq 0. \tag{1.11}$$

Finally, by using the Schur complement and the substitution of variables $S = P^{-1}$ and $O = KP^{-1}$, (1.11) can be rewritten as an LMI:

$$\begin{bmatrix} (1-\kappa)S & (AS+BO)^\top & S & O^\top \\ AS+BO & S & 0 & 0 \\ S & 0 & Q^{-1} & 0 \\ O & 0 & 0 & R^{-1} \end{bmatrix} \geq 0. \tag{1.12}$$

For details about the usage of LMIs in control theory, we refer to [20].

1.1.2 MPC for regulation with terminal equality constraint

The optimal control problem formulated in (1.3) is still consistent with the regulation MPC scheme with terminal equality constraint. It is sufficient to define the terminal ingredients as follows:

$$\mathcal{X}_f = \{0\},$$

$$V_f(x) = 0.$$

As you can see, the terminal region degenerates to a point, which has an impact on the conditions that have to be met to enable the implementation of an MPC scheme with terminal equality constraint that guarantees recursive feasibility and stability of the closed loop. Moreover, setting the terminal cost function as the zero function is sensible. Indeed, the value taken by $V_f(x)$ is always zero when the optimal control problem $P_{N_p}(x)$ is feasible, as its feasibility requires the terminal equality constraint to be verified.

Now, let us analyze the assumptions under which the previously discussed QIH-MPC was guaranteed to be recursively feasible and to stabilize the closed-loop system, looking for conditions that might be inconsistent with the currently investigated scheme with terminal equality constraint. Note that, by choosing $\kappa_f(x) = 0$, all the conditions of Assumptions 1.3 and 1.4 are met also in the current case, therefore allowing us to reuse such assumptions also to prove the recursive feasibility and stability of the MPC with terminal equality constraint, hereafter abbreviated as TEC-MPC.

Hence, before introducing the theorem that confirms such properties, we analyze the proof of Theorem 1.1 to understand whether the aforementioned theorem can be reused in the currently discussed context. We note that, with the aforementioned choice of defining $\kappa_f(x) = 0$, Assumptions 1.2-1.4 are enough to prove the existence of a lower bound for $V^*_{N_p}(\cdot)$, the descent property of $V^*_{N_p}(\cdot)$ and the recursive feasibility of $P_{N_p}(\cdot)$. Nonetheless, these assumptions are insufficient to prove the existence of an upper bound for $V^*_{N_p}(\cdot)$ in $\mathcal{X}_{N_p}$. As a consequence, we need to make a further assumption.

Assumption 1.5: Weak controllability

A function $\alpha(\cdot) \in \mathcal{K}_\infty$ exists, such that

$$V^*_{N_p}(x) \leq \alpha(\|x\|) \; \forall x \in \mathcal{X}_{N_p}.$$

Let us discuss the meaning of Assumption 1.5, as its impact on the MPC design process is non-negligible. This assumption is limiting the operation of a TEC-MPC to all those states that can be steered to the origin in N_p steps and requires the cost of such movement to be bounded from above by a $\mathcal{K}_\infty$ function. While the second consequence is exactly what we wanted to obtain to close the proof of stability of the TEC-MPC, the first consequence is a clear limitation of the currently discussed scheme. As a matter of fact, the result of these considerations can be summarized in the following two statements:

- In order for a TEC-MPC to have the same region of attraction $\mathcal{X}_{N_p}$ of a QIH-MPC, it is necessary to impose a longer prediction horizon.

- In case of stabilizable but not controllable systems, a TEC-MPC cannot be correctly implemented.

Remark 1.3
It is necessary to further discuss the last point of the previous bullet list. In particular, we want to justify the possible inconsistencies between the here-presented theoretical concepts and the results that one could get in practice when implementing a TEC-MPC. In fact, while it is true that a stabilizable but uncontrollable system cannot get exactly to the origin in a finite number of steps, a practical implementation of a TEC-MPC might still result to be feasible. This is simply due to the fact that the available solvers evaluate equality constraints with a certain tolerance.

Yet, the last bullet list should not suggest that MPC schemes with terminal equality constraint are rarely used. In fact, they constitute a practical choice whenever the design of a QIH-MPC is troublesome. Now that Assumption 1.5 has been formulated and discussed in detail, we are ready to introduce the theorem that proves the recursive feasibility and the stabilizing properties of the TEC-MPC.

Theorem 1.2: Stability of the MPC with terminal equality constraint
Under Assumptions 1.1-1.5, the MPC for regulation with terminal equality constraint is recursively feasible and the origin is asymptotically stable for the closed-loop system.

Proof. The same arguments used to prove the existence of a lower bound function for $V^*_{N_p}(\cdot)$, the descent property of $V^*_{N_p}(\cdot)$ and the recursive feasibility of the controller can be taken from the Proof of Theorem 1.1. The upper bound function is provided by Assumption 1.5, by which we have that $V^*_{N_p}(x) \leq \alpha(\|x\|)\ \forall x \in \mathcal{X}_{N_p}$.
As a consequence, it can be stated that the MPC for regulation with terminal equality constraint is recursively feasible and the origin is asymptotically stable for the closed-loop system. ■

1.1.3 MPC for regulation with terminal cost function and without terminal constraint

In the last two subsections, we treated two valid and very common MPC schemes and showed both their advantages and drawbacks. Nonetheless, we analyzed such formulations only from a "control theorist's point of view", therefore focusing on the properties that a system must have to allow for the correct design of an MPC. Therefore, the main issues related to the MPC schemes presented in the previous subsections can be shortened in the following list:

- A QIH-MPC is particularly hard to design when the linearization of the system at the origin is not fully controllable.
- A TEC-MPC cannot be correctly implemented if the system is not controllable to the origin in a finite number of steps.

However, in the real world the challenges related to the design and implementation of an MPC are not limited to the properties of the system to be controlled. In fact, differently from the 1970s, MPC is nowadays increasingly used for the control of processes with relatively fast dynamics, which require to compute the control actions to be applied with high frequency. Consequently, even with the advanced technological resources of today, it is crucial to investigate MPC formulations that require solving simple optimization problems.

Among the actions that can be taken to reduce the time that is needed to solve the optimal control problem inherent in MPC, we surely find (i) the employment of short prediction horizons, and (ii) a thoughtful selection of the solver to be used to solve the optimization problem. Yet, in some contexts it is possible to work on another factor that greatly influences the complexity of an optimization problem, i.e. the constraints. Indeed, as explained at the beginning of Section 1.1, two of the main ingredients of a constrained optimal control problem are the hard state constraint set $\mathcal{X}$ and the hard input constraint set $\mathcal{U}$. However, while the input constraints are practically always present, as they represent actuator saturation levels or resource limits, it is not unlikely to find optimal control problems without state constraints. As a consequence, in these cases the only constraints on the state predictions are the stability-related terminal constraints, be they implemented as equality or inequality constraints. In the following, we show under what conditions it is possible to drop such constraints, still keeping the recursive feasibility and the stabilizing capabilities of the MPC. For further details about this topic, we refer to [85].

First, the currently analyzed context is formulated in a mathematical way, so as to confirm the consistency of the here-explained MPC scheme with the optimal control problem formulated in (1.3). In particular,

$$\mathcal{X}_f = \mathcal{X},$$

while the choice of the terminal cost function $V_f(\cdot)$ is discussed later.

We are now ready to present the theoretical concepts about the MPC for regulation with terminal cost function and without terminal constraint. In the following, for reasons that will become clear later in the text, we make use of a function $F : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$, which is then used to build the terminal cost function $V_f(\cdot)$. The conditions that this function has to meet are listed in the following assumption.

Assumption 1.6: Locally stabilizing control law and CLF sublevel set invariance

There exists a locally stabilizing control law $\kappa_f(\cdot)$, a function $F(\cdot)$ and a set Ω such that:

- *$F(x)$ is a CLF that verifies $\alpha_3(\|x\|) \leq F(x) \leq \alpha_4(\|x\|)$, where $\alpha_3(\cdot), \alpha_4(\cdot) \in \mathcal{K}$;*
- *the set Ω is defined as $\Omega := \{x \in \mathbb{R}^n : \; F(x) \leq \rho\}$, with $\rho > 0$;*
- *the control law $\kappa_f(\cdot)$ is feasible for any $x \in \Omega$ and is such that $F\left(f\left(x, \kappa_f(x)\right)\right) - F(x) \leq -\ell\left(x, \kappa_f(x)\right)$.*

Assumption 1.6 and 1.3 are similar, but it is important to note the following differences:

- Assumption 1.3 directly poses conditions on the terminal ingredients that have to be employed in a QIH-MPC. On the contrary, Assumption 1.6 poses conditions on a set Ω that is not used in a terminal inequality constraint, and requires $F(\cdot)$ to meet some conditions even though, as will be seen later, in the context of the currently treated MPC it is not always true that $V_f(x) = F(x)$.
- While Assumption 1.3 does not impose the terminal region $\mathcal{X}_f$ to have a particular shape, Assumption 1.6 imposes Ω to be a sublevel set of $F(\cdot)$.

We now make a further assumption on the stage cost function $\ell(\cdot, \cdot)$.

Assumption 1.7: Stage cost function infimum outside Ω

There exist a function $\alpha_\ell(\cdot) \in \mathcal{K}$ and a constant $d > 0$ such that:

1. *$\ell(x, u) \geq \alpha_\ell(\|x\|) \; \forall u \in \mathcal{U}$;*
2. *$\ell(x, u) > d$, $\forall x \notin \Omega$ and $\forall u \in \mathcal{U}$.*

Note that Assumption 1.7 is absolutely non-restrictive. De facto, it requires $\ell(\cdot, \cdot)$ to be lower-bounded exactly like in Assumption 1.4, but without posing conditions on the terminal cost function. Moreover, given the lower bound $\alpha_\ell(\cdot)$ and given that the origin is in the interior of Ω by Assumption 1.6, a constant $d > 0$ verifying Assumption 1.7 surely exists.

The constant d is used in the next theorem not only to prove the stabilizing capability of the MPC with terminal cost function and without terminal constraint, but also to provide the analytical formulation of a positively invariant set for the closed-loop system.

Theorem 1.3: Stability of the MPC with terminal cost function and without terminal constraint

Under Assumptions 1.1, 1.2, 1.6 and 1.7, the MPC defined in (1.3)*, with $N_p \geq 1$, $\mathcal{X}_f = \mathcal{X}$ and $V_f(\cdot) = F(\cdot)$ is recursively feasible and the origin is asymptotically stable for the closed-loop system starting from any initial state in*

$$\hat{\mathcal{X}}_{N_p} := \left\{x \in \mathbb{R}^n : \; V_{N_p}^*(x) \leq \ell\left(x, \kappa_{N_p}(x)\right) + (N_p - 1) \cdot d + \rho\right\}.$$

Proof. The proof is provided in [85]. ∎

Additionally, [85] presents further results that show how the domain of attraction of the MPC with terminal cost function and without terminal constraint can be enlarged by employing a terminal cost function $V_f(\cdot) = \chi F(\cdot)$, with $\chi > 1$.

As anticipated earlier in this subsection, an MPC without terminal constraint can prove useful to reduce the computational burden of the optimization problem $P_{N_p}(\cdot)$. Nevertheless, going back to a control theorist's perspective, it still requires to compute (offline) a terminal region Ω and a CLF $F(\cdot)$ that maintain the main properties of the terminal ingredients introduced in Section 1.1.1. For this reason, while the implementation of an MPC with terminal cost function and without terminal constraint can provide the aforementioned computational advantages, the same limits of a QIH-MPC are kept. In particular, we refer to the fact that, if the linearization of the system at the origin is not fully controllable, the computation of Ω and $F(\cdot)$ can still get challenging, if not impossible.

For this reason, in the next subsection we investigate an MPC scheme that makes use of no terminal ingredients.

1.1.4 MPC for regulation without terminal ingredients

In the following, we briefly summarize the conditions under which an MPC without terminal ingredients can still be proved to be recursively feasible and to stabilize the closed-loop system, similarly as done in [73]. In particular, details about such MPC formulation can be found in [55, Chapter 6].

First, as done also for the previously introduced schemes, we provide a mathematical definition of the terminal ingredients that makes the here-presented MPC consistent with (1.3). In detail, we have that

$$\mathcal{X}_f = \mathcal{X},$$

$$V_f(x) = 0.$$

The intuition behind this scheme comes from practical experience. Indeed, every MPC designer is aware of the fact that, by choosing a sufficiently large prediction horizon N_p, an MPC without terminal ingredients is often able to stabilize the closed-loop system. Nonetheless, it is crucial to understand the conditions that a system must meet to allow for the implementation of such an MPC and to quantitatively evaluate when a prediction horizon is "sufficiently large".

The main assumption that has to be made is the following.

Assumption 1.8: Asymptotic controllability

There exist functions $\alpha_{N_p}(\cdot) \in \mathcal{K}_\infty$ such that, for each $N_p \geq 1$ and for every $x \in \mathcal{X}$, the following inequality holds:

$$V^*_{N_p}(x) \leq \alpha_{N_p}\big(\inf_{u \in \mathcal{U}} \ell(x,u)\big).$$

Assumption 1.8 ensures that, for any state $x \in \mathcal{X}$, an open-loop trajectory that asymptotically steers the system state to the origin exists.

We now present the theorem that proves the stabilizing capabilities of MPC formulations without terminal ingredients.

Theorem 1.4: Stability of the MPC for regulation without terminal ingredients

Under Assumptions 1.1, 1.2, 1.4 and 1.8, there exists a constant $N_0 > 0$ such that, for any prediction horizon $N_p > N_0$, the MPC defined in (1.3), with $\mathcal{X}_f = \mathcal{X}$ and $V_f(\cdot) = 0$, is recursively feasible and the origin is asymptotically stable for the closed-loop system starting from any initial state in $\mathcal{X}$.

Proof. The proof is provided in [55, Chapter 6]. ■

In the context of this book, we are not interested in delving into the details of how the constant N_0 is computed. We refer the reader to [55, Chapter 6] for such details.

By looking at Theorem 1.4, a difference that stands out compared to the theorems presented earlier is that the domain of attraction is ensured to correspond to the entire state constraint set $\mathcal{X}$. This comes from the fact that Assumption 1.8 is quite stricter than the assumptions presented earlier, as its conditions are required to hold for all the states $x \in \mathcal{X}$. Moreover, finding $\mathcal{K}_\infty$ functions $\alpha_{N_p}(\cdot)$ verifying Assumption 1.8 can be hard. This is why the result introduced in the current subsection is of great importance, but allows us to conclude that, if possible, it is preferable to opt for the previously introduced MPC schemes, which might render the optimization problem to be solved more complex, but are based on assumptions that are easier to verify.

1.2 MPC for tracking changing setpoints

In Section 1.1 we presented different state-of-the-art MPC schemes for regulation, analyzed under what conditions they can stabilize the closed-loop system, and discussed their advantages and drawbacks. Now, a different control problem is considered, i.e. the problem of tracking changing setpoints.

Before introducing the MPC schemes for tracking changing setpoints that inspired the works introduced in this book, the context in which an MPC of this kind can prove useful is first examined. The control of complex processes is usually carried out by means of a multi-layer control structure [138]. This hierarchical structure is usually composed of the following elements:

- On the lowest level, we find the process to be controlled, with all its actuators and sensors.
- Immediately above, we find the low-level controllers that have to deal with the regulation of the process, acting directly on its actuators. This task is usually accomplished by standard controllers, e.g. Proportional-Integral-Derivative (PID) controllers, or by using Programmable Logic Controllers (PLC).
- The upper level is usually filled by more advanced multivariable controllers, e.g. MPC, which provide the needed references to the controllers of the lower level in order to keep the system at the desired operating point. Nonetheless, given that today's technological resources allow for the solution of complex optimization problems in a short time, it may happen that this and the lower level are merged and a single MPC is used to directly command the actuators.
- On the next level we find the real-time optimizer (RTO), which is responsible for finding the best operating point that meets the desired economic and performance conditions and the management requirements, ensuring that the found operating point is consistent with the properties of the system model used inside the lower-level MPC. The target operating point is usually given in the form of an output reference y_t.
- On the top, we find the information management system (IMS), which groups data coming from managers and from the company information technology (IT) systems to provide the information that is needed by the RTO to select the desired operating point of the process.

A schematic view of this hierarchical structure can be seen in Figure 1. In this context, whenever some conditions change, e.g. the cost of energy, the availability of resources, the production goal, etc., the RTO is expected to provide a new target y_t to the underlying MPC. Therefore, a question that arises is whether the MPC schemes for regulation presented in Section 1.2 can still provide the

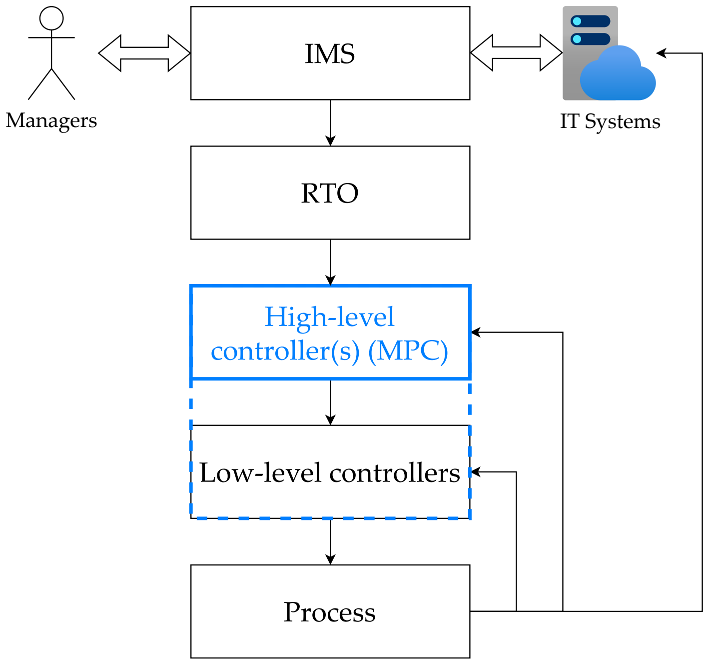


Figure 1: Hierarchical control structure where MPC is often employed.

previously mentioned stability guarantees and keep recursive feasibility. Unfortunately, the answer to this question is negative, leading to the reference signal not being tracked correctly by the closed-loop system [13, 110, 128]. In fact:

- A terminal region $\mathcal{X}_f$ (or a positively invariant set Ω), a terminal cost function $V_f(\cdot)$ and a local control law $\kappa_f(\cdot)$ that verify all the needed conditions for a certain setpoint, might not verify the same conditions for a different setpoint. One might be tempted to just translate the set $\mathcal{X}_f$ (or Ω) and center it in the new reference point, and reuse the same terminal cost function $V_f(\cdot)$ and the same local control law $\kappa_f(\cdot)$. However, $\kappa_f(\cdot)$ might not be feasible in the new region, $V_f(\cdot)$ might not behave as a local CLF and $\mathcal{X}_f$ (or Ω) might lose its invariance property. Moreover, given the new position of $\mathcal{X}_f$ (or Ω), the previously chosen prediction horizon is not guaranteed to be sufficiently large to allow for the terminal region to be reached within N_p steps. As a result, neither recursive feasibility nor stability of the closed-loop system can be ensured.

- In the case of regulation TEC-MPC schemes, a prediction horizon that ensures recursive feasibility and stability for a certain reference point, might prove insufficient for a new setpoint. Therefore, neither with such schemes the recursive feasibility and the stability of the closed-loop system can be guaranteed when the setpoint changes.

The last considerations have led the scientific community to study new solutions that could deal with changing setpoints without losing the stabilizing properties of a common MPC for regulation. A first solution to this issue can be found in dual-mode controllers, e.g. [26], which operate as regulators in the neighborhood of the desired equilibrium, recovering feasibility whenever the setpoint is changed. Another possible solution is given by MPC schemes that guarantee local stability and asymptotic tracking of constant references, like in [94]. Another family of controllers that generated interest in this context is provided by *reference governors*, e.g. [8, 13, 48], two-layer controllers that work as follows: (i) the upper-level filter takes the real reference as input and, based on the knowledge of the lower-level controller and of the system to be controlled, generates a modified (artificial) reference that maintains the feasibility and stabilizing capabilities of the lower-level controller, then (ii) the lower-level controller, designed as an unconstrained control law, is responsible for the stabilization of the artificial reference.

Nonetheless, today's state-of-the-art solutions for dealing with nonlinear systems and changing setpoints are often inspired by the MPC schemes introduced in [44, 84, 86] for the control of linear systems. An interesting survey about nonlinear MPC schemes for tracking changing setpoints is given in [77]. In particular, we would like to mention the methods proposed in [16, 28, 46, 90, 132]. The intuition behind these schemes is simple; given that the issues that a changing setpoint can provide are mainly due to inconsistencies between the setpoint and the system constraints, the terminal ingredients losing their properties and the prediction horizon being insufficiently large, one could act as follows:

- A set of feasible setpoints is pre-selected offline and terminal ingredients verifying the conditions explained in Section 1.2 for the entire set of feasible setpoints are designed.

- An *artificial reference* that belongs to the set of feasible setpoints is added to the optimal control problem to be solved as a decision variable, consequently letting the MPC select a fictitious setpoint that makes the optimization problem feasible.

- The cost function of the MPC is designed to penalize both the tracking error with respect to the artificial reference and the deviation between such artificial reference and the real setpoint. In particular, the latter requires the addition of an *offset cost function*.

Even from this qualitative description, it is possible to understand that an MPC designed in this way always keeps its feasibility property. Indeed, the controller is always given the freedom to choose an artificial reference that is close to the current system state. Another nice property of such schemes is that the feasibility of the MPC is maintained also when the real setpoint is infeasible, as the controller can still choose an artificial reference that is consistent with the system constraints. What this brief description cannot clarify is whether these MPC formulations guarantee the closed-loop stability of the system. In particular, we would like the closed-loop system state to stabilize at the feasible equilibrium that is the closest (with respect to some norm) to the real reference y_t. The aim of the following subsections is to formally provide the conditions under which this property is guaranteed. In detail, we focus on the tracking MPC schemes for nonlinear systems presented in [90], which inspired part of the contributions of this dissertation. Before going into the theoretical details of such methods, the control problem to be solved is formulated in a formal way below.

We consider a nonlinear time-invariant discrete-time system described by the equations

$$\begin{aligned} x(k+1) &= f\left(x(k), u(k)\right), \\ y(k) &= h\left(x(k), u(k)\right), \end{aligned} \tag{1.13}$$

where $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ is the system input, and $y \in \mathbb{R}^p$ is the system output. The system state and input are subject to hard constraints, namely

$$x(k) \in \mathcal{X} \subset \mathbb{R}^n, u(k) \in \mathcal{U} \subset \mathbb{R}^m, \forall k \geq 0. \tag{1.14}$$

The equilibria of the system are such that

$$x_s = f\left(x_s, u_s\right), \tag{1.15a}$$

$$y_s = h\left(x_s, u_s\right). \tag{1.15b}$$

Before proceeding, a more concise notation for the constraints defined in (1.14) is introduced. Defining the set $\mathcal{Z} := \mathcal{X} \times \mathcal{U} \subset \mathbb{R}^n \times \mathbb{R}^m$, the constraints (1.14) rewrite as

$$(x(k), u(k)) \in \mathcal{Z}, \ \forall k \geq 0.$$

Moreover, some assumptions that are shared by all the MPC for tracking schemes that are explained next are provided below.

Assumption 1.9: Continuity of the state and output functions

The state and output functions $f(\cdot,\cdot)$ and $h(\cdot,\cdot)$ are continuous at any equilibrium point.

Assumption 1.10: Properties of the constraint sets

The sets $\mathcal{X}$ and $\mathcal{U}$ are closed and their interior is non-empty.

In particular, Assumption 1.10 ensures that also the set $\mathcal{Z}$ is closed and has a non-empty interior. While Assumptions 1.9 and 1.10 recall Assumptions 1.1 and 1.2, which were provided at the beginning of Section 1.1 and were employed to prove the stabilizing capabilities of several regulation MPC schemes, the tracking context requires us to make some further assumptions. In particular, given that tracking problems usually require to find and stabilize the equilibrium (x_t, u_t) related to a certain output reference y_t, we must avoid any ambiguity and limit ourselves to the control of systems for which it is impossible to find different equilibria providing the same output. Before formalizing this assumption, let us define some sets that will be used later in the text.

As anticipated earlier, tracking MPC schemes require to build a priori a set of feasible setpoints. For reasons that will be cleared later on, the equilibrium points corresponding to the feasible setpoints are desired to be points with inactive constraints, namely they should belong to the set

$$\hat{\mathcal{Z}} := \{z \in \mathcal{Z} : \ z + \delta_z \in \mathcal{Z}, \ \forall \delta_z \text{ s.t. } \|\delta_z\| \leq \epsilon\}, \tag{1.16}$$

where $\epsilon > 0$ can be arbitrarily small, therefore allowing the points in $\hat{\mathcal{Z}}$ to be arbitrarily close to the boundaries of $\mathcal{Z}$. Then, the set of equilibrium states and inputs with inactive constraints is given by

$$\mathcal{Z}_s := \left\{(x, u) \in \hat{\mathcal{Z}} : \ x = f(x, u)\right\}, \tag{1.17}$$

and the corresponding outputs belong to the set

$$\mathcal{Y}_s := \{y = h(x, u) : \ (x, u) \in \mathcal{Z}_s\}. \tag{1.18}$$

Now that the aforementioned sets have been defined, we can formalize the anticipated assumption on the output function $h(\cdot, \cdot)$.

Assumption 1.11: Properties of the output function

The output function $h(\cdot, \cdot)$ is such that, for any given y_s, there exists a unique equilibrium point (x_s, u_s) such that $x_s = f(x_s, u_s)$ and $y_s = h(x_s, u_s)$. Furthermore, there exist a locally Lipschitz function $g_x : \mathcal{Y}_s \to \mathbb{R}^n$ and a continuous function $g_u : \mathcal{Y}_s \to \mathbb{R}^m$ such that

$$x_s = g_x(y_s), \quad u_s = g_u(y_s). \tag{1.19}$$

Remark 1.4

A wide family of systems that verify Assumption 1.11 is that of systems described by functions $f(\cdot, \cdot)$ and $h(\cdot, \cdot)$ that are continuously differentiable and whose Jacobian matrix

$$\begin{bmatrix} (A(x_s, u_s) - I_n) & B(x_s, u_s) \\ C(x_s, u_s) & D(x_s, u_s) \end{bmatrix},$$

where

$$A(x_s,u_s) = \frac{\partial f(x,u)}{\partial x}(x_s,u_s), \quad B(x_s,u_s) = \frac{\partial f(x,u)}{\partial u}(x_s,u_s),$$
$$C(x_s,u_s) = \frac{\partial h(x,u)}{\partial x}(x_s,u_s), \quad D(x_s,u_s) = \frac{\partial h(x,u)}{\partial u}(x_s,u_s),$$

is non-singular for all $(x_s,u_s) \in \mathcal{Z}_s$.

We are now ready to provide a generic formulation of the optimal control problem that has to be solved at each instant $k \geq 0$, here denoted as $P_{N_c,N_p}(x(k), y_t)$ and considered to have control and prediction horizons such that $N_c < N_p$:

$$\begin{aligned}\min_{(\mathbf{u}(k),y_s(k))} V_{N_c,N_p}(x(k), y_t, \mathbf{u}(k), y_s(k)) \coloneqq & \sum_{j=0}^{N_c-1} \ell\left(\hat{x}(j|k) - x_s(k), u(j|k) - u_s(k)\right) \\ & + \sum_{j=N_c}^{N_p-1} \ell\left(\hat{x}(j|k) - x_s(k), \kappa_f\left(\hat{x}(j|k), y_s(k)\right) - u_s(k)\right) \\ & + V_f\left(\hat{x}(N_p|k) - x_s(k), y_s(k)\right) + V_O\left(y_s(k) - y_t\right)\end{aligned} \tag{1.20a}$$

subject to :

$$\hat{x}(0|k) = x(k), \tag{1.20b}$$

$$\hat{x}(j|k) = f\left(\hat{x}(j-1|k), u(j-1|k)\right), \forall j \in [1, N_c], \tag{1.20c}$$

$$\hat{x}(j|k) = f\left(\hat{x}(j-1|k), \kappa_f\left(\hat{x}(j-1|k), y_s(k)\right)\right), \forall j \in [N_c+1, N_p], \tag{1.20d}$$

$$(\hat{x}(j|k), u(j|k)) \in \mathcal{Z}, \forall j \in [0, N_c-1], \tag{1.20e}$$

$$\left(\hat{x}(j|k), \kappa_f\left(\hat{x}(j|k), y_s(k)\right)\right) \in \mathcal{Z}, \forall j \in [N_c, N_p-1], \tag{1.20f}$$

$$x_s(k) = g_x(y_s(k)), \; u_s(k) = g_u(y_s(k)), \tag{1.20g}$$

$$\left(\hat{x}(N_p|k), y_s(k)\right) \in \mathcal{T}. \tag{1.20h}$$

In detail:

- $\mathbf{u}(k) \coloneqq \{u(0|k), \ldots, u(N_c-1|k)\}$ is the control sequence to be optimized, together with the artificial reference $y_s(k)$;

- $\kappa_f(\cdot,\cdot)$ is a local control law whose properties will be clarified later;

- $N_p \in \mathbb{N} \setminus \{0\}$ is the prediction horizon and $N_c \in \{1, \ldots, N_p\}$ is the control horizon;

- (1.20a) is the cost function that represents the control goal, and its main ingredients are the stage cost function $\ell(\cdot, \cdot)$, the (possible) terminal cost function $V_f(\cdot)$ and the offset cost function $V_O(\cdot)$;
- (1.20b) imposes the initial condition $\hat{x}(0|k)$ to be equal to the current state $x(k)$;
- (1.20c) and (1.20d) impose the predictions $\hat{x}(j|k)$ to be consistent with the system dynamics, defined by the function $f(\cdot, \cdot)$;
- (1.20e) and (1.20f) impose the state predictions $\hat{x}(j|k)$ and the chosen control actions $u(j|k)$ to verify the state and input constraints;
- (1.20g) explicates the relationship between the value taken by the artificial reference $y_s(k)$ and the corresponding equilibrium point $(x_s(k), u_s(k))$;
- (1.20h) is the terminal constraint for tracking to be imposed on the terminal state $\hat{x}(N_p|k)$ and the artificial reference $y_s(k)$ for stability reasons.

The optimal control sequence resulting from the solution of $P_{N_c,N_p}(x(k), y_t)$ is hereafter denoted as $(\mathbf{u}^0(x(k), y_t), y_s^0(x(k), y_t))$, or $(\mathbf{u}^0(k), y_s^0(k))$ for brevity, where

$$\mathbf{u}^0(k) = \left\{u^0(0|k), \ldots, u^0(N_c - 1|k)\right\},$$

while the optimal value of the cost function given by $(\mathbf{u}^0(k), y_s^0(k))$ is $V_{N_p}^0(x(k), y_t) \coloneqq V_{N_p}(x(k), y_t, \mathbf{u}^0(k), y_s^0(k))$. The control law implied by this MPC scheme is defined as $\kappa_{N_c,N_p}(x(k), y_t) \coloneqq u^0(0|k)$.

Problem (1.20) shows some differences with respect to (1.3). In detail, $V_{N_c,N_p}^0(x, y_t)$ is also a function of y_t. This is obvious, as the optimal solution to $P_{N_c,N_p}(x, y_t)$ has to depend on the current reference y_t. Nevertheless, y_t only appears in the cost function (1.20a), while the constraints (1.20b)-(1.20h) do not depend on y_t. For this reason, there surely exists a region of attraction $\mathcal{X}_{N_c,N_p} \subseteq \mathbb{R}^n$ where $P_{N_c,N_p}(x, y_t)$ is feasible for all $y_t \in \mathbb{R}^p$. The control law $\kappa_f(\cdot, y_s)$ plays a similar role to the local control law $\kappa_f(\cdot)$ considered for the computation of the terminal ingredients of a standard MPC for regulation, but is considered to be a function of both the state x and the artificial reference y_s. One aspect of the notation that the reader may have noticed is that we are now using 0 as a superscript when referring to optimal values obtained by solving $P_{N_c,N_p}(x, y_t)$, while in Section 1.1 we used the character $*$. To understand the motivation of this change it is necessary to anticipate some concepts. Indeed, in tracking MPC schemes with artificial references there is a difference between the best admissible steady output y_s^* and the artificial steady output y_s^0 that is selected at each instant k by the

MPC itself. Moreover, the closed-loop stability proofs exploit Lyapunov functions whose argument is $x(k) - x_s^0(k)$, namely the deviation of the current state from the artificial steady state chosen by the MPC, and then show how such Lyapunov functions can also prove the convergence to the best admissible steady state x_s^*.

Next, we provide the conditions under which MPC schemes for tracking with different terminal ingredients ensure recursive feasibility and stability of the best admissible equilibrium. In detail, (i) we first discuss the MPC for tracking with terminal inequality constraint and terminal cost function, (ii) then we introduce the MPC for tracking with terminal equality constraint, and (iii) finally we present the MPC for tracking without terminal constraint and with terminal cost function.

1.2.1 MPC for tracking changing setpoints with terminal inequality constraint and terminal cost function

Similarly to what was done in Section 1.1.1, we here analyze the conditions that an MPC for tracking with terminal inequality constraint and terminal cost function must meet to ensure the stability of the closed-loop system. However, given that the terminal constraint set $\mathcal{T}$ used in (1.20h) defines a constraint on both the terminal state and the artificial reference y_s, such set is first discussed below. In [90], $\mathcal{T}$ is referred to as *invariant set for tracking*, the definition of which is given next.

Definition 1.1: Invariant set for tracking

Given a constraint set $\mathcal{Z}$, a set of feasible setpoints $\mathcal{Y}_t \subseteq \mathcal{Y}_s$ and a local control law $\kappa_f(x, y_s)$, a set $\mathcal{T} \subset \mathbb{R}^n \times \mathbb{R}^p$ is an admissible invariant set for tracking for system (1.13) *if, for all $(x, y_s) \in \mathcal{T}$, the local control law $\kappa_f(x, y_s)$ is feasible and $\left(f\left(x, \kappa_f(x, y_s)\right), y_s\right) \in \mathcal{T}$.*

For a better understanding of what an invariant set for tracking is, considering an augmented system whose state x_a is given by (x, y_s), the dynamics of this system can be assumed to write as

$$\begin{cases} x(k+1) = f\left(x(k), \kappa_f(x(k), y_s(k))\right) \\ y_s(k+1) = y_s(k) \end{cases}$$

Then, $\mathcal{T}$ is nothing but a positively invariant set for the augmented system, i.e. a set of initial states x and references y_s such that, considering the reference y_s to be constant, provide an admissible evolution of the closed-loop system under the local control law $\kappa_f(x, y_s)$.

We can now list the assumptions that the terminal ingredients and the cost function must verify to guarantee recursive feasibility and closed-loop stability.

Assumption 1.12: Terminal ingredient properties

The control law $\kappa_f(\cdot,\cdot)$, the terminal cost function $V_f(\cdot,\cdot)$ and the set $\mathcal{T}$ must fulfill the following conditions:

1. *The set $\mathcal{T}$ is an invariant set for tracking for the system $x(k+1) = f\left(x(k), \kappa_f\left(x(k), y_s\right)\right)$.*
2. *The control law $\kappa_f\left(x(k), y_s\right)$ is continuous at (x_s, y_s) for all $y_s \in \mathcal{Y}_t$ and such that the equilibrium (x_s, u_s), with $x_s = g_x(y_s)$ and $u_s = g_u(y_s)$, is asymptotically stable for the system $x(k+1) = f\left(x(k), \kappa_f\left(x(k), y_s\right)\right)$, for all $(x, y_s) \in \mathcal{T}$.*
3. *The terminal cost function $V_f(x - x_s, y_s)$ is a CLF for the system $x(k+1) = f\left(x(k), \kappa_f\left(x(k), y_s\right)\right)$ in $\mathcal{T}$, i.e.*

$$V_f\left(f\left(x, \kappa_f\left(x, y_s\right)\right) - x_s, y_s\right) - V_f\left(x - x_s, y_s\right) \leq -\ell\left(x - x_s, \kappa_f(x, y_s) - u_s\right).$$

Assumption 1.13: Stage cost function and terminal cost function bounds

There exist a $\mathcal{K}_\infty$ function $\alpha_\ell(\cdot)$ and real constants $b > 0$ and $\sigma > 1$ that verify the following conditions:

$$\ell(x, u) \geq \alpha_\ell(\|x\|), \ \forall (x, u) \in \mathbb{R}^n \times \mathbb{R}^m,$$

$$V_f(x - x_s, y_s) \leq b\|x - x_s\|^\sigma, \ \forall (x, y_s) \in \mathcal{T}.$$

Assumption 1.12 represents a tracking version of Assumption 1.3, which lists the properties that the terminal ingredients of an MPC for regulation must have. The main difference resides in the fact that such conditions are required to be met for any output reference in a pre-defined set $\mathcal{Y}_t$, which translates in the need of designing a control law $\kappa_f(\cdot,\cdot)$ that stabilizes any equilibrium point defined by the feasible setpoints in $\mathcal{Y}_t$. Assumption 1.13 is instead similar to Assumption 1.4. As a matter of fact, the only notable difference is that Assumption 1.13 imposes the upper bound of $V_f(\cdot,\cdot)$ to belong to the family of power functions with power exponent greater than 1. This is stricter than requiring $V_f(\cdot,\cdot)$ to be upper-bounded by any $\mathcal{K}_\infty$ function, yet it is compatible with the design choices that are usually made in the MPC design process, as the terminal cost function is often designed as a quadratic function.

As anticipated earlier, tracking MPC schemes require to add an offset cost function to the usual MPC cost function, and also to pre-select the set $\mathcal{Y}_t$ of feasible setpoints. For these reasons, the next assumption provides the conditions that such offset cost function and feasible setpoint set must fulfill.

Assumption 1.14: Properties of the offset cost function and the feasible setpoint set

The feasible setpoint set $\mathcal{Y}_t$ and the offset cost function $V_O(\cdot)$ are such that:

1. *$\mathcal{Y}_t$ is a convex subset of $\mathcal{Y}_s$.*
2. *$V_O : \mathbb{R}^p \to \mathbb{R}_{\geq 0}$ is a sub-differentiable convex positive definite function with a unique minimizer $y_s^* = \arg\min_{y_s \in \mathcal{Y}_t} V_O(y_s - y_t)$.*
3. *There exists a function $\alpha_O(\cdot) \in \mathcal{K}_\infty$ such that*

$$V_O(y_s - y_t) - V_O(y_s^* - y_t) \geq \alpha_O(\|y_s - y_s^*\|) \; \forall y_s \in \mathcal{Y}_t.$$

Assumption 1.14 is not particularly restrictive. Indeed, if the set $\mathcal{Y}_s$ is convex, the feasible setpoint set can be directly set as $\mathcal{Y}_t = \mathcal{Y}_s$. Moreover, the conditions on $V_O(\cdot)$ can be easily met by employing a norm function as offset cost function.

The next theorem ensures the recursive feasibility and stabilizing capabilities of the MPC for tracking with terminal inequality constraint and terminal cost function.

Theorem 1.5: Stability of the MPC for tracking with terminal inequality constraint and terminal cost function

Under Assumptions 1.9-1.14 and considering a constant setpoint $y_t \in \mathbb{R}^p$, the MPC for tracking with terminal inequality constraint and terminal cost function is recursively feasible, the closed-loop system is stable, fulfills the constraints $\forall k \geq 0$, and converges to an equilibrium point such that:

1. *if $y_t \in \mathcal{Y}_t$, then $\lim_{k\to\infty} \|y(k) - y_t\| = 0$;*
2. *if $y_t \notin \mathcal{Y}_t$, then $\lim_{k\to\infty} \|y(k) - y_s^*\| = 0$, where $y_s^* = \arg\min_{y_s \in \mathcal{Y}_t} V_O(y_s - y_t)$.*

Proof. The proof is provided in [90, Proof of Theorem 1] and here briefly sketched.

Starting from a state $x(k)$ such that $P_{N_c,N_p}(x(k), y_t)$ is feasible and the optimal solution is given by $(\mathbf{u}^0(k), y_s^0(k))$, a feasible solution to $P_{N_c,N_p}(x(k+1), y_t)$ is given by the couple $(\mathbf{u}^+(k+1), y_s^+(k+1))$, where $y_s^+(k+1) = y_s^0(k)$ and $\mathbf{u}^+(k+1) \coloneqq \{u^0(1|k), \ldots, u^0(N_c-1|k), \kappa_f(\hat{x}^0(N_c|k), y_s^0(k))\}$, therefore proving recursive stability.

The asymptotic stability of the best admissible equilibrium (x_s^*, u_s^*) is proved by showing that the function $W(x, y_t) \coloneqq V^0_{N_c,N_p}(x, y_t) - V_O(y_s^* - y_t)$ is a Lyapunov function for the closed-loop system in a neighborhood of the equilibrium point. Therefore, a lower and an upper bound are found for $W(x, y_t)$

using standard arguments, while the descent property of $W(x, y_t)$ is proved in a different way than usual. Indeed, it is found that $W(x(k+1), y_t) - W(x(k), y_t) \leq -\alpha_\ell(\|x(k) - x_s^0\|)$, which only proves that x_s^0 is an asymptotically stable equilibrium, as x_s^* does not appear in the inequality. Nonetheless, such descent property leads to

$$\lim_{k\to\infty} \|x(k) - x_s^0(x(k), y_t)\| = 0,$$

which is proved to imply

$$\lim_{k\to\infty} \|x(k) - x_s^*\| = 0,$$

meaning that the closed-loop system converges to best admissible steady-state x_s^*. ■

The result shown in Theorem 1.5 is important and is based on assumptions that are not much different from those that ensure the stabilizing capability of a regulation QIH-MPC. However, the similarities between the assumptions of such schemes causes the here-presented MPC for tracking to inherit also the disadvantages of the previously treated QIH-MPC. Consequently, if the system is controllable but makes the computation of the terminal ingredients difficult, one could consider to opt for an MPC for tracking with terminal equality constraint.

1.2.2 MPC for tracking changing setpoints with terminal equality constraint

Designing an MPC for tracking changing setpoints with terminal equality constraint is equivalent to employing the MPC defined in (1.20) and setting the terminal ingredients as

$$\mathcal{T} = \{g_x(y_s)\} \times \mathcal{Y}_t,$$

$$V_f(x - x_s, y_s) = 0.$$

Such terminal ingredients read as follows: the artificial reference y_s is still allowed to take values in $\mathcal{Y}_t$, while the terminal state is imposed to be exactly equal to $x_s = g_x(y_s)$. Moreover, given that defining a locally stabilizing control law $\kappa_f(\cdot,\cdot)$ is quite unusual in MPC schemes with terminal equality constraint, as it should guarantee to reach the desired equilibrium in a finite number of steps, it is convenient to set $N_c = N_p$.

Even though the tracking context is different from the regulation one, the correct usage of a terminal equality constraint still requires a controllability assumption, which is formalized next.

Assumption 1.15: Controllability of the system

The function $f(x,u)$ *is differentiable at any equilibrium point* $(x_s,u_s) \in \mathcal{Z}_s$ *and the couple* $(A(x_s,u_s), B(x_s,u_s))$ *obtained by linearizing the model is controllable* $\forall (x_s,u_s) \in \mathcal{Z}_s$. *Furthermore, there exist real constants* $\epsilon, b > 0$ *and* $\sigma > 1$ *such that*

$$\sum_{j=0}^{N_p-1} \ell\left(\hat{x}(j|k) - x_s, u(j|k) - u_s\right) \leq b\|x - x_s\|^\sigma$$

is verified for any feasible solution $(\mathbf{u}, y_s)$ *of* $P_{N_p}(x, y_t)$ *such that* $\|x - x_s\| \leq \epsilon$ *and* $\|u(j|k) - u_s\| \leq \epsilon$.

The conditions of Assumption 1.15 are met, for example, if the stage cost function is quadratic and the linearized model at each equilibrium point is controllable.

We are now ready to introduce the theorem that ensures the recursive feasibility and the stabilizing capabilities of the MPC for tracking with terminal equality constraint.

Theorem 1.6: Stability of the MPC for tracking with terminal equality constraint

Under Assumptions 1.9, 1.10, 1.11, 1.14 and 1.15, and considering a constant setpoint $y_t \in \mathbb{R}^p$, *the MPC for tracking with terminal equality constraint with* $N_c = N_p \geq n$ *is recursively feasible, the closed-loop system is stable, fulfills the constraints* $\forall k \geq 0$, *and converges to an equilibrium point such that:*

1. *if* $y_t \in \mathcal{Y}_t$, *then* $\lim_{k\to\infty} \|y(k) - y_t\| = 0$;
2. *if* $y_t \notin \mathcal{Y}_t$, *then* $\lim_{k\to\infty} \|y(k) - y_s^*\| = 0$, *where* $y_s^* = \arg\min_{y_s \in \mathcal{Y}_t} V_O(y_s - y_t)$.

Proof. The proof is provided in [90, Proof of Theorem 2]. However, consistently with standard MPC theory, the proof mainly aims at exploiting Assumption 1.15 to prove that an upper bound for the function $W(x, y_t)$ defined in the Proof of Theorem 1.5 exists also in the terminal equality constraint setup. Recursive feasibility, as well as the descent property of $W(x, y_t)$ and the existence of a lower bound, are proved by following the same arguments of the Proof of Theorem 1.5. ∎

1.2.3 MPC for tracking changing setpoints with terminal cost function and without terminal constraint

In cases in which the computation of a terminal cost function $V_f(\cdot,\cdot)$ that behaves as a CLF and a control law $\kappa_f(\cdot,\cdot)$ stabilizing all the desired equilibria is possible, but it is sensible not to impose a terminal constraint, an alternative to the MPC for tracking with terminal equality constraint is given by MPC schemes for tracking with terminal cost function and without terminal constraint. In this context, the same optimal control problem in (1.20) can be solved by defining $\mathcal{T}$ as

$$\mathcal{T} = \mathcal{X} \times \mathcal{Y}_t.$$

Therefore, the terminal state is just required to verify the state constraints and the artificial reference y_s is still allowed to take all the possible values in $\mathcal{Y}_t$.

Consistently with the theory presented in Section 1.1.3, which inspired the MPC explained in the current subsection, we make the next assumptions.

Assumption 1.16: Locally stabilizing control law and CLF sublevel set invariance

There exist a control law $\kappa_f(\cdot,\cdot)$, a function $F(\cdot,\cdot)$ and a set Ω that fulfill the following conditions:

1. *The set $\Omega := \{(x, y_s) : V_f(x - g_x(y_s), y_s) \leq \rho,\ y_s \in \mathcal{Y}_t\}$, with $\rho > 0$, is an invariant set for tracking for the system $x(k+1) = f\left(x(k), \kappa_f\left(x(k), y_s\right)\right)$.*

2. *The control law $\kappa_f\left(x(k), y_s\right)$ is continuous at (x_s, y_s) for all $y_s \in \mathcal{Y}_t$ and such that the equilibrium (x_s, u_s), with $x_s = g_x(y_s)$ and $u_s = g_u(y_s)$, is asymptotically stable for the system $x(k+1) = f\left(x(k), \kappa_f\left(x(k), y_s\right)\right)$, for all $(x, y_s) \in \Omega$.*

3. *The function $F(x - x_s, y_s)$ is a CLF for the system $x(k+1) = f\left(x(k), \kappa_f\left(x(k), y_s\right)\right)$ in Ω, i.e.*

$$F\left(f\left(x, \kappa_f\left(x, y_s\right)\right) - x_s, y_s\right) - F\left(x - x_s, y_s\right) \leq -\ell\left(x - x_s, \kappa_f(x, y_s) - u_s\right).$$

Assumption 1.17: Stage cost function infimum outside Ω

There exists a constant $d > 0$ such that for any $y_s \in \mathcal{Y}_t$ we have that $\ell(x - g_x(y_s), u - g_u(y_s)) > d$, $\forall x \notin \Omega$ and $\forall u \in \mathcal{U}$.

In particular, Assumption 1.16 is equivalent to Assumption 1.12, but does not require the CLF $F(\cdot,\cdot)$ and the invariant set for tracking Ω to be used as terminal cost function and terminal set.

Given the knowledge of Ω, ρ and d, we can now present the theorem that ensures recursive feasibility and the stabilizing capabilities of the MPC for tracking with terminal cost function and without terminal constraint.

Theorem 1.7: Stability of the MPC for tracking with terminal cost function and without terminal constraint

Under Assumptions 1.9, 1.10, 1.11, 1.14, 1.16 and 1.17, considering a constant setpoint $y_t \in \mathbb{R}^p$ *and employing a terminal cost function* $V_f(\cdot,\cdot) = \chi F(\cdot,\cdot)$, *with* $\chi \geq 1$, *for any initial condition* $x \in \left\{x : V^0_{N_c,N_p}(x, y_t) - V_O(y^*_s - y_t) \leq N_p d + \chi\rho\right\}$, *the MPC for tracking with terminal cost function and without terminal constraint is recursively feasible, the closed-loop system is stable, fulfills the constraints* $\forall k \geq 0$, *and converges to an equilibrium point such that:*

1. *if* $y_t \in \mathcal{Y}_t$, *then* $\lim_{k\to\infty} \|y(k) - y_t\| = 0$;
2. *if* $y_t \notin \mathcal{Y}_t$, *then* $\lim_{k\to\infty} \|y(k) - y^*_s\| = 0$, *where* $y^*_s = \arg\min_{y_s \in \mathcal{Y}_t} V_O(y_s - y_t)$.

Proof. The proof is provided in [90, Proof of Theorem 2]. ■

1.3 Robust MPC

Sections 1.1 and 1.2 treat MPC schemes that accomplish two different control goals, i.e. the regulation to a feasible equilibrium of the system and the tracking of a changing (and potentially infeasible) setpoint. As we have seen, the difference in the aim of such MPC formulations results in a different structure of the optimal control problem to be solved. Nonetheless, the MPC schemes that we have analyzed share an important underlying hypothesis: there are no mismatches between the system equations and the model used to make predictions, and there are no external disturbances acting on the system. This hypothesis has a huge impact on the structure of the optimal control problem to be solved and on the feasibility and stability properties of the MPC. In particular:

- Imposing that the state predictions verify the constraints is sufficient to ensure that the closed-loop system state also verifies the constraints.

- Proving recursive feasibility is relatively simple. Indeed, given a certain time instant k, an initial condition $x(k)$, a feasible control sequence obtained from the solution to an MPC (be it a regulation or tracking MPC) $\mathbf{u}(k)$, and the predicted state trajectory $\hat{\mathbf{x}}(k) = \left\{\hat{x}(1|k), \ldots, \hat{x}(N_p|k)\right\}$, we have that $x(k+1) = \hat{x}(1|k)$. This equality is fundamental, as it allows us to easily find a candidate feasible control sequence $\mathbf{u}^+(k+1)$ for the solution of the optimal control problem at time $k+1$.

- It is possible to prove that the closed-loop system is asymptotically stable, i.e. that the optimal cost function of the MPC is a Lyapunov function. Specifically, it is not hard to prove the descent property of such function.

A question that arises is what happens when such hypothesis is false, considering that in reality it is practically impossible to have no model mismatches and disturbances. Before continuing this discussion, a linguistic simplification is made. Indeed, with a little abuse of language, model mismatches and external disturbances are referred to as "perturbations".
A mathematical formulation of the problem is provided below. Considering a time-invariant discrete-time system described by the equation

$$x(k+1) = f\left(x(k), u(k), w(k)\right), \tag{1.21}$$

where $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ is the system input, and $w \in \mathcal{W} \subseteq \mathbb{R}^r$ is the perturbation acting on the system. The system state and input are subject to hard constraints, namely

$$x(k) \in \mathcal{X} \subset \mathbb{R}^n, u(k) \in \mathcal{U} \subset \mathbb{R}^m, \forall k \geq 0. \tag{1.22}$$

Hence, can a standard MPC deal with the perturbation $w \in \mathcal{W}$, or is it necessary to act differently? It depends. If $\mathcal{W}$ is unbounded there is no possibility for an MPC (and for any other control law) to ensure constraint satisfaction for all the possible perturbation realizations. In such cases, it is common to ensure constraint satisfaction with a certain probability. This is the field of stochastic MPC, which we do not address. For details about generic stochastic MPC schemes, we refer to [126, Section 3.7], [71, Part 3] and [121], while an example of stochastic MPC for tracking (for linear systems) is given in [41]. Another common case is that of a bounded perturbation set $\mathcal{W}$. In this case, as pointed out in [126, Section 3.2] and [111], if the perturbations are sufficiently small, a standard MPC maintains its recursive feasibility in a subset of the nominal region of attraction. However, asymptotic stability cannot be ensured. This is obvious, as the presence of perturbations preclude any controller from steering the system state to an exact point. The best we can do is guaranteeing that the effect of bounded perturbations on the system state is bounded, a condition that allows us to prove that the closed-loop system state converges to a small neighborhood of the desired equilibrium. This is the fundamental concept behind input-to-state stability, briefly summarized in the Appendix A.5 and for which we refer to [87].
Nevertheless, analyzing the inherent robustness of standard MPC schemes often turns out to be less convenient than directly synthesizing an MPC that is robust by design. Indeed, the perturbation bounds are usually not chosen, but are given by the context. Consequently, we are often more interested in

designing predictive controllers that are robust with respect to perturbations that meet pre-determined bounds than in analyzing the inherent robustness of a standard MPC and checking that the found robustness limits are consistent with the pre-determined perturbation bounds. This is why the robust MPC research strand was born.

Before introducing the main robust MPC formulations, let us address some concepts that will prove useful for their understanding. It is well known that, when there are no perturbations on the system to be controlled, open-loop control and feedback control provide the same performances. Feedback controllers become much more sensible in the presence of perturbations, as these can be counteracted whenever the loop is closed. This idea can also be leveraged inside MPC. The MPC setups shown in Sections 1.1 and 1.2 belong to the wider family of *open-loop MPC*, as they optimize over a sequence of control actions. As we will see, most robust MPC formulations belong to the family of *feedback MPC*, namely they aim at finding a sequence of control laws that minimizes a performance index. Minimizing over a sequence of control laws is beneficial from a control perspective, as it allows the MPC to significantly reduce the effect of perturbations on the closed-loop performances. However, an optimal control problem involving a sequence of control laws as decision variable is an infinite-dimensional problem, which cannot be solved. Thus, it is necessary to give up optimality in favor of the solvability of the problem, enforcing the sequence of control laws to belong to a smaller family of parameterized laws, and allowing the MPC to optimize over their parameters [126, Section 3.1].

Remark 1.5

A common parameterized feedback control law that is used in feedback MPC schemes is given by linear controllers of the kind

$$u = K\,(x - x_s) + v,$$

where K is the feedback gain, x_s is the equilibrium to be stabilized and v is a decision variable. The gain K is usually designed to make the linearization of the system at x_s asymptotically stable.

We are now ready to introduce the two main approaches that are used to build a robust MPC, namely *min-max MPC* and *tube-based MPC*.

Min-max MPC schemes ensure robustness by computing the sequence of control laws (or control actions) that minimizes the worst-case performance index, i.e. the performance index is maximized with respect to the perturbation sequence and minimized with respect to the control sequence. Furthermore, the control sequence is chosen to verify the problem constraints under all the possible perturbation realizations. We refer to [120] for an overview of the stabilizing capabilities of min-max MPC schemes for nonlinear systems. Examples of min-max regulation setups for both linear and nonlinear systems can be found in [1, 14, 39, 40, 65, 70, 80, 91, 141].

While minimizing the worst-case performance index is desirable, it also results in a relevant increase in the computational burden of the MPC. For this reason, different formulations, i.e. tube-based, have been designed to ensure robustness while keeping the typical computational complexity of a standard MPC. Tube-based MPC schemes can also be implemented in an open-loop or feedback fashion, but get rid of the main drawback of min-max formulations by minimizing the performance index related to the nominal case, i.e. without perturbations, and ensure that the constraints are satisfied in closed loop by including in the optimal control problem a sequence of tightened constraints that take into account all the possible deviations from the nominal trajectory, which are given by the perturbations and the propagation of their effect over the entire prediction horizon. Among the available tube-based MPC schemes for regulation, we find formulations that are based on conservative tubes that are pre-computed offline, as [27, 104, 105, 145], and other predictive controllers that exploit a parameterized version of the tubes that allows them to be added as decision variables in the optimization problem that has to be solved online, as in [76, 92, 122, 123, 124]. Effective robust tube-based tracking MPC schemes were proposed for the control of linear systems subject to additive perturbations in [45, 88], while a solution for nonlinear affine systems was presented in [146] for tracking changing setpoints in case of slowly varying perturbations.

In the next subsection, we focus on a robust tube-based MPC for the regulation of nonlinear systems that was presented in [89], given that it forms the basis of much of the theoretical contributions of this book.

1.3.1 A robust tube-based MPC for the regulation of nonlinear systems

Consistently with the context of [89], we assume that we want to stabilize the origin, which requires the equilibrium $0 = f(0,0,0)$ to exist, and we consider the perturbation set to be a ball, i.e.

$$\mathcal{W} = \{w \in \mathbb{R}^r : \ \|w\| \leq \bar{w}\} .$$

Next, we discuss the concepts that are needed for the understanding of the here-explained MPC, formulate the optimal control problem to be solved, and provide the assumptions under which this formulation guarantees input-to-state stability of the origin for the closed-loop system.

Before proceeding, as done also in Section 1.2, the set $\mathcal{Z} := \mathcal{X} \times \mathcal{U}$ is defined, such that the constraints (1.22) rewrite as

$$(x(k), u(k)) \in \mathcal{Z}, \ \forall k \geq 0. \tag{1.23}$$

Following the same order used in Sections 1.1 and 1.2, we first make some assumptions about the continuity of the state function $f(\cdot,\cdot,\cdot)$ and the properties of the constraint sets.

Assumption 1.18: Uniform continuity of the state function

The state function $f(\cdot,\cdot,\cdot)$ is uniformly continuous with respect to all its arguments in the set $\mathcal{Z}\times\mathcal{W}$, i.e. there exist functions $\sigma_x(\cdot), \sigma_u(\cdot), \sigma_w(\cdot) \in \mathcal{K}$ such that

$$\|f(x,u,w) - f(\check{x},\check{u},\check{w})\| \leq \sigma_x(\|x-\check{x}\|) + \sigma_u(\|u-\check{u}\|) + \sigma_w(\|w-\check{w}\|),$$

for all (x,u,w) and $(\check{x},\check{u},\check{w})$ in $\mathcal{Z}\times\mathcal{W}$.

Assumption 1.19: Properties of the constraint sets

The sets $\mathcal{X}$ and $\mathcal{U}$ are closed and contain the origin in their interior.

Assumption 1.19 ensures that also the set $\mathcal{Z}$ is closed and includes the origin in its interior, while Assumption 1.18 is a continuity assumption that is stricter than those employed in the previously addressed MPC schemes, but it is necessary to enable the design of the employed tightened constraints. The MPC presented in [89] is designed in a feedback fashion. Consequently, it is considered to minimize an objective function with respect to a sequence of parameterized control laws. In particular, we consider to employ a family of control laws that is given by

$$u(k) = \pi\left(x(k), v(k)\right), \tag{1.24}$$

and that is considered to be uniformly continuous in its domain. Given that $v(k) \in \mathbb{R}^s$ plays the role of a controlled input and is the new decision variable, the constraints (1.23) can be reformulated as

$$(x(k), v(k)) \in \mathcal{Z}_\pi, \ \forall k \geq 0,$$

where $\mathcal{Z}_\pi$ is defined as

$$\mathcal{Z}_\pi := \{(x,v) \in \mathbb{R}^n \times \mathbb{R}^s : \ (x, \pi(x,v)) \in \mathcal{Z}\}.$$

Furthermore, we introduce a novel notation to denote the state trajectories of the closed-loop system under the control law $\pi(\cdot,\cdot)$. In detail, with $\phi_\pi(j; x, \mathbf{v}, \mathbf{w})$ we denote the solution of system (1.21) at the time instant j given the initial condition $x(0) = x$, the feedback policy $\pi(x,v)$, where v comes from the sequence $\mathbf{v}$, and the perturbation sequence $\mathbf{w}$.

The next two assumptions define set sequences that are employed in the computation of the tubes of the MPC presented in [89] and that are needed to ensure its recursive feasibility.

Assumption 1.20: Bounds on the effect of the initial condition on the state trajectory

There exists a sequence of sets $\{\mathcal{F}(j)\}_{j\geq 0}$ *such that:*

- $\mathcal{F}(0) := \{x \in \mathbb{R}^n : \|x\| \leq \sigma_w(\bar{w})\}$;
- $\forall j > 0$, *the sets* $\mathcal{F}(j)$ *ensure that, for all the feasible* $(x, \mathbf{v})$ *and for every* $\check{x}$ *such that* $\check{x} - x \in \mathcal{F}(0)$, *the condition*

$$\phi_\pi(j; \check{x}, \mathbf{v}, \mathbf{0}) \in \phi_\pi(j; x, \mathbf{v}, \mathbf{0}) \oplus \mathcal{F}(j)$$

is verified.

Assumption 1.21: Bounds on the effect of the perturbations on the state trajectory

There exists a sequence of sets $\{\mathcal{R}(j)\}_{j\geq 0}$ *such that, defining* $\mathcal{R}(0) := \{0\}$, *it verifies the following conditions:*

- *for all the feasible* $(x, \mathbf{v})$ *and* $\forall j > 0$, *the sets* $\mathcal{R}(j)$ *ensure that*

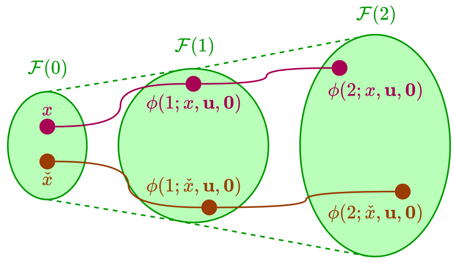
$$\phi_\pi(j; x, \mathbf{v}, \mathbf{w}) \in \phi_\pi(j; x, \mathbf{v}, \mathbf{0}) \oplus \mathcal{R}(j), \ \forall \mathbf{w} \in \mathcal{W}^j;$$

- $\mathcal{F}(j) \oplus \mathcal{R}(j) \subseteq \mathcal{R}(j+1), \ \forall j \geq 0.$

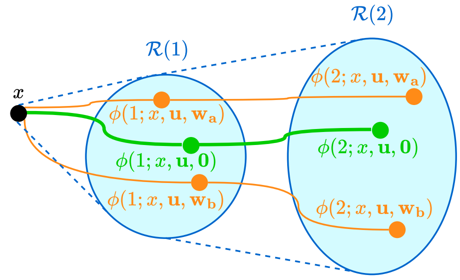
The sequence $\{\mathcal{F}(j)\}_{j\geq 0}$ defined in Assumption 1.20 bounds the gap between two state trajectories obtained by starting at close initial conditions and employing the same control sequence $\mathbf{v}$, while the sequence $\{\mathcal{R}(j)\}_{j\geq 0}$ defined in Assumption 1.21 bounds the gap between a perturbation-free state trajectory and all the possible perturbed trajectories. A schematic view of the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ is given in Figure 2. Moreover, as will become clear later in the text, the second condition of Assumption 1.21 is a monotonicity assumption that is essential to prove recursive feasibility. [89, Lemma 3] proves that Assumption 1.18 is sufficient to ensure that the conditions of Assumptions 1.20 and 1.21 are verified and provides a method for computing the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ based on the uniform continuity of the state function $f(\cdot, \cdot, \cdot)$.

As anticipated earlier, the sequence of tightened constraints is built based on the aforementioned sequences. In detail, provided a prediction horizon N_p, the sequence of tightened constraints $\{\mathcal{Z}_\pi(j)\}_{j\in[0,N_p]}$ is such that

$$\mathcal{Z}_\pi(j) := \mathcal{Z}_\pi \ominus (\mathcal{R}(j) \times \{0\}) .$$



(a) Graphic example that shows how the sequence $\{\mathcal{F}(j)\}_{j\geq 0}$ bounds the gap between nominal trajectories obtained starting from close initial conditions.



(b) Graphic example that shows how the sequence $\{\mathcal{R}(j)\}_{j\geq 0}$ bounds the gap between nominal trajectories and perturbed trajectories obtained starting from the same initial condition.

Figure 2: Graphical description of the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$.

We now have all the ingredients that are needed to present the optimal control problem $P_{N_p}(x(k))$ that has to be solved $\forall k \geq 0$:

$$\min_{\mathbf{v}(k)} V_{N_p}(x(k), \mathbf{v}(k)) \coloneqq \sum_{j=0}^{N_p-1} \ell\left(\hat{x}(j|k), v(j|k)\right) + V_f\left(\hat{x}(N_p|k)\right) \tag{1.25a}$$

subject to :

$$\hat{x}(0|k) = x(k), \tag{1.25b}$$

$$\hat{x}(j|k) = f\left(\hat{x}(j-1|k), \pi\left(\hat{x}(j-1|k), v(j-1|k)\right), 0\right), \forall j \in [1, N_p], \tag{1.25c}$$

$$\left(\hat{x}(j|k), v(j|k)\right) \in \mathcal{Z}_\pi(j), \forall j \in [0, N_p - 1], \tag{1.25d}$$

$$\hat{x}(N_p|k) \in \mathcal{X}_f. \tag{1.25e}$$

In detail, the main differences with respect to the standard regulation MPC optimal control problem (1.3) are the following:

- the cost function (1.25a) is minimized with respect to the sequence $\mathbf{v}(k)$, i.e. a sequence of parameters that define the control laws $\pi(\cdot, \cdot)$;
- the state and input constraints (1.25d) are tightened and change along the prediction horizon.

The optimal control sequence resulting from the solution of $P_{N_p}(x(k))$ is hereafter denoted as $\mathbf{v}^*(x(k))$, or $\mathbf{v}^*(k)$ for brevity, where $\mathbf{v}^*(k) = \left\{v^*(0|k), \ldots, v^*(N_p-1|k)\right\}$, while the optimal value of the cost function given by $\mathbf{v}^*(k)$ is $V^*_{N_p}(x(k)) \coloneqq V_{N_p}(x(k), \mathbf{v}^*(k))$. The control law implied by this MPC scheme is defined as $\kappa_{N_p}(x(k)) \coloneqq \pi(x(k), v^*(0|k))$.

We now turn our attention to the terminal ingredients and the cost function, which have to meet the conditions defined in the following assumptions.

Assumption 1.22: Terminal ingredient properties

There exist an input $v_f \in \mathbb{R}^s$, and two sets $\Omega, \mathcal{X}_f \subset \mathcal{X}$ such that:

1. Ω *and* $\mathcal{X}_f$ *are positively invariant sets for the nominal system* $\hat{x}(k+1) = f\left(\hat{x}(k), \pi(\hat{x}(k), v_f), 0\right)$;
2. $\Omega \times \{v_f\} \subseteq \mathcal{Z}_\pi(N_p - 1)$;
3. $\mathcal{X}_f \times \{v_f\} \subseteq \mathcal{Z}_\pi(N_p)$;
4. $\mathcal{X}_f \oplus \mathcal{F}(N_p - 1) \subseteq \Omega$;
5. $f\left(\hat{x}, \pi(\hat{x}, v_f), 0\right) \in \mathcal{X}_f$, *for all* $\hat{x} \in \Omega$.

Moreover, the terminal cost function $V_f(\cdot)$ *is a CLF in* Ω, *such that*

$$V_f\left(f(\hat{x}, \pi(\hat{x}, v_f), 0)\right) - V_f(\hat{x}) \leq -\ell(\hat{x}, v_f), \ \forall \hat{x} \in \Omega.$$

Assumption 1.23: Stage cost function and terminal cost function bounds

The stage cost function $\ell(\cdot,\cdot)$ *and the terminal cost function* $V_f(\cdot)$ *verify the next conditions:*

1. *There exists a function* $\alpha_\ell(\cdot) \in \mathcal{K}$ *such that* $\ell(x, v) \geq \alpha_\ell(\|x\|), \ \forall (x, v) \in \mathcal{Z}_\pi$.
2. *The stage cost function* $\ell(\cdot,\cdot)$ *is uniformly continuous in* $\mathcal{Z}_\pi$. *Therefore, there exist functions* $\sigma_{\ell,x}(\cdot), \sigma_{\ell,v}(\cdot) \in \mathcal{K}$ *such that*

$$|\ell(x, v) - \ell(\check{x}, \check{v})| \leq \sigma_{\ell,x}(\|x - \check{x}\|) + \sigma_{\ell,v}(\|v - \check{v}\|), \forall (x, v), (\check{x}, \check{v}) \in \mathcal{Z}_\pi.$$

3. *There exists a function* $\alpha_f(\cdot) \in \mathcal{K}$ *such that* $V_f(x) \leq \alpha_f(\|x\|), \ \forall x \in \Omega$.
4. *The terminal cost function* $V_f(\cdot)$ *is uniformly continuous in* Ω. *Therefore, there exists a function* $\sigma_f(\cdot) \in \mathcal{K}$ *such that*

$$|V_f(x) - V_f(\check{x})| \leq \sigma_f(\|x - \check{x}\|), \forall x, \check{x} \in \Omega.$$

Assumption 1.22 poses conditions on the terminal ingredients that are stricter than those listed in Sections 1.1 and 1.2. In detail, it requires the set Ω to be a contractive invariant set. Indeed, it is explicitly required that, for all the states $x \in \Omega$, the local control law $\pi(x, v_f)$ is able to bring the system state to the set $\mathcal{X}_f$, which is contained in Ω. The consequence of this is that the set Ω is a robust positively invariant set (RPI), the definition of which is provided in the Appendix A.1, for the system $x(k+1) = f\left(x(k), \pi\left(x(k), v_f\right), 0\right) + e$, where $e \in \mathcal{F}(N_p - 1)$. This is a needed condition to ensure the recursive feasibility and the stabilizing capabilities of the method. Assumption 1.23, instead, poses conditions on the ingredients of the cost function that are similar to the standard ones, as the only difference is that the usual continuity condition on $\ell(\cdot,\cdot)$ and $V_f(\cdot)$ is replaced by a uniform continuity assumption.

Next, we present the theorem that ensures the recursive feasibility of the analyzed tube-based MPC for regulation and the input-to-state stability of the closed-loop system.

Theorem 1.8: Stability of the tube-based MPC for the regulation of nonlinear systems

Under Assumptions 1.18-1.23, the tube-based MPC for the regulation of nonlinear systems is recursively feasible and the origin is input-to-state stable (ISS) for the closed-loop system.

Proof. We need to prove that the optimal cost function $V^*_{N_p}(\cdot)$ is an ISS-Lyapunov function, the definition of which is provided in the Appendix A.4. In order to do so, we first prove that $V^*_{N_p}(\cdot)$ admits two $\mathcal{K}$ functions as lower bound and upper bound, and then prove the descent property of $V^*_{N_p}(\cdot)$ and the recursive feasibility of the presented tube-based MPC scheme.

Lower and upper bounds for $V^*_{N_p}(\cdot)$. The existence of the two bounding functions is proved like in the Proof of Theorem 1.1. Given that the region of attraction is such that $\mathcal{X}_{N_p} \subseteq \mathcal{X}$, $V^*_{N_p}(x) \geq \ell(x,v) \geq \alpha_\ell(\|x\|)$, $\forall x \in \mathcal{X}_{N_p}$ due to Assumption 1.23. Then, due to standard arguments in the MPC theory, the upper bound on $V_f(\cdot)$ in Ω given by Assumption 1.23 ensures that there exists a function $\alpha_2(\cdot) \in \mathcal{K}$ such that $V^*_{N_p}(x) \leq \alpha_2(\|x\|), \forall x \in \mathcal{X}_{N_p}$.

Descent property of $V^*_{N_p}(\cdot)$ and recursive feasibility. Consider a feasible state $x(k) \in \mathcal{X}_{N_p}$ and the optimal control sequence $\mathbf{v}^*(k) = \left\{v^*(0|k), \ldots, v^*(N_p-1|k)\right\}$ resulting from the solution of $P_{N_p}(x(k))$, that is the control sequence leading to $V^*_{N_p}(x(k))$. We define the resulting optimal state sequence as

$$\hat{\mathbf{x}}^*(k) := \left\{\hat{x}^*(0|k), \hat{x}^*(1|k), \ldots, \hat{x}^*(N_p|k)\right\}.$$

Consider the uncertain state $x(k+1) = f\left(x(k), \pi\left(x(k), v^*(0|k)\right), w(k)\right)$. To prove recursive feasibility and the descent property of $V^*_{N_p}(\cdot)$, we make use of a candidate solution to $P_{N_p}\left(x(k+1)\right)$ that is given by

$$\mathbf{v}^+(k+1) := \left\{v^*(1|k), \ldots, v^*(N_p-1|k), v_f\right\}.$$

The perturbation $w(k)$ acting on the system causes the uncertain state $x(k+1)$ to differ from $\hat{x}^*(1|k)$. However, because of the uniform continuity of the state function $f(\cdot,\cdot,\cdot)$, we have that

$$\|x(k+1) - \hat{x}(1|k)\| \leq \sigma_w\left(\|w(k)\|\right) \leq \sigma_w(\bar{w}),$$

which means that $x(k+1) \in \hat{x}(1|k) \oplus \mathcal{F}(0)$ due to Assumption 1.20. However, because of the feasibility of $P_{N_p}(x(k))$, $(\hat{x}(1|k), v^*(1|k)) \in \mathcal{Z}_\pi(1) = \mathcal{Z}_\pi \ominus (\mathcal{R}(1) \times \{0\})$. This, together with the fact that $\mathcal{F}(0) \oplus \mathcal{R}(0) \subseteq \mathcal{R}(1)$, allows us to conclude that

$$(x(k+1), v^*(1|k)) \in \mathcal{Z}_\pi \ominus (\mathcal{R}(1) \times \{0\}) \oplus (\mathcal{R}(1) \times \{0\}) \ominus (\mathcal{R}(0) \times \{0\}) \subseteq \mathcal{Z}_\pi,$$

which proves that $x(k+1)$ verifies the state constraints. Moreover, since the condition $x(k+1) \in \hat{x}(1|k) \oplus \mathcal{F}(0)$ is verified, $\forall j \in [0, N_p - 1]$, we obtain that

$$\phi_\pi\left(j; x(k+1), \mathbf{v}^+(k+1), \mathbf{0}\right) \in \phi_\pi\left(j+1; x(k), \mathbf{v}^*(k), \mathbf{0}\right) \oplus \mathcal{F}(j). \tag{1.26}$$

However, due to the feasibility of $P_{N_p}(x(k))$, the optimal state trajectory $\hat{\mathbf{x}}^*(k)$ is such that

$$\left(\phi_\pi\left(j; x(k), \mathbf{v}^*(k), \mathbf{0}\right), v^*(j|k)\right) \in \mathcal{Z}_\pi(j),\ \forall j \in [0, N_p - 1]$$

and

$$\phi_\pi\left(N_p; x(k), \mathbf{v}^*(k), \mathbf{0}\right) \in \mathcal{X}_f.$$

Therefore, considering that $v^+(j-1|k+1) = v^*(j|k)$ for all $j \in [1, N_p - 1]$ and recalling that $\mathcal{F}(j) \oplus \mathcal{R}(j) \subseteq \mathcal{R}(j+1)$ by Assumption 1.21, (1.26) implies that, $\forall j \in [1, N_p - 1]$

$$\begin{aligned}
&\left(\phi_\pi\left(j; x(k+1), \mathbf{v}^+(k+1), \mathbf{0}\right), v^+(j|k+1)\right) \in \mathcal{Z}_\pi(j+1) \oplus (\mathcal{F}(j) \times \{0\}) \\
&\subseteq \mathcal{Z}_\pi \ominus (\mathcal{R}(j+1) \times \{0\}) \oplus (\mathcal{R}(j+1) \times \{0\}) \ominus (\mathcal{R}(j) \times \{0\}) \\
&\subseteq \mathcal{Z}_\pi(j).
\end{aligned} \tag{1.27}$$

Moreover, since $\hat{x}^*(N_p|k) = \phi_\pi(N_p; x(k), \mathbf{v}^*(k), \mathbf{0}) \in \mathcal{X}_f$, and provided that Assumption 1.22 ensures that the condition $\mathcal{X}_f \oplus \mathcal{F}(N_p - 1) \subseteq \Omega$ is met, we get that $\phi_\pi(N_p - 1; x(k+1), \mathbf{v}^+(k+1), \mathbf{0}) \in \Omega$. However, the input v_f, which is the last element of the candidate sequence $\mathbf{v}^+(k+1)$, makes the set Ω contractive. In particular, always due to Assumption 1.22, it ensures that $\phi_\pi(N_p; x(k+1), \mathbf{v}^+(k+1), \mathbf{0}) \in \mathcal{X}_f$. This finding, together with (1.27), proves that the candidate sequence $\mathbf{v}^+(k+1)$ is a feasible solution of $P_{N_p}(x(k+1))$, i.e. the presented MPC is recursively feasible.
To prove the descent property of $V^*_{N_p}(\cdot)$, analyzing $\Delta V_{N_p} \coloneqq V_{N_p}(x(k+1), \mathbf{v}^+(k+1)) - V^*_{N_p}(x(k))$ and defining $\hat{x}^+(j|k+1) \coloneqq \phi_\pi(j; x(k+1), \mathbf{v}^+(k+1), \mathbf{0})$, ΔV_{N_p} can be expressed as

$$\begin{aligned}
\Delta V_{N_p} = &\sum_{j=1}^{N_p-1} \left[\ell(\hat{x}^+(j-1|k+1), v^+(j-1|k+1)) - \ell(\hat{x}^*(j|k), v^*(j|k))\right] \\
&- \ell(x(k), v^*(0|k)) + \ell(\hat{x}^+(N_p - 1|k+1), v_f) \\
&+ V_f\left(\hat{x}^+(N_p|k+1)\right) - V_f\left(\hat{x}^*(N_p|k)\right).
\end{aligned} \tag{1.28}$$

Recall that the state function, the stage cost function and the terminal cost function are uniformly continuous thanks to Assumptions 1.18 and 1.23. Hence,

$$\begin{aligned}
&\ell(\hat{x}^{+}(j-1|k+1), v^{+}(j-1|k+1)) - \ell(\hat{x}^{*}(j|k), v^{*}(j|k)) \\
&\leq |\ell(\hat{x}^{+}(j-1|k+1), v^{+}(j-1|k+1)) - \ell(\hat{x}^{*}(j|k), v^{*}(j|k))| \\
&\leq \sigma_{\ell,x} \circ \sigma_x^{j-1} \circ \sigma_w(\|w(k)\|).
\end{aligned} \tag{1.29}$$

Furthermore,

$$\begin{aligned}
&V_f\left(\hat{x}^{+}(N_p|k+1)\right) - V_f\left(\hat{x}^{*}(N_p|k)\right) = V_f\left(\hat{x}^{+}(N_p|k+1)\right) - V_f\left(\hat{x}^{+}(N_p-1|k+1)\right) \\
&+ V_f\left(\hat{x}^{+}(N_p-1|k+1)\right) - V_f\left(\hat{x}^{*}(N_p|k)\right) \\
&\leq V_f\left(\hat{x}^{+}(N_p|k+1)\right) - V_f\left(\hat{x}^{+}(N_p-1|k+1)\right) \\
&+ |V_f\left(\hat{x}^{+}(N_p-1|k+1)\right) - V_f\left(\hat{x}^{*}(N_p|k)\right)| \\
&\leq V_f\left(\hat{x}^{+}(N_p|k+1)\right) - V_f\left(\hat{x}^{+}(N_p-1|k+1)\right) + \sigma_f \circ \sigma_x^{N_p-1} \circ \sigma_w(\|w(k)\|).
\end{aligned} \tag{1.30}$$

However, since $\hat{x}^{+}(N_p-1|k+1)$ has already been proved to stay in the contractive invariant set Ω, and given that $V_f(\cdot)$ is a CLF defined in Ω due to Assumption 1.22, we get that

$$V_f\left(\hat{x}^{+}(N_p|k+1)\right) - V_f\left(\hat{x}^{+}(N_p-1|k+1)\right) \leq -\ell(\hat{x}^{+}(N_p-1|k+1), v_f). \tag{1.31}$$

Putting together the results obtained in (1.29), (1.30) and (1.31), and recalling that $\ell(x,v) \geq \alpha_\ell(\|x\|)$ due to Assumption 1.23, we can rewrite (1.28) as

$$\begin{aligned}
\Delta V_{N_p} &\leq \sum_{j=1}^{N_p-1} \sigma_{\ell,x} \circ \sigma_x^{j-1} \circ \sigma_w(\|w(k)\|) - \ell(x(k), v^{*}(0|k)) + \sigma_f \circ \sigma_x^{N_p-1} \circ \sigma_w(\|w(k)\|) \\
&\leq -\alpha_\ell(\|x(k)\|) + \sum_{j=1}^{N_p-1} \sigma_{\ell,x} \circ \sigma_x^{j-1} \circ \sigma_w(\|w(k)\|) \\
&+ \sigma_f \circ \sigma_x^{N_p-1} \circ \sigma_w(\|w(k)\|).
\end{aligned} \tag{1.32}$$

Employing a function $\lambda(\cdot) \in \mathcal{K}$ such that $\lambda(\|w\|) \geq \sum_{j=1}^{N_p-1} \sigma_{\ell,x} \circ \sigma_x^{j-1} \circ \sigma_w(\|w\|) + +\sigma_f \circ \sigma_x^{N_p-1} \circ \sigma_w(\|w\|)$, (1.32) rewrites as

$$\Delta V_{N_p} \leq -\alpha_\ell(\|x(k)\|) + \lambda(\|w(k)\|). \tag{1.33}$$

Given that $V_{N_p}(x(k+1), \mathbf{v}^{+}(k+1))$ is not necessarily optimal, (1.33) implies that

$$V_{N_p}^{*}(x(k+1)) - V_{N_p}^{*}(x(k)) \leq \Delta V_{N_p} \leq -\alpha_\ell(\|x(k)\|) + \lambda(\|w(k)\|),$$

which proves that $V_{N_p}^{*}(\cdot)$ is an ISS-Lyapunov function. Thus, the origin is ISS for the closed-loop system under the described tube-based MPC for regulation. ∎

Comparing the Proof of Theorem 1.8 and the Proof of Theorem 1.1, it is straightforward to realize what the main difficulty is in proving the stability of a robust MPC. Both the proofs assume to consider a candidate solution for $P_{N_p}(x(k+1))$ that is almost identical to the optimal control sequence obtained by solving $P_{N_p}(x(k))$, the only difference residing in the first element of the sequence being dropped and an element consistent with the terminal ingredient conditions being added at the end. Then, it is proved that such candidate solution is feasible and ensures the desired descent property of $V^*_{N_p}(\cdot)$. Nevertheless, in the perturbation-free case such candidate solution provides a predicted state trajectory $\hat{\mathbf{x}}^+(k+1)$ that, with the exception of the terminal state, perfectly coincides with the optimal state trajectory that was obtained at time k, i.e. $\hat{\mathbf{x}}^*(k)$. This is not true in the perturbed case, due to the perturbation acting on the feedback state, that is the initial condition of the new prediction. That is the reason why it is necessary to make further assumptions on the dynamics of the system and the employed cost function, as we would like to ensure that the gap between such trajectories is bounded and that the impact of such gap on the optimal cost function $V^*_{N_p}(\cdot)$ is also limited. In this way, it becomes possible to resort to an input-to-state stability proof, which requires the nominal system to be asymptotically stable and the effect of the perturbations acting on the system to be bounded.

Part I

Theoretical contributions

Chapter 2. Robust MPC for tracking changing setpoints for perturbed nonlinear systems

2.1 Introduction

In Section 1.2 we addressed several MPC formulations for nonlinear systems that aim at tracking changing setpoints. In particular, we discussed the schemes presented in [90], which differ from each other for the way the terminal ingredients are designed and implemented. Nevertheless, such schemes assume to have no model mismatches and no external disturbances, while it would be desirable to have tracking MPC schemes for nonlinear systems that are designed to be robust with respect to bounded perturbations.

As anticipated in Section 1.3, some robust tracking MPC schemes were proposed for the control of linear systems subject to additive perturbations, e.g. [45, 88], while the design of robust tracking MPC schemes for nonlinear systems is still an active subject of investigation. Interesting results were obtained in [146] for the restricted family of nonlinear affine systems and slowly varying perturbations and in [31] for nonlinear systems that are subject to bounded additive perturbations.

In this chapter, we present a robust MPC for tracking changing setpoints for the control of a wide family of nonlinear systems and that is proved to be recursively feasible and to ensure the input-to-state stability of the closed-loop system, only requiring the perturbations acting on the system to be bounded. The method is then validated in simulation. The robust MPC for tracking belongs to the family of tube-based feedback MPC schemes, as it minimizes the objective function related to the nominal evolution of the system state and exploits a parameterized family of control laws to reduce the effect of the perturbations on the state predictions. This approach was inspired by [90] and makes assumptions on the continuity of the system state and output functions and the objective function that are similar (though not equal) to those made in [89] to ensure the input-to-state stability of the closed-loop system under a tube-based MPC for regulation. Moreover, stability of the closed-loop system is proved following a rigorous process inspired by [74], where a nonlinear MPC for tracking periodic signals is introduced. However, while [74] assumes to have no perturbations and to employ quadratic stage cost, terminal cost, and offset cost functions, in this chapter we provide a proof of input-to-state stability for perturbed systems, considering a wider family of cost functions. The results presented in this chapter were published in [116].

2.2 Problem formulation

We consider a nonlinear time-invariant discrete-time system described by the equations

$$\begin{aligned} x(k+1) &= f\big(x(k), u(k), w(k)\big), \\ y(k) &= h\big(x(k), u(k)\big), \end{aligned} \tag{2.1}$$

where $x(k) \in \mathbb{R}^n$ is the system state, $u(k) \in \mathbb{R}^m$ is the controlled input, $y(k) \in \mathbb{R}^p$ is the system output, and $w(k) \in \mathcal{W} \subset \mathbb{R}^r$ is an unknown but bounded perturbation that lies in a known hypercube $\mathcal{W} := \{w : \ |w_i| \leq \bar{w}_i, \ \forall i \in [1, r]\}$. The system state x is assumed to be measured at each sampling instant $k \geq 0$.

The system is subject to hard state and input constraints, namely

$$(x(k), u(k)) \in \mathcal{Z} := \mathcal{X} \times \mathcal{U}, \quad \forall k \geq 0. \tag{2.2}$$

All the equilibrium points of the nominal system (x_s, u_s, y_s) are such that

$$x_s = f(x_s, u_s, 0), \tag{2.3a}$$

$$y_s = h(x_s, u_s). \tag{2.3b}$$

Coherently with the previous chapters, a continuity assumption for the functions $f(\cdot,\cdot,\cdot)$ and $h(\cdot,\cdot)$ is provided, as well as the conditions that the constraint sets $\mathcal{X}$ and $\mathcal{U}$ have to fulfill to allow for the implementation of the here-presented tube-based MPC for tracking for nonlinear systems.

Assumption 2.1: Continuity of the state and output functions

The output function $h(\cdot,\cdot)$ is continuous at any equilibrium point. The state function $f(\cdot,\cdot,\cdot)$ is such that each component $f_i(\cdot,\cdot,\cdot)$, $\forall i \in [1, n]$, is component-wise uniformly continuous. Consequently, there exist functions $\sigma_{x,ia}(\cdot), \sigma_{u,ib}(\cdot), \sigma_{w,ic}(\cdot) \in \mathcal{K}$ such that, for all $i \in [1, n]$,

$$\begin{aligned} |f_i(x, u, w) - f_i(\check{x}, \check{u}, \check{w})| \leq & \sum_{a=1}^{n} \sigma_{x,ia}\left(x_a - \check{x}_a\right) + \sum_{b=1}^{m} \sigma_{u,ib}\left(|u_b - \check{u}_b|\right) \\ & + \sum_{c=1}^{r} \sigma_{w,ic}\left(|w_c - \check{w}_c|\right), \end{aligned} \tag{2.4}$$

for all (x, u, w) and $(\check{x}, \check{u}, \check{w})$ in $\mathcal{Z} \times \mathcal{W}$.

Assumption 2.2: Properties of the constraint sets

The sets $\mathcal{X}$ and $\mathcal{U}$ are closed and their interior is non-empty.

Assumption 2.2 is identical to Assumption 1.10, which was needed to prove the stability of a standard MPC for tracking. On the contrary, Assumption 2.1 is inspired by Assumption 1.18, which required the state function of a nonlinear perturbed system to be uniformly continuous, in order to enable the proof of input-to-state stability of a tube-based MPC for regulation. However, in this case we require a finer property, i.e. each component of $f(\cdot,\cdot,\cdot)$ has to be component-wise uniformly continuous, as this allows one to get less conservative tubes.

Remark 2.1

A particular case of component-wise uniformly continuous functions is that of component-wise Lipschitz continuous functions, which is a wide family of functions.

Indeed, in the common case in which the constraint set $\mathcal{Z}$ is compact, Lipschitz continuity is a condition that is easy to verify. Moreover, as proved in [102], Lipschitz continuity implies also component-wise Lipschitz continuity.

An algorithm for the computation of the component-wise Lipschit constants of a system of this kind is provided in Section 2.7.2.

Exactly as in Section 1.2, also the robust MPC for tracking presented in this chapter needs to know a set of feasible setpoints. In order to build it, we first define the restricted constraint set

$$\hat{\mathcal{Z}} := \{z : \ z + \delta_z \in \mathcal{Z}, \ \forall \delta_z \text{ s.t. } \|\delta_z\| \leq \epsilon\}, \tag{2.5}$$

where $\epsilon > 0$ can be arbitrarily small. The set (2.5) defines the feasible points (x, u) with inactive constraints. The sets defining the equilibrium points where the constraints are not active, which are supposed to be non-empty, are therefore defined as

$$\mathcal{Z}_s := \left\{(x, u) \in \hat{\mathcal{Z}} : \ x = f(x, u, 0)\right\}, \tag{2.6a}$$

$$\mathcal{Y}_s := \{y = h(x, u) : \ (x, u) \in \mathcal{Z}_s\}. \tag{2.6b}$$

The conditions that the output function $h(\cdot,\cdot)$ and the set $\mathcal{Y}_s$ have to undergo to allow for the correct design of the here-presented MPC are the same listed in Assumption 1.11 for a non-robust MPC for tracking. We report them in the next assumption for the sake of completeness.

Assumption 2.3: Properties of the output function

The system output is such that each steady output y_s uniquely defines an equilibrium point (x_s, u_s). Moreover, there exist a locally Lipschitz continuous function $g_x : \mathcal{Y}_s \to \mathbb{R}^n$ and a continuous function $g_u : \mathcal{Y}_s \to \mathbb{R}^m$ such that

$$x_s = g_x(y_s), \ u_s = g_u(y_s). \tag{2.7}$$

Remark 2.2

A sufficient condition for Assumption 2.3 to be verified, very similar to the condition given in Remark 1.4 for a non-robust MPC for tracking, is that the functions $f(x,u,0)$ and $h(x,u)$ are continuously differentiable and that the Jacobian matrix

$$\begin{bmatrix} (A(x_s,u_s) - I_n) & B(x_s,u_s) \\ C(x_s,u_s) & D(x_s,u_s) \end{bmatrix},$$

where

$$\begin{aligned} A(x_s,u_s) &= \frac{\partial f(x,u,0)}{\partial x}(x_s,u_s), & B(x_s,u_s) &= \frac{\partial f(x,u,0)}{\partial u}(x_s,u_s), \\ C(x_s,u_s) &= \frac{\partial h(x,u)}{\partial x}(x_s,u_s), & D(x_s,u_s) &= \frac{\partial h(x,u)}{\partial u}(x_s,u_s), \end{aligned}$$

is non-singular for all $(x_s,u_s) \in \mathcal{Z}_s$.

The aim of the remainder of the chapter is to design a state-feedback tube-based MPC for tracking based on nominal predictions that is able to stabilize the closed-loop system and to steer its output to a bounded set containing the admissible steady output that is the closest to the actual setpoint y_t. Furthermore, this property must hold even when the setpoint y_t is changed to a new constant value, i.e. y_t is a piece-wise constant reference signal.

2.3 Equilibrium-dependent feedback policies

The here-proposed MPC belongs to the family of feedback tube-based MPC schemes, and the size of its tubes depends on the functions $\sigma_{x,ia}(\cdot)$ and $\sigma_{w,ic}(\cdot)$, in particular on the rate at which these functions grow. For this reason, consistently with the rationale of feedback MPC, we exploit a family of pre-stabilizing feedback policies

$$u(k) = \pi(y_s, x(k), v(k)), \tag{2.8}$$

where $v(k) \in \mathbb{R}^s$, with the aim of decreasing the growth rate of $\sigma_{x,ia}(\cdot)$ and $\sigma_{w,ic}(\cdot)$, therefore reducing also the effect of uncertainty propagation. Nonetheless, in order to keep the component-wise uniform continuity of the closed-loop system under the feedback policies $\pi(y_s,x,v)$, these have to fulfill a uniform continuity assumption.

Assumption 2.4: Continuity of the feedback policies and existence of a continuous function $g_v : \mathcal{Y}_s \to \mathbb{R}^s$

The feedback policies $\pi(y_s, x, v)$ are uniformly continuous with respect to x and v. Furthermore, for each equilibrium point, there exists a unique v_s such that $u_s = \pi(y_s, x_s, v_s)$. There also exists a continuous function $g_v : \mathcal{Y}_s \to \mathbb{R}^s$, such that $v_s = g_v(y_s)$.

Remark 2.3

It is important to note that the policies $\pi(y_s, x, v)$ are not required to be uniformly continuous with respect to y_s. For this reason, it is possible to design different policies for different equilibria or groups of equilibria.

Within this feedback framework, the task of the MPC becomes that of finding the optimal value for y_s and $v(k)$. For this reason, we redefine the constraint sets $\mathcal{Z}$ and $\hat{\mathcal{Z}}$ as

$$\mathcal{Z}_\pi \coloneqq \{(x, v) : \ \exists y_s \in \mathcal{Y}_s \text{ s.t. } (x, \pi(y_s, x, v)) \in \mathcal{Z}\}, \tag{2.9}$$

and

$$\hat{\mathcal{Z}}_\pi \coloneqq \{(x, v) : \ (x, v) + \delta_z \in \mathcal{Z}_\pi, \ \forall \delta_z \text{ s.t. } \|\delta_z\| \le \epsilon\}. \tag{2.10}$$

The new sets of the admissible steady states and outputs are obtained as projections of the following set:

$$\mathcal{E}_{\pi,s} \coloneqq \Big\{(x, v, y) : (x, v) \in \hat{\mathcal{Z}}_\pi, \ x = f(x, \pi(y, x, v), 0), y = h(x, \pi(y, x, v))\Big\}.$$

In particular,

$$\mathcal{Z}_{\pi,s} \coloneqq \big\{(x, v) : \ \exists (x, v, y) \in \mathcal{E}_{\pi,s}\big\}, \tag{2.11}$$

and

$$\mathcal{Y}_{\pi,s} \coloneqq \big\{y : \ \exists (x, v, y) \in \mathcal{E}_{\pi,s}\big\}. \tag{2.12}$$

The previously defined sets play an essential role in the design of the proposed robust MPC for tracking.

2.4 Nominal model predictive control for tracking

As for the nonlinear tracking MPC formulations introduced in Section 1.2, i.e. [44, 45, 74, 86, 88, 90, 146], the here-proposed MPC is given the ability to steer the system output to a set containing the admissible steady output that is the closest to the real setpoint y_t, and to keep its feasibility properties also when y_t changes. This is done by adding an artificial setpoint y_s as an extra decision variable, by designing a cost function that includes a term $V_O(y_s - y_t)$, which penalizes the deviation between the artificial reference and the actual setpoint, and by designing an invariant set for tracking to be used as terminal constraint. Next, the fundamental ingredients of the optimization problem that has to be solved at each sampling instant are presented and explained.

For given state x and setpoint y_t, considering the general case where the control and prediction horizons N_c and N_p are different, the cost function of the proposed MPC is defined as

$$\begin{aligned} V_{N_c,N_p}\left(x(k), y_t, \mathbf{v}(k), y_s(k)\right) \coloneqq & \sum_{j=0}^{N_c-1} \ell\big(\hat{x}(j|k) - x_s(k), v(j|k) - v_s(k)\big) \\ & + \sum_{j=N_c}^{N_p-1} \ell\big(\hat{x}(j|k) - x_s(k), v_f(y_s(k)) - v_s(k)\big) \\ & + V_f\big(\hat{x}(N_p|k) - x_s(k), y_s(k)\big) + V_O(y_s(k) - y_t) \end{aligned} \tag{2.13}$$

where $\mathbf{v}(k) \coloneqq \{v(0|k), \dots, v(N_c - 1|k)\}$ is the sequence of control inputs to be optimized, $v_f(y_s(k))$ is an input such that $u = \pi\left(y_s(k), x, v_f(y_s(k))\right)$, and whose properties will be explained later in the text, $y_s(k)$ is the artificial reference, such that $x_s(k) = g_x(y_s(k))$, $u_s(k) = g_u(y_s(k))$, and $v_s(k) = g_v(y_s(k))$, while $\hat{x}(j|k)$, for $j = 1, \dots, N_p$, are nominal state predictions, i.e.

$$\hat{x}(j+1|k) = f\left(\hat{x}(j|k), \pi\left(y_s(k), \hat{x}(j|k), v(j|k)\right), 0\right).$$

The stage cost function $\ell(x - x_s, v - v_s)$ and the terminal cost function $V_f(x - x_s, y_s)$ penalize the tracking error with respect to the artificial reference y_s, while $V_O(y_s - y_t)$ penalizes the distance between the artificial reference y_s and the actual setpoint y_t.

The optimization problem $P_{N_c,N_p}(x(k), y_t)$ that is solved at each sampling instant by the here-proposed MPC is the following:

$$\min_{(\mathbf{v}(k), y_s(k))} \; V_{N_c,N_p}(x(k), y_t, \mathbf{v}(k), y_s(k)) \tag{2.14a}$$

subject to :

$$\hat{x}(0|k) = x(k), \tag{2.14b}$$

$$\hat{x}(j+1|k) = f\left(\hat{x}(j|k), \pi\left(y_s(k), \hat{x}(j|k), v(j|k)\right), 0\right), \forall j \in [0, N_c - 1] \tag{2.14c}$$

$$\big(\hat{x}(j|k), v(j|k)\big) \in \mathcal{Z}_\pi(j), \;\; \forall j \in [0, N_c - 1] \tag{2.14d}$$

$$\hat{x}(j+1|k) = f\left(\hat{x}(j|k), \pi\left(y_s(k), \hat{x}(j|k), v_f(y_s(k))\right), 0\right), \forall j \in [N_c, N_p - 1] \tag{2.14e}$$

$$\big(\hat{x}(j|k), v_f(y_s(k))\big) \in \mathcal{Z}_\pi(j), \;\; \forall j \in [N_c, N_p - 1] \tag{2.14f}$$

$$x_s(k) = g_x(y_s(k)), \;\; v_s(k) = g_v(y_s(k)), \tag{2.14g}$$

$$\big(\hat{x}(N_p|k), y_s(k)\big) \in \mathcal{T}, \tag{2.14h}$$

where the set sequence $\{\mathcal{Z}_\pi(j)\}_{\geq 0}$ and the the terminal constraint set $\mathcal{T}$, which shares properties with both the invariant sets for tracking explained in Definition 1.1 and the RPIs addressed in Definition A.3, will be defined later in the text. The optimal solution to this optimization problem and the optimal cost function are denoted as $\left(\mathbf{v}^0(x(k), y_t), y_s^0(x(k), y_t)\right)$, or $(\mathbf{v}^0(k), y_s^0(k))$ for brevity, and $V^0_{N_c,N_p}(x(k), y_t)$, respectively. Considering the receding horizon policy, the control law provided by the proposed MPC is defined as

$$\kappa_{N_c,N_p}(x(k), y_t) := \pi\left(y_s^0(k), x(k), v^0(0|k)\right),$$

where $v^0(0|k)$ is the first element of $\mathbf{v}^0(x(k), y_t)$. Note that, consistently with the tracking schemes presented in Section 1.2, an extended terminal constraint is imposed on both the terminal state $\hat{x}(N_p|k)$ and the artificial reference y_s, and that none of the constraints involved in $P_{N_c,N_p}(x(k), y_t)$ depends on the actual setpoint y_t, which leads to the existence of a region of attraction $\mathcal{X}_{N_c,N_p}$ where $P_{N_c,N_p}(x(k), y_t)$ is feasible for all $y_t \in \mathbb{R}^p$.

2.5 Robust design of the controller

The robustness of the MPC for tracking is ensured by imposing that the nominal system trajectory stays inside the set sequence $\{\mathcal{Z}_\pi(j)\}_{j\geq 0}$ of tightened constraints [87, 89]. Before explaining in detail how to compute such sequence, we introduce the following notation: with $\phi_\pi\,(j; x, y_s, \mathbf{v}, \mathbf{w})$ we denote the solution of system (2.1) at sampling time j given the initial state x, the feedback policy $\pi(y_s, x, v)$, where v comes from the sequence $\mathbf{v}$, and the uncertainty signal $\mathbf{w}$.

The computation of $\{\mathcal{Z}_\pi(j)\}_{j\geq 0}$ requires the existence of the set sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$, whose properties are outlined in the following two assumptions, that are an adaptation of Assumptions 1.20 and 1.21 to the tracking context and to component-wise uniformly continuous systems.

Assumption 2.5: Bounds on the effect of the initial condition on the state trajectory

The sequence $\{\mathcal{F}(j)\}_{j\geq 0}$ is such that:

- $\mathcal{F}(0) := \{x \in \mathbb{R}^n : |x_i| \leq \sum_{a=1}^{r} \sigma_{w,ia}(\bar{w}_a), \forall i \in [1, n]\}$;
- $\forall j > 0$, *the sets* $\mathcal{F}(j)$ *ensure that, for all* $y_s \in \mathcal{Y}_{\pi,s}$, *for every feasible* $(x, \mathbf{v})$, *and for every* $\check{x}$ *such that* $\check{x} - x \in \mathcal{F}(0)$, *the condition*

$$\phi_\pi\left(j; \check{x}, y_s, \mathbf{v}, \mathbf{0}\right) \in \phi_\pi\left(j; x, y_s, \mathbf{v}, \mathbf{0}\right) \oplus \mathcal{F}(j)$$

is verified.

Assumption 2.6: Bounds on the effect of the perturbations on the state trajectory

The set sequence $\{\mathcal{R}(j)\}_{j\geq 0}$ is such that, defining $\mathcal{R}(0) \coloneqq \{0\}$, it ensures that for all $y_s \in \mathcal{Y}_{\pi,s}$:

1. *for every feasible $(x, \mathbf{v})$, the sets $\mathcal{R}(j)$, for $j > 0$, are such that*

$$\phi_\pi\,(j; x, y_s, \mathbf{v}, \mathbf{w}) \in \phi_\pi\,(j; x, y_s, \mathbf{v}, \mathbf{0}) \oplus \mathcal{R}(j)\ \forall \mathbf{w} \in \mathcal{W}^j;$$

2. *$\mathcal{F}(j) \oplus \mathcal{R}(j) \subseteq \mathcal{R}(j+1)$, $\forall j \geq 0$.*

Even though the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ defined by Assumptions 2.5 and 2.6 have to undergo stronger conditions than those defined in Section 1.3 for a tube-based MPC for regulation, as such conditions have to be met for all $y_s \in \mathcal{Y}_{\pi,s}$, their meaning is exactly the same. Therefore, the existence of $\{\mathcal{F}(j)\}_{j\geq 0}$ ensures that, starting from close initial conditions, the distance between two different trajectories is bounded, while $\{\mathcal{R}(j)\}_{j\geq 0}$ bounds the distance between the nominal trajectories computed by the MPC and the possible trajectories of the perturbed system.

Remark 2.4

Note that the conditions of the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ are required to hold for all the feedback policies $\pi(y_s, x, v)$. From a practical point of view, it is possible to compute a different pair of sequences $\{\mathcal{F}(y_s, j)\}_{j\geq 0}$ and $\{\mathcal{R}(y_s, j)\}_{j\geq 0}$ for each $y_s \in \mathcal{Y}_{\pi,s}$ and finally compute $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ as

$$\mathcal{F}(j) = \cup_{y_s \in \mathcal{Y}_t} \mathcal{F}(y_s, j),\ \ \forall j \geq 0,$$

and

$$\mathcal{R}(j) = \cup_{y_s \in \mathcal{Y}_t} \mathcal{R}(y_s, j),\ \ \forall j \geq 0.$$

A method to compute these sequences in the particular case of component-wise Lipschitz continuous systems is reported in Section 2.7.3.

Given the sequence $\{\mathcal{R}(j)\}_{j\geq 0}$, the tightened constraint set sequence $\{\mathcal{Z}_\pi(j)\}_{j\geq 0}$ is defined as

$$\mathcal{Z}_\pi(j) \coloneqq \mathcal{Z}_\pi \ominus (\mathcal{R}(j) \times \{0\})\,. \tag{2.15}$$

Now that the tightened constraints introduced in the control problem $P_{N_c,N_p}(\cdot,\cdot)$ have been defined, the set of feasible setpoints $\mathcal{Y}_t$ is introduced as

$$\mathcal{Y}_t \coloneqq \Big\{ y_s \in \mathcal{Y}_{\pi,s} : \big(g_x(y_s), g_v(y_s)\big) \in \mathcal{Z}_\pi(N_p) \Big\}. \tag{2.16}$$

2.6 Recursive feasibility and convergence to a neighborhood of the best admissible steady output

In order to prove recursive feasibility of $P_{N_c,N_p}(\cdot,\cdot)$, closed-loop stability, and convergence of the closed-loop system output to a neighborhood of the best admissible setpoint, the stage cost function, the terminal ingredients and the offset cost function must verify some conditions, which are listed in the following assumptions, before presenting the main result of this chapter.

In order to ensure the recursive feasibility of $P_{N_c,N_p}(\cdot,\cdot)$, the terminal ingredients $\mathcal{T}$ and $V_f(\cdot)$ must satisfy the following assumption.

Assumption 2.7: Terminal ingredient properties

The inputs $v_f(y_s) \in \mathbb{R}^s$, the set $\mathcal{T} := \left(\bigcup_{y_s \in \mathcal{Y}_t} \mathcal{X}_f(y_s)\right) \times \mathcal{Y}_t$ and the sets $\Omega(y_s)$ are such that:

1. *$\Omega(y_s)$ and $\mathcal{X}_f(y_s)$ are positively invariant sets for the nominal system $\hat{x}(k+1) = f\left(\hat{x}(k), \pi\left(y_s, \hat{x}(k), v_f(y_s)\right), 0\right)$;*
2. *$\Omega(y_s) \times \{v_f(y_s)\} \subseteq \mathcal{Z}_\pi(N_p - 1)$;*
3. *$\mathcal{X}_f(y_s) \times \{v_f(y_s)\} \subseteq \mathcal{Z}_\pi(N_p)$;*
4. *$\mathcal{X}_f(y_s) \oplus \mathcal{F}(N_p - 1) \subseteq \Omega(y_s)$;*
5. *$f\left(\hat{x}, \pi\left(y_s, \hat{x}, v_f(y_s)\right), 0\right) \in \mathcal{X}_f(y_s)$, for all $\hat{x} \in \Omega(y_s)$.*

Moreover, the terminal cost function $V_f(\cdot,\cdot)$ is a CLF such that, for all $y_s \in \mathcal{Y}_t$ and for all $x \in \Omega(y_s)$, it ensures

$$V_f\left(f\left(x, \pi\left(y_s, x, v_f(y_s)\right), 0\right) - x_s, y_s\right) - V_f(x - x_s, y_s) \leq -\ell\left(x - x_s, v_f(y_s) - v_s\right).$$

The conditions given by Assumption 2.7 ensure that the set $\mathcal{T}$ is a *robust invariant set for tracking*, the definition of which is provided hereafter.

Definition 2.1: Robust invariant set for tracking

For a given set of constraints $\mathcal{Z}_\pi$, a set of feasible setpoints $\mathcal{Y}_t \subseteq \mathcal{Y}_{\pi,s}$, and a local control law $v = \kappa(x, y_s)$, a set $\mathcal{T} \subset \mathbb{R}^n \times \mathbb{R}^s$ is an (admissible) robust invariant set for tracking for the system $x^+ = f(x, \pi(y_s, x, v), w)$ if, for all $(x, y_s) \in \mathcal{T}$, we have that $(x, \kappa(x, y_s)) \in \mathcal{Z}_\pi$, $y_s \in \mathcal{Y}_t$, and $(f(x, \pi(y_s, x, \kappa(x, y_s)), w), y_s) \in \mathcal{T}$, $\forall w \in \mathcal{W}$.

In detail, the robustness of $\mathcal{T}$, which is used as terminal inequality constraint on both the terminal state $\hat{x}(N_p|k)$ and the artificial reference y_s, is with respect to the terminal state. Indeed, the sets $\Omega(y_s)$ are required to be robustly invariant for the system $x(k+1) = f\big(x(k), \pi(y_s, x(k), v_f(y_s)), 0\big) + e$, where $e \in \mathcal{F}(N_p - 1)$, while $\mathcal{Y}_t$, i.e. the set of feasible setpoints, is required to fulfill the typical conditions of a standard invariant set for tracking. Assumption 2.7 also requires to design control inputs $v_f(y_s)$ that are able to locally asymptotically stabilize the system to any steady state contained in $\mathcal{Z}_{\pi,s}$.

Next, we state the conditions that the cost function must verify to enable the proof of recursive feasibility and input-to-state stability of the closed-loop system.

Assumption 2.8: Stage cost function and terminal cost function bounds

The stage cost function $\ell(\cdot,\cdot)$ and the terminal cost function $V_f(\cdot,\cdot)$ meet the next conditions:

1. *There exists a function $\alpha_\ell(\cdot) \in \mathcal{K}_\infty$ such that $\ell(x, v) \geq \alpha_\ell(\|x\|)$ for all $(x, v) \in \mathbb{R}^n \times \mathbb{R}^s$. The stage cost function $\ell(\cdot,\cdot)$ is uniformly continuous in $\mathcal{Z}_\pi$. Therefore, there exist functions $\sigma_{\ell,x}(\cdot), \sigma_{\ell,v}(\cdot) \in \mathcal{K}$ such that*

$$|\ell(x, v) - \ell(\check{x}, \check{v})| \leq \sigma_{\ell,x}(\|x - \check{x}\|) + \sigma_{\ell,v}(\|v - \check{v}\|), \forall (x, v), (\check{x}, \check{v}) \in \mathcal{Z}_\pi.$$

2. *There exists a quadratic function $\alpha_f(\|x\|) \coloneqq \bar{\alpha}_f\|x\|^2$ such that $V_f(x - x_s, y_s) \leq \bar{\alpha}_f\|x - x_s\|^2$ for all $y_s \in \mathcal{Y}_t$ and for all $x \in \Omega(y_s)$.*

3. *The terminal cost function $V_f(\cdot,\cdot)$ is uniformly continuous with respect to its first argument, namely there exists a function $\sigma_f(\cdot) \in \mathcal{K}$ such that*

$$|V_f(x - x_s, y_s) - V_f(\check{x} - x_s, y_s)| \leq \sigma_f(\|x - \check{x}\|).$$

Remark 2.5

The function $V_f(x - x_s, y_s)$ is not required to be continuous with respect to y_s. This allows the designer to choose different terminal cost functions for different equilibria or groups of equilibria. Nonetheless, Assumption 2.8 requires the functions $V_f(x - x_s, y_s)$ to share an upper bound and a function $\sigma_f(\cdot)$ defining a common uniform continuity property. Of course, when it is possible, a simple way to verify such conditions is to use a unique terminal cost function for all the feasible setpoints $y_s \in \mathcal{Y}_t$.

Several algorithms for the combined design of the pre-stabilizing control laws $\pi(\cdot,\cdot,\cdot)$, the terminal cost functions $V_f(\cdot,\cdot)$ and the robust invariant set for tracking $\mathcal{T}$ are provided in Section 2.7.1.

In order to ensure that the closed-loop system output converges to a neighborhood of the best admissible steady output, it is fundamental that the offset function $V_O(\cdot)$ and the set $\mathcal{Y}_t$ verify the following conditions.

Assumption 2.9: Properties of the offset cost function and the feasible setpoint set

1. *The set of feasible setpoints $\mathcal{Y}_t$ is a convex and bounded subset of $\mathcal{Y}_{\pi,s}$.*

2. *The offset cost function $V_O : \mathbb{R}^p \to \mathbb{R}$ is a convex positive definite function such that the minimizer*

$$y_s^* = \arg \min_{y_s \in \mathcal{Y}_t} V_O(y_s - y_t) \tag{2.17}$$

is unique. Moreover, there exists a $\mathcal{K}_\infty$ function $\alpha_O(\cdot)$ such that

$$V_O(y_s - y_t) - V_O(y_s^* - y_t) \geq \alpha_O(\|y_s - y_s^*\|) \; \forall y_s \in \mathcal{Y}_t.$$

Remark 2.6

Assumption 2.9 resembles Assumption 1.14, but requires an additional condition that is more restrictive, that is the boundedness of the set $\mathcal{Y}_t$. This can be obtained by either having an output function $h(\cdot,\cdot)$ that is bounded in $\mathcal{Z}$, or by accepting to let the MPC steer the closed-loop output to setpoints belonging to a bounded set $\mathcal{Y}_t$ even when $\mathcal{Y}_s$ has no bounds. In case of a compact constraint set $\mathcal{Z}$, $\mathcal{Y}_s$ is bounded, therefore allowing the designer to choose $\mathcal{Y}_t = \mathcal{Y}_s$ if $\mathcal{Y}_s$ is also convex.

Assumption 2.9 implies that, given the boundedness of $\mathcal{Y}_t$, there exists $V_{O,\max} > 0$ such that $V_O(y_s - \check{y}_s) \leq V_{O,\max}$ for any $y_s, \check{y}_s \in \mathcal{Y}_t$.

For reasons that will become more clear later in the text, in order to prove the input-to-state stability of the closed-loop system, it is necessary to make a further (non-restrictive) assumption on the function $\alpha_O(\cdot)$. For the sake of simplicity, the next assumption makes use of the set $\mathcal{S}$, which is defined as

$$\mathcal{S} \coloneqq \{\|y_s - \check{y}_s\| : y_s, \check{y}_s \in \mathcal{Y}_t\}.$$

Note that, due to the boundedness of $\mathcal{Y}_t$, $\mathcal{S} \subset \mathbb{R}$ is a bounded interval of real numbers, in detail $\mathcal{S} \equiv [0, \sup \|y_s - \check{y}_s\|]$, with $y_s, \check{y}_s \in \mathcal{Y}_t$.

Assumption 2.10: Joint conditions on $V_f(\cdot,\cdot)$ and $V_O(\cdot)$

Given the Lipschitz constant L_g, for which $\|x_s - \check{x}_s\| \leq L_g \|y_s - \check{y}_s\|$, the functions $V_f(\cdot,\cdot)$ and $V_O(\cdot)$ are such that

$$\mathfrak{b}_1 := \inf\left\{\frac{\alpha_O(s)}{4\bar{\alpha}_f L_g^2 s^2} \ :\ s \in \mathcal{S} \setminus \{0\}\right\} > 0,$$

and

$$\mathfrak{b}_2 := \inf\left\{\frac{\alpha_O(s_2) - \alpha_O(s_1)}{4\bar{\alpha}_f L_g^2 s_2^2 - 4\bar{\alpha}_f L_g^2 s_1^2} : s_1, s_2 \in \mathcal{S}, s_2 > s_1\right\} > 0.$$

Remark 2.7

Thanks to the boundedness of $\mathcal{S}$, Assumption 2.10 is not particularly restrictive, as it can easily be verified by choosing an offset function $V_O(\cdot)$ such that, for $s \to 0^+$, $\alpha_O(s)$ goes to 0 at most as rapidly as s^2.

Examples of functions that verify the last conditions are quadratic functions and polynomials of degree equal to or lower than 2.

In the following, we analyze how the aforementioned assumptions are sufficient to ensure the recursive feasibility and the stabilizing capabilities of the robust MPC for tracking for perturbed nonlinear systems. Given that the proof of stability of the closed loop is complex, we first address recursive feasibility in Lemma 2.1.

Lemma 2.1: Recursive feasibility of the robust MPC for tracking for nonlinear systems

Consider the system $x(k+1) = f\big(x(k), \pi(y_s, x(k), v(k)), w(k)\big)$ and the set sequence $\{\mathcal{Z}_\pi(j)\}_{j\geq 0}$, based on $\{\mathcal{R}(j)\}_{j\geq 0}$ and $\{\mathcal{F}(j)\}_{j\geq 0}$, which satisfy Assumptions 2.5 and 2.6. Let the triplet $\big(v_f(y_s^0), \Omega(y_s^0), \mathcal{T}\big)$ fulfill Assumption 2.7. Consider now a feasible state $x(k) \in \mathcal{X}_{N_c,N_p}$ and the optimal solution $\big(\mathbf{v}^0(k), y_s^0(k)\big)$ to the problem $P_{N_c,N_p}(x(k), y_t)$. Let $x(k+1)$ be the next uncertain state and define a candidate artificial reference $y_s^+(k+1) = y_s^0(k)$ and a sequence of inputs $\mathbf{v}^+(k+1) := \{v^+(0|k+1), \ldots, v^+(N_c-1|k+1)\} = \{v^0(1|k), \ldots, v^0(N_c-1|k), v_f(y_s^+(k+1))\}$. Then the following properties hold:

1. $\big(\phi_\pi(j; y_s^+(k+1), x(k+1), \mathbf{v}^+(k+1), \mathbf{0}), v^+(j|k+1)\big) \in \mathcal{Z}_\pi(j)$,

2. $\big(\phi_\pi(N_p; y_s^+(k+1), x(k+1), \mathbf{v}^+(k+1), \mathbf{0}), y_s^+(k+1)\big) \in \mathcal{T}$.

Proof. **First statement:** Since $\phi_\pi(j; y_s^+(k+1), x(k+1), \mathbf{v}^+(k+1), \mathbf{0}) - \phi_\pi(j+1; y_s^0(k), x(k), \mathbf{v}^0(k), \mathbf{0}) \in \mathcal{F}(j)$, we have that

$$\begin{aligned}\phi_\pi(j; y_s^+(k+1), x(k+1), \mathbf{v}^+(k+1), \mathbf{0}) &\in \phi_\pi(j+1; y_s^0(k), x(k), \mathbf{v}^0(k), \mathbf{0}) \oplus \mathcal{F}(j) \\ &\subseteq \phi_\pi(j+1; y_s^0(k), x(k), \mathbf{v}^0(k), \mathbf{0}) \oplus \mathcal{R}(j+1) \ominus \mathcal{R}(j).\end{aligned}$$

Since $\big(\phi_\pi(j+1; y_s^0(k), x(k), \mathbf{v}^0(k), \mathbf{0}), v^0(j+1|k)\big) \in \mathcal{Z}_\pi(j+1)$, we have that

$$\begin{aligned}\big(\phi_\pi(j; y_s^+(k+1), x(k+1), \mathbf{v}^+(k+1), \mathbf{0}), v^0(j+1|k)\big) \in \Big(&\mathcal{Z}_\pi(j+1) \\ &\oplus \big(\mathcal{R}(j+1) \times \{0\}\big) \\ &\ominus \big(\mathcal{R}(j) \times \{0\}\big)\Big).\end{aligned}$$

By definition,

$$\mathcal{Z}_\pi(j+1) \oplus \big(\mathcal{R}(j+1) \times \{0\}\big) \subseteq \mathcal{Z}_\pi$$

and $v^0(j+1|k) = v^+(j|k)$, which implies that

$$\big(\phi_\pi(j; y_s^+(k+1), x(k+1), \mathbf{v}^+(k+1), \mathbf{0}), v^+(j|k+1)\big) \in \mathcal{Z}_\pi \ominus \big(\mathcal{R}(j) \times \{0\}\big) \subseteq \mathcal{Z}_\pi(j).$$

Second statement: The feasibility of the solution $\big(\mathbf{v}^0(k), y_s^0(k)\big)$ implies that

$$\big(\phi_\pi(N_p; y_s^0(k), x(k), \mathbf{v}^0(k), \mathbf{0}), y_s^0(k)\big) \in \mathcal{T}.$$

In particular, $\phi_\pi(N_p; y_s^0(k), x(k), \mathbf{v}^0(k), \mathbf{0}) \in \mathcal{X}_f(y_s^0(k))$ and $y_s^0(k) \in \mathcal{Y}_{\pi,s}$. On the other hand, from the definition of set $\mathcal{F}(N_p-1)$, we have that

$$\begin{aligned}\phi_\pi(N_p-1; y_s^+(k+1), x(k+1), \mathbf{v}^+(k+1), \mathbf{0}) \in\; &\phi_\pi(N_p; y_s^0(k), x(k), \mathbf{v}^0(k), \mathbf{0}) \\ &\oplus \mathcal{F}(N_p-1).\end{aligned}$$

Then

$$\phi_\pi(N_p-1; y_s^+(k+1), x(k+1), \mathbf{v}^+(k+1), \mathbf{0}) \in \mathcal{X}_f(y_s^+(k+1)) \oplus \mathcal{F}(N_p-1).$$

Since $y_s^+(k+1) = y_s^0(k)$, $\mathcal{X}_f(y_s^+(k+1)) = \mathcal{X}_f(y_s^0(k))$ and $y_s^+(k+1) \in \mathcal{Y}_{\pi,s}$, that is $\big(\phi_\pi(N_p; y_s^+(k+1), x(k+1), \mathbf{v}^+(k+1), \mathbf{0}), y_s^+(k+1)\big) \in \mathcal{T}$. ∎

The following theorem ensures the input-to-state stability of the closed-loop system controlled by the proposed robust MPC for tracking and convergence of the system output to a neighborhood of the best admissible steady output.

Theorem 2.1: Stability of the tube-based MPC for tracking for perturbed nonlinear systems

Suppose that Assumptions 2.1-2.10 hold, and consider a given constant setpoint y_t. Then for any feasible initial state $x \in \mathcal{X}_{N_c,N_p}$, the system controlled by $\kappa_{N_c,N_p}(x, y_t)$ fulfills the constraints throughout the time, the equilibrium $x_s^* = g_x(y_s^*)$ is input-to-state stable for the closed-loop system, whose output converges to a bounded set containing the best admissible steady output y_s^*.

Proof. This proof is divided in two parts. First, the recursive feasibility of $P_{N_c,N_p}(\cdot,\cdot)$ is proved. In the second part, the input-to-state stability of x_s^* and the convergence to a neighborhood of the best admissible steady output y_s^* are proved.

Recursive feasibility: the proof of recursive feasibility is given in Lemma 2.1.

Input-to-state stability and convergence: Let (x_s^*, u_s^*) be the equilibrium point given by $x_s^* = g_x(y_s^*)$ and $u_s^* = g_u(y_s^*)$. The goal is to show that the closed-loop system is ISS and converges to a neighborhood of the best feasible steady output y_s^*. To this end, we first show that the function

$$W(x(k), y_t) := V^0_{N_c,N_p}(x(k), y_t) - V_O(y_s^* - y_t)$$

nominally decreases in $\mathcal{X}_{N_c,N_p}$ with respect to the equilibrium $(x_s^0(k), u_s^0(k))$, and then prove that the same happens also with respect to the optimal couple (x_s^*, u_s^*). Defining $\hat{x}^0(j|k) = \phi_\pi(j; y_s^0(k), x(k), \mathbf{v}^0(k), \mathbf{0})$ and $\hat{x}^+(j|k+1) = \phi_\pi(j; y_s^+(k+1), x(k+1), \mathbf{v}^+(k+1), \mathbf{0})$, then

$$\begin{aligned}
&V_{N_c,N_p}(x(k+1), y_t, \mathbf{v}^+(k+1), y_s^+(k+1)) \\
&= \sum_{j=0}^{N_c-1} \ell\big(\hat{x}^+(j|k+1) - x_s^+(k+1), v^+(j|k+1) - v_s^+(k+1)\big) \\
&+ \sum_{j=N_c}^{N_p-1} \ell\big(\hat{x}^+(j|k+1) - x_s^+(k+1), v_f(y_s^+(k+1)) - v_s^+(k+1)\big) \\
&+ V_f\big(\hat{x}^+(N_p|k+1) - x_s^+(k+1), y_s^+(k+1)\big) + V_O(y_s^+(k+1) - y_t).
\end{aligned}$$

From the component-wise uniform continuity of the model, we have that, for all $i \in [1, n]$,

$$|\hat{x}_i^+(0|k+1) - \hat{x}_i^0(1|k)| \le \sum_{a=1}^{r} \sigma_{w,ia}(w_a(k)) =: c_{i,0}(w(k)).$$

In the same way,

$$\begin{aligned}
|\hat{x}_i^+(1|k+1) - \hat{x}_i^0(2|k)| &\le \sum_{b=1}^{n} \sigma_{x,ib}\left(\sum_{a=1}^{r} \sigma_{w,ba}(w_a(k))\right) \\
&= \sum_{b=1}^{n} \sigma_{x,ib}(c_{b,0}(w(k))) =: c_{i,1}(w(k)).
\end{aligned}$$

Generalizing for all $j > 0$,

$$|\hat{x}_i^+(j|k+1) - \hat{x}_i^0(j+1|k)| \leq \sum_{b=1}^{n} \sigma_{x,ib}\left(c_{b,j-1}(w(k))\right) =: c_{i,j}(w(k)).$$

Then, considering the uniform continuity of the stage cost function $\ell(\cdot,\cdot)$ and the uniform continuity of the terminal cost $V_f(\cdot, y_s)$ with respect to x, we obtain:

$$\begin{aligned}
&W(x(k+1), y_t) - W(x(k), y_t) = V^0_{N_c,N_p}(x(k+1), y_t) - V^0_{N_c,N_p}(x(k), y_t)\\
&\leq V_{N_c,N_p}(x(k+1), y_t, \mathbf{v}^+(k+1), y_s^+(k+1)) - V^0_{N_c,N_p}(x(k), y_t)\\
&= \sum_{j=0}^{N_c-2}\Big[\ell\big(\hat{x}^+(j|k+1) - x_s^+(k+1), v^+(j|k+1) - v_s^+(k+1)\big)\\
&- \ell\big(\hat{x}^0(j+1|k) - x_s^0(k), v^0(j+1|k) - v_s^0(k)\big)\Big]\\
&+ \sum_{j=N_c-1}^{N_p-2}\Big[\ell\big(\hat{x}^+(j|k) - x_s^+(k+1), v_f(y_s^+(k+1)) - v_s^+(k+1)\big)\\
&- \ell\big(\hat{x}^0(j+1|k) - x_s^0(k), v_f(y_s^0(k)) - v_s^0(k)\big)\Big]\\
&- \ell\big(x(k) - x_s^0(k), v^0(0|k) - v_s^0(k)\big)\\
&+ V_f\big(\hat{x}^+(N_p|k+1) - x_s^+(k+1), y_s^+(k+1)\big) - V_f\big(\hat{x}^0(N_p|k) - x_s^0(k), y_s^0(k)\big)\\
&+ \ell\big(\hat{x}^+(N_p-1|k+1) - x_s^+(k+1), v_f(y_s^+(k+1)) - v_s^+(k+1)\big).
\end{aligned}$$

Analyzing the stage cost function, we find that

$$\begin{aligned}
&\ell\big(\hat{x}^+(j|k+1) - x_s^+(k+1), v^+(j|k+1) - v_s^+(k+1)\big)\\
&- \ell\big(\hat{x}^0(j+1|k) - x_s^0(k), v^0(j+1|k) - v_s^0(k)\big)\\
&\leq \sigma_{\ell,x}\big(\|[c_{1,j}(w(k)) \ \ldots \ c_{n,j}(w(k))]^\top\|\big).
\end{aligned}$$

Defining $r_j(w) \coloneqq \|[c_{1,j}(w) \ \ldots \ c_{n,j}(w)]^\top\|$, the last inequality becomes

$$\begin{aligned}
&\ell\big(\hat{x}^+(j|k+1) - x_s^+(k+1), v^+(j|k+1) - v_s^+(k+1)\big)\\
&- \ell\big(\hat{x}^0(j+1|k) - x_s^0(k), v^0(j+1|k) - v_s^0(k)\big)\\
&\leq \sigma_{\ell,x} \circ r_j(w(k)).
\end{aligned}$$

Moreover, analyzing the terminal cost we get that

$$\begin{aligned}
V_f\big(\hat{x}^+(N_p-1|k+1) - x_s^+(k+1), y_s^+(k+1)\big) - V_f\big(\hat{x}^0(N_p|k) - x_s^0(k), y_s^0(k)\big)&\\
\leq \sigma_f \circ r_{N_p-1}(w(k)).&
\end{aligned}$$

Due to the choice $y_s^+(k+1) = y_s^0(k)$, which implies $x_s^+(k+1) = x_s^0(k)$, and given that

$$
\begin{aligned}
&V_f\big(\hat{x}^+(N_p|k+1) - x_s^+(k+1), y_s^+(k+1)\big) - V_f\big(\hat{x}^0(N_p|k) - x_s^0(k), y_s^0(k)\big) \\
&= V_f\big(\hat{x}^+(N_p|k+1) - x_s^+(k+1), y_s^+(k+1)\big) \\
&- V_f\big(\hat{x}^+(N_p-1|k+1) - x_s^+(k+1), y_s^+(k+1)\big) \\
&+ V_f\big(\hat{x}^+(N_p-1|k+1) - x_s^+(k+1), y_s^+(k+1)\big) \\
&- V_f\big(\hat{x}^0(N_p|k) - x_s^0(k), y_s^0(k)\big),
\end{aligned}
$$

we get

$$
\begin{aligned}
\Delta W \coloneqq W(x(k+1), y_t) - W(x(k), y_t) \leq \sum_{j=0}^{N_p-2} \sigma_{\ell,x} \circ r_j(w(k)) + \sigma_f \circ r_{N_p-1}(w(k)) \\
- \ell\big(x(k) - x_s^0(k), v^0(0|k) - v_s^0(k)\big) \\
+ \ell\big(\hat{x}^+(N_p-1|k+1) - x_s^+(k+1), v_f(y_s^+(k+1)) - v_s^+(k+1)\big) \\
+ V_f\big(\hat{x}^+(N_p|k+1) - x_s^+(k+1), y_s^+(k+1)\big) \\
- V_f\big(\hat{x}^+(N_p-1|k+1) - x_s^0 + (k+1), y_s^+(k+1)\big).
\end{aligned}
$$

Thanks to Assumption 2.7, we have that

$$
\begin{aligned}
&V_f(\hat{x}^+(N_p|k+1) - x_s^+(k+1), y_s^+(k+1)) \\
&- V_f(\hat{x}^+(N_p-1|k+1) - x_s^+(k+1), y_s^+(k+1)) \\
&\leq -\ell\big(\hat{x}^+(N_p-1|k+1) - x_s^+(k+1), v_f(y_s^+(k+1)) - v_s^+(k+1)\big)
\end{aligned}
$$

in $\Omega(y_s^+(k+1))$, and hence, given that the functions $r_j(w)$ are vector norms, there exists a $\mathcal{K}$ function $\lambda_1(\cdot)$ such that

$$
\Delta W \leq -\ell\big(x(k) - x_s^0(k), v^0(0|k) - v_s^0(k)\big) + \lambda_1(\|w(k)\|).
$$

Given that $\ell(x, v) \geq \alpha_\ell(\|x\|)$, the last finding results in

$$
\Delta W \leq -\alpha_\ell(\|x(k) - x_s^0(k)\|) + \lambda_1(\|w(k)\|). \tag{2.18}
$$

To prove the convergence of the system output to the steady output y_s^*, we first need to prove input-to-state stability of the equilibrium x_s^*. In order for this to be true, $W(\cdot, y_t)$ has to be an ISS-Lyapunov function with respect to this equilibrium point.

We start by looking for a $\mathcal{K}_\infty$ function that bounds $W(x(k), y_t)$ from below. From the definition of the function $W(\cdot, y_t)$, and due to Assumptions 2.8 and 2.9, we have that

$$
\begin{aligned}
W(x(k), y_t) &\geq \ell\big(x(k) - x_s^0(k), v^0(0|k) - v_s^0(k)\big) + V_O(y_s^0(k) - y_t) - V_O(y_s^* - y_t) \\
&\geq \alpha_\ell(\|x(k) - x_s^0(k)\|) + \alpha_O(\|y_s^0(k) - y_s^*\|).
\end{aligned}
$$

Because of the Lipschitz continuity of $g_x(\cdot)$, for which $\|x_s^0(k) - x_s^*\| \leq L_g \|y_s^0(k) - y_s^*\|$, we have that

$$W(x(k), y_t) \geq \alpha_\ell(\|x(k) - x_s^0(k)\|) + \alpha_O \left(\frac{1}{L_g} \|x_s^0(k) - x_s^*\| \right).$$

Provided that $\alpha_\ell(\cdot,\cdot)$ and $\alpha_O(\cdot)$ are $\mathcal{K}_\infty$ functions, the function $\alpha_{\ell O}(\cdot) \coloneqq \min\{\alpha_\ell(\cdot,\cdot), \alpha_O(\cdot)\}$ is also a $\mathcal{K}_\infty$ function and, thanks to standard properties of $\mathcal{K}_\infty$ functions, allows us to prove that

$$\begin{aligned} W(x(k), y_t) &\geq \alpha_{\ell O}(\|x(k) - x_s^0(k)\|) + \alpha_{\ell O} \left(\frac{1}{L_g} \|x_s^0(k) - x_s^*\| \right) \\ &\geq \alpha_{\ell O} \left(\frac{1}{2} \|x(k) - x_s^0(k)\| + \frac{1}{2L_g} \|x_s^0(k) - x_s^*\| \right) \\ &\geq \alpha_{\ell O} \left(\min\left\{ \frac{1}{2}, \frac{1}{2L_g} \right\} \|x(k) - x_s^*\| \right). \end{aligned}$$

Defining $\underline{\alpha}(\cdot)$ as a $\mathcal{K}_\infty$ function such that $\underline{\alpha}(\|x\|) \leq \alpha_{\ell O}\left(\min\left\{\frac{1}{2}, \frac{1}{2L_g}\right\}\|x\|\right)$ for all $x \in \mathbb{R}^n$, we finally get that

$$W(x(k), y_t) \geq \underline{\alpha}(\|x(k) - x_s^*\|). \tag{2.19}$$

The upper bound of $W(x(k), y_t)$ can be obtained thanks to Assumption 2.7, which implies $V_f(x(k) - x_s^*, y_s^*) \leq \alpha_f(\|x(k) - x_s^*\|)$ in $\Omega(y_s^*)$. This, together with $\mathcal{Z}$ being a closed set, implies that there exists a $\mathcal{K}_\infty$ function $\bar{\alpha}(\cdot)$ such that

$$W(x(k), y_t) \leq \bar{\alpha}(\|x(k) - x_s^*\|) \tag{2.20}$$

for all $x(k) \in \mathcal{X}_{N_c,N_p}$.

In order to prove the nominal descent property of $W(\cdot, y_t)$ with respect to x_s^*, which is needed to prove that $W(\cdot, y_t)$ is an ISS-Lyapunov function, we analyze two different cases. The logic behind the two cases is the following, and is inspired by [74]: first, we want to prove that, if the tracking error is big, the nominal descent property of $W(\cdot, y_t)$ with respect to $x_s^0(k)$ implies that $W(\cdot, y_t)$ nominally decreases also with respect to x_s^*; then, we want to prove that, if at instant k the tracking error is small and $y_s^0(k) \neq y_s^*$, namely the closed-loop system state is close to the equilibrium $x_s^0(k)$, at time instant $k+1$ it is always possible to find a feasible candidate solution $(\tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1))$ such that $V_O(\tilde{y}_s(k+1) - y_t) < V_O(y_s^0(k) - y_t)$, and the decrease in $V_O(\cdot)$ also ensures the nominal descent of $W(\cdot, y_t)$ with respect to x_s^*.

The threshold that is used to formally discern the two cases is the following: given a constant $\gamma > 0$, which will be specified later, having a large tracking error means that $\alpha_\ell(\|x(k) - x_s^0(k)\|) \geq \gamma\alpha_O(\|y_s^0(k) - y_s^*\|)$ (case 1), while having a small tracking error results in $\alpha_\ell(\|x(k) - x_s^0(k)\|) \leq \gamma\alpha_O(\|y_s^0(k) - y_s^*\|)$ (case 2). The two cases are analyzed below.

Case 1: Provided that

$$\alpha_\ell(\|x(k) - x_s^0(k)\|) \geq \gamma\alpha_O(\|y_s^0(k) - y_s^*\|), \tag{2.21}$$

(2.18) and the Lipschitz continuity of $g_x(\cdot)$ imply

$$\begin{aligned}
\Delta W &\leq \lambda_1(\|w(k)\|) - \alpha_\ell(\|x(k) - x_s^0(k)\|) \\
&= \lambda_1(\|w(k)\|) - \frac{1}{2}\alpha_\ell(\|x(k) - x_s^0(k)\|) - \frac{1}{2}\alpha_\ell(\|x(k) - x_s^0(k)\|) \\
&\leq \lambda_1(\|w(k)\|) - \frac{1}{2}\Big[\alpha_\ell(\|x(k) - x_s^0(k)\|) + \gamma\alpha_O(\|y_s^0(k) - y_s^*\|)\Big] \\
&\leq \lambda_1(\|w(k)\|) - \frac{1}{2}\Big[\alpha_\ell(\|x(k) - x_s^0(k)\|) + \gamma\alpha_O\left(\frac{1}{L_g}\|x_s^0(k) - x_s^*\|\right)\Big] \\
&\leq \lambda_1(\|w(k)\|) - \frac{1}{2}\min\{1,\gamma\}\Big[\alpha_\ell(\|x(k) - x_s^0(k)\|) + \alpha_O\left(\frac{1}{L_g}\|x_s^0(k) - x_s^*\|\right)\Big].
\end{aligned}$$

Using the functions $\alpha_{\ell O}(\cdot)$ and $\underline{\alpha}(\cdot)$ defined earlier, we get

$$\begin{aligned}
\Delta W &\leq \lambda_1(\|w(k)\|) - \frac{1}{2}\min\{1,\gamma\}\Big[\alpha_{\ell O}(\|x(k) - x_s^0(k)\|) + \alpha_{\ell O}\left(\frac{1}{L_g}\|x_s^0(k) - x_s^*\|\right)\Big] \\
&\leq \lambda_1(\|w(k)\|) - \frac{1}{2}\min\{1,\gamma\}\alpha_{\ell O}\left(\frac{1}{2}\|x(k) - x_s^0(k)\| + \frac{1}{2L_g}\|x_s^0(k) - x_s^*\|\right) \\
&\leq \lambda_1(\|w(k)\|) - \frac{1}{2}\min\{1,\gamma\}\cdot\alpha_{\ell O}\left(\min\left\{\frac{1}{2},\frac{1}{2L_g}\right\}\|x(k) - x_s^*\|\right) \\
&\leq \lambda_1(\|w(k)\|) - \frac{1}{2}\min\{1,\gamma\}\underline{\alpha}(\|x(k) - x_s^*\|).
\end{aligned} \tag{2.22}$$

Case 2: Assume

$$\alpha_\ell(\|x(k) - x_s^0(k)\|) \leq \gamma\alpha_O(\|y_s^0(k) - y_s^*\|). \tag{2.23}$$

The proof of the nominal descent of $W(\cdot, y_t)$ in this case is divided into five steps:

1. We first prove that, by choosing a small enough γ, (2.23) implies that $x(k) \in \mathcal{X}_f(y_s^0(k))$; in particular, $\alpha_\ell(\|x(k) - x_s^0(k)\|) \leq \epsilon_x$, where $\epsilon_x > 0$ is a real constant, whose meaning will be clarified later.

2. We prove that, given a state $x(k)$ such that $\alpha_\ell(\|x(k) - x_s^0(k)\|) \leq \epsilon_x$, the following nominal state $\hat{x}^0(1|k)$ is such that $\alpha_\ell(\|\hat{x}^0(1|k) - x_s^0(k)\|) \leq \epsilon_x/2$.

3. We prove that a candidate artificial reference $\tilde{y}_s(k+1)$ such that $V_O(\tilde{y}_s(k+1) - y_t) < V_O(y_s^0(k) - y_t)$ would be feasible for the following nominal state $\hat{x}^0(1|k)$.

4. We prove that the candidate artificial reference $\tilde{y}_s(k+1)$ makes the function $W(\cdot, y_t)$ decrease in the perturbation-free case.

5. We prove that the gap $W(x(k+1)) - W(\hat{x}^0(1|k))$, where $x(k+1)$ is the uncertain following state, is bounded from above by a $\mathcal{K}$ function $\lambda_2 : \mathcal{W} \to \mathbb{R}_{\geq 0}$, which allows us to state that $W(\cdot, y_t)$ is an ISS-Lyapunov function.

Let us first discuss the role of the constant ϵ_x. Assuming to have terminal regions $\mathcal{X}_f(y_s)$ such that $\{x_s\} \subset \mathcal{X}_f(y_s)$, namely the sets $\mathcal{X}_f(y_s)$ contain more than one point, there surely exists a constant $\epsilon_x > 0$ such that the condition $\alpha_\ell(\|x - x_s\|) \leq \epsilon_x$ implies $x \in \mathcal{X}_f(y_s)$, for all $y_s \in \mathcal{Y}_t$.
Moreover, provided that $\alpha_\ell(\|x - x_s\|) \leq \epsilon_x$ implies $x \in \mathcal{X}_f(y_s)$, we also have that, for all x that verify this condition, the couple $\big(\mathbf{v}, y_s\big)$, with $\mathbf{v} := \{v_f(y_s), \ldots, v_f(y_s)\}$, is a feasible solution and the next inequality holds true:

$$\begin{aligned} V^0_{N_c,N_p}(x, y_t) &\leq V_f(x - x_s, y_s) + V_O(y_s - y_t) \\ &\leq \alpha_f(\|x - x_s\|) + V_O(y_s - y_t). \end{aligned} \tag{2.24}$$

Provided that the set $\mathcal{Y}_t$ is bounded, there exists a real constant $\alpha_{O,\max} > 0$ such that $\alpha_O(\|y_1 - y_2\|) \leq \alpha_{O,\max}$ for any $y_1, y_2 \in \mathcal{Y}_t$. Therefore, by choosing $\gamma \leq \gamma_1 := \frac{\epsilon_x}{\alpha_{O,\max}}$, we get that

$$\alpha_\ell(\|x(k) - x^0_s(k)\|) \leq \gamma\alpha_{O,\max} \leq \epsilon_x.$$

Now, given the optimal solution $(\mathbf{v}^0(k), y^0_s(k))$, we use the previously defined notation $\hat{x}^0(j|k) = \phi_\pi(j; y^0_s(k), x(k), \mathbf{v}^0(k), \mathbf{0})$ and focus on the following nominal state $\hat{x}^0(1|k)$. Specifically, we note that (2.23) and (2.24) imply

$$\begin{aligned} \alpha_\ell(\|\hat{x}^0(1|k) - x^0_s(k)\|) &\leq \ell\big(\hat{x}^0(1|k) - x^0_s(k), v^0(1|k) - v^0_s(k)\big) \\ &\leq \alpha_f(\|x(k) - x^0_s(k)\|). \end{aligned}$$

Since $\alpha_\ell(\|x(k) - x^0_s(k)\|) \leq \gamma\alpha_{O,\max}$ and $\alpha_\ell(\cdot)$ is a $\mathcal{K}_\infty$ function, it also holds true that $\|x(k) - x^0_s(k)\| \leq \alpha_\ell^{-1}(\gamma\alpha_{O,\max})$, which implies that

$$\begin{aligned} \alpha_\ell(\|\hat{x}^0(1|k) - x^0_s(k)\|) &\leq \alpha_f(\|x(k) - x^0_s(k)\|) \\ &\leq \alpha_f \circ \alpha_\ell^{-1}(\gamma\alpha_{O,\max}). \end{aligned}$$

For $\gamma \leq \gamma_2 := \frac{\alpha_\ell \circ \alpha_f^{-1}(\epsilon_x/2)}{\alpha_{O,\max}}$, we have that $\alpha_\ell(\|\hat{x}^0(1|k) - x^0_s(k)\|) \leq \epsilon_x/2$. This means that the nominal state $\hat{x}^0(1|k)$ is inside the interior of $\big\{x : \alpha_\ell(\|x - x^0_s\|) \leq \epsilon_x\big\}$, which would make the problem $P_{N_c,N_p}(\hat{x}^0(1|k), y_t)$ feasible, as $\hat{x}^0(1|k) \in \mathcal{X}_f(y^0_s(k))$ and the terminal control input $v_f(y^0_s(k))$ is feasible.
We now look for a better candidate solution. In detail, we seek a candidate $\tilde{y}_s(k+1)$ such that $V_O(\tilde{y}_s(k+1) - y_t) < V_O(y^+_s(k+1) - y_t)$, recalling that $y^+_s(k+1) \equiv y^0_s(k)$.

Exploiting the convexity of $V_O(\cdot)$, there exists a new candidate $\tilde{y}_s(k+1) \in \mathcal{Y}_t$ such that

$$\tilde{y}_s(k+1) \coloneqq \beta y_s^0(k) + (1-\beta) y_s^*, \quad \beta \in [0,1]. \tag{2.25}$$

Equation (2.25) can be rewritten as

$$\tilde{y}_s(k+1) - y_s^0(k) = (1-\beta)(y_s^* - y_s^0(k)).$$

Due to this new candidate solution, the offset cost satisfies

$$\begin{aligned} V_O(\tilde{y}_s(k+1) - y_t) - V_O(y_s^0(k) - y_t) = V_O\big(\beta(y_s^0(k) - y_t) + (1-\beta)(y_s^* - y_t)\big) \\ - V_O(y_s^0(k) - y_t). \end{aligned} \tag{2.26}$$

Given the convexity of $V_O(\cdot)$, (2.26) becomes

$$\begin{aligned} V_O(\tilde{y}_s(k+1) - y_t) - V_O(y_s^0(k) - y_t) &\leq \beta V_O\big(y_s^0(k) - y_t\big) + (1-\beta) V_O\big(y_s^* - y_t\big) \\ &\quad - V_O(y_s^0(k) - y_t) \\ &= -(1-\beta)\Big[V_O\big(y_s^0(k) - y_t\big) - V_O\big(y_s^* - y_t\big)\Big] \\ &\leq -(1-\beta)\alpha_O(\|y_s^0(k) - y_s^*\|), \end{aligned} \tag{2.27}$$

which, for $\beta \in [0,1)$, proves that $\tilde{y}_s(k+1)$ is such that $V_O(\tilde{y}_s(k+1) - y_t) < V_O(y_s^0(k) - y_t)$. We now need to find a lower bound β_1 such that, for all $\beta \geq \beta_1$, the new candidate $\tilde{y}_s(k+1)$ ensures also that $\alpha_\ell(\|\hat{x}^0(1|k) - \tilde{x}_s(k+1)\|) \leq \epsilon_x$. In order to find β_1, we start by stating that, because of the Lipschitz continuity of $g_x(\cdot)$,

$$\begin{aligned} \alpha_\ell\big(\|\hat{x}^0(1|k) - \tilde{x}_s(k+1)\|\big) &\leq \alpha_\ell\big(\|\hat{x}^0(1|k) - x_s^0(k)\| + \|x_s^0(k) - \tilde{x}_s(k+1)\|\big) \\ &\leq \alpha_\ell\big(\|\hat{x}^0(1|k) - x_s^0(k)\| + L_g\|y_s^0(k) - \tilde{y}_s(k+1)\|\big) \\ &= \alpha_\ell\big(\|\hat{x}^0(1|k) - x_s^0(k)\| + L_g(1-\beta)\|y_s^0(k) - y_s^*\|\big) \\ &\leq \alpha_\ell\big(\alpha_\ell^{-1}(\epsilon_x/2) + L_g(1-\beta)\|y_s^0(k) - y_s^*\|\big). \end{aligned} \tag{2.28}$$

Recall that $\alpha_O(\|y_s^0(k) - y_s^*\|) \leq \alpha_{O,\max}$, which implies $\|y_s^0(k) - y_s^*\| \leq \alpha_O^{-1}(\alpha_{O,\max})$. For this reason, (2.28) can be rewritten as

$$\alpha_\ell\big(\|\hat{x}^0(1|k) - \tilde{x}_s(k+1)\|\big) \leq \alpha_\ell\big(\alpha_\ell^{-1}(\epsilon_x/2) + L_g(1-\beta)\alpha_O^{-1}(\alpha_{O,\max})\big).$$

Therefore, for $\beta \geq \beta_1 \coloneqq 1 - \frac{\alpha_\ell^{-1}(\epsilon_x) - \alpha_\ell^{-1}(\epsilon_x/2)}{L_g \cdot \alpha_O^{-1}(\alpha_{O,\max})}$, $\alpha_\ell(\|\hat{x}^0(1|k) - \tilde{x}_s(k+1)\|) \leq \epsilon_x$, which ensures that:

1. $\hat{x}^0(1|k) \in \mathcal{X}_f(\tilde{y}_s(k+1))$, guaranteeing that there exists a candidate control sequence $\tilde{\mathbf{v}}(k+1)$ such that the couple $(\tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1))$ is a feasible solution to $P_{N_c,N_p}(\hat{x}^0(1|k), y_t)$; indeed, $\tilde{\mathbf{v}}(k+1) \coloneqq \{v_f(\tilde{y}_s(k+1)), \ldots, v_f(\tilde{y}_s(k+1))\}$ is a feasible control sequence.

2. The optimal control sequence $\tilde{\mathbf{v}}(k+1)$ is such that

$$V^0_{N_c,N_p}(\hat{x}^0(1|k), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1)) - V_O(\tilde{y}_s(k+1) - y_t) \\ \leq \alpha_f(\|\hat{x}^0(1|k) - \tilde{x}_s(k+1)\|).$$

Due to $\alpha_\ell(\|\hat{x}^0(1|k) - \tilde{x}_s(k+1)\|) \leq \epsilon_x$ and to the Lipschitz continuity of $g_x(\cdot)$, we have that

$$\begin{aligned} V_{N_c,N_p}&(\hat{x}^0(1|k), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1)) - V_O(\tilde{y}_s - y_t) \\ &\leq \alpha_f(\|\hat{x}^0(1|k) - \tilde{x}_s(k+1)\|) \\ &\leq \alpha_f\big(\|\hat{x}^0(1|k) - x^0_s(k)\| + \|x^0_s(k) - \tilde{x}_s(k+1)\|\big) \\ &\leq \alpha_f\big(\|\hat{x}^0(1|k) - x^0_s(k)\| + L_g\|y^0_s(k) - \tilde{y}_s(k+1)\|\big) \\ &= \alpha_f\big(\|\hat{x}^0(1|k) - x^0_s(k)\| + L_g(1-\beta)\|y^0_s(k) - y^*_s\|\big). \end{aligned} \tag{2.29}$$

To analyze $W(x(k+1), y_t) - W(x(k), y_t) = V^0_{N_c,N_p}(x(k+1), y_t) - V^0_{N_c,N_p}(x(k), y_t)$, where $x(k+1)$ is the following uncertain state, we note that

$$\begin{aligned} &V^0_{N_c,N_p}(x(k+1), y_t) - V^0_{N_c,N_p}(x(k), y_t) \\ &\leq V_{N_c,N_p}(x(k+1), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1)) - V^0_{N_c,N_p}(x(k), y_t) \\ &\leq V_{N_c,N_p}(x(k+1), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1)) - V_{N_c,N_p}(\hat{x}^0(1|k), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1)) \\ &+ V_{N_c,N_p}(\hat{x}^0(1|k), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1)) - V^0_{N_c,N_p}(x(k), y_t). \end{aligned}$$

We analyze this difference in two steps: (i) first, we analyze the nominal descent of $W(\cdot, y_t)$, which is provided by the term $V_{N_c,N_p}(\hat{x}^0(1|k), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1)) - V^0_{N_c,N_p}(x(k), y_t)$, (ii) then we analyze the contribution of the perturbation $w(k)$, which is provided by $V_{N_c,N_p}(x(k+1), y_t; \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1)) - V_{N_c,N_p}(\hat{x}^0(1|k), y_t; \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1))$.

Analyzing $V_{N_c,N_p}(\hat{x}^0(1|k), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1)) - V^0_{N_c,N_p}(x(k), y_t)$:

$$\begin{aligned} &V_{N_c,N_p}(\hat{x}^0(1|k), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1)) - V^0_{N_c,N_p}(x(k), y_t) \\ &\leq \alpha_f(\|\hat{x}^0(1|k) - \tilde{x}_s(k+1)\|) - \alpha_\ell(\|x(k) - x^0_s(k)\|) \\ &+ V_O(\tilde{y}_s(k+1) - y_t) - V_O(y^0_s(k) - y_t). \end{aligned} \tag{2.30}$$

Equations (2.27) and (2.29) imply that (2.30) becomes

$$\begin{aligned} &V_{N_c,N_p}(\hat{x}^0(1|k), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1)) - V^0_{N_c,N_p}(x(k), y_t) \\ &\leq \alpha_f\big(\|\hat{x}^0(1|k) - x^0_s(k)\| + L_g(1-\beta)\|y^0_s(k) - y^*_s\|\big) \\ &- \alpha_\ell(\|x(k) - x^0_s(k)\|) - (1-\beta)\alpha_O(\|y^0_s(k) - y^*_s\|) \\ &\leq \alpha_f\big(2\|\hat{x}^0(1|k) - x^0_s(k)\|\big) + \alpha_f\big(2L_g(1-\beta)\|y^0_s(k) - y^*_s\|\big) \\ &- (1-\beta)\alpha_O(\|y^0_s(k) - y^*_s\|) - \alpha_\ell(\|x(k) - x^0_s(k)\|). \end{aligned} \tag{2.31}$$

Recalling that $\alpha_f(\|x\|) = \bar{\alpha}_f \cdot \|x\|^2$, (2.31) rewrites as

$$\begin{aligned} V_{N_c,N_p}(\hat{x}^0(1|k), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1)) - V^0_{N_c,N_p}(x(k), y_t) \leq\ & 4\bar{\alpha}_f \|\hat{x}^0(1|k) - x^0_s(k)\|^2 \\ & + 4\bar{\alpha}_f \cdot L_g^2 (1-\beta)^2 \|y^0_s(k) - y^*_s\|^2 \\ & - (1-\beta)\alpha_O(\|y^0_s(k) - y^*_s\|) \\ & - \alpha_\ell(\|x(k) - x^0_s(k)\|). \end{aligned} \tag{2.32}$$

Defining the function $g : \mathbb{R} \times \mathcal{S} \to \mathbb{R}$ as

$$g(\beta, s) \coloneqq (1-\beta)\alpha_O(s) - 4\bar{\alpha}_f L_g^2 (1-\beta)^2 s^2,$$

such that (2.32) becomes

$$\begin{aligned} & V_{N_c,N_p}(\hat{x}^0(1|k), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1)) - V^0_{N_c,N_p}(x(k), y_t) \\ & \leq 4\bar{\alpha}_f \|\hat{x}^0(1|k) - x^0_s(k)\|^2 - g(\beta, \|y^0_s(k) - y^*_s\|) - \alpha_\ell(\|x(k) - x^0_s(k)\|). \end{aligned} \tag{2.33}$$

In order to prove the nominal descent property of $W(\cdot, y_t)$, we must ensure that there exists a $\mathcal{K}_\infty$ function $\alpha_\beta(\cdot)$ such that $g(\beta, \|y^0_s(k) - y^*_s\|) \geq \alpha_\beta(\|y^0_s(k) - y^*_s\|)$. Thanks to Assumption 2.10, in order for these conditions to hold true, it is sufficient to choose $\beta \in (\beta_2, 1)$, where $\beta_2 \coloneqq 1 - \min\{\mathfrak{b}_1, \mathfrak{b}_2\}$. Since both $\mathfrak{b}_1$ and $\mathfrak{b}_2$ are assumed to be positive, the interval $(\beta_2, 1)$ is guaranteed to be non-empty. This is shown at the end of the current proof.

Given the function $\alpha_\beta(\cdot)$, (2.33) can be rewritten as

$$\begin{aligned} & V_{N_c,N_p}(\hat{x}^0(1|k), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1)) - V^0_{N_c,N_p}(x(k), y_t) \\ & \leq 4\bar{\alpha}_f \|\hat{x}^0(1|k) - x^0_s(k)\|^2 - \alpha_\beta(\|y^0_s(k) - y^*_s\|) - \alpha_\ell(\|x(k) - x^0_s(k)\|) \\ & \leq 4\bar{\alpha}_f \|\hat{x}^0(1|k) - x^0_s(k)\|^2 - \frac{1}{2}\alpha_\beta(\|y^0_s(k) - y^*_s\|) \\ & - \frac{1}{2}\alpha_\beta(\|y^0_s(k) - y^*_s\|) - \alpha_\ell(\|x(k) - x^0_s(k)\|). \end{aligned} \tag{2.34}$$

We now analyze (2.34) in two steps:

1. we first prove that it is always possible to choose $\gamma \leq \gamma_3$, where γ_3 will be defined later, such that $4\bar{\alpha}_f \|\hat{x}^0(1|k) - x^0_s(k)\|^2 - \frac{1}{2}\alpha_\beta(\|y^0_s(k) - y^*_s\|) \leq 0$;

2. we then prove that the term $\frac{1}{2}\alpha_\beta(\|y^0_s(k) - y^*_s\|) + \alpha_\ell(\|x(k) - x^0_s(k)\|)$ can be lower-bounded by a $\mathcal{K}_\infty$ function $\alpha_2(\|x(k) - x^*_s\|)$, which is then used to prove that $W(\cdot, y_t)$ is an ISS-Lyapunov function.

Starting from the first point, recall that $\alpha_\ell(\|\hat{x}^0(1|k) - x^0_s(k)\|) \leq \bar{\alpha}_f \|x(k) - x^0_s(k)\|^2$ and also that $\alpha_\ell(\|x(k) - x^0_s(k)\|) \leq \gamma\alpha_O(\|y^0_s(k) - y^*_s\|)$. Due to these inequalities, we get

$$\|\hat{x}^0(1|k) - x^0_s(k)\| \leq \alpha_\ell^{-1}\left(\bar{\alpha}_f \cdot \left[\alpha_\ell^{-1}\left(\gamma\alpha_O(\|y^0_s(k) - y^*_s\|)\right)\right]^2\right),$$

and, as a consequence, $4\bar{\alpha}_f\|\hat{x}^0(1|k) - x_s^0(k)\|^2 - \frac{1}{2}\alpha_\beta(\|y_s^0(k) - y_s^*\|) \leq 0$ for $\gamma \leq \gamma_3$, where γ_3 is defined as

$$\gamma_3 := \frac{\alpha_\ell\left(\sqrt{\frac{1}{\bar{\alpha}_f}\cdot\alpha_\ell\left(\sqrt{\frac{1}{8\bar{\alpha}_f}\alpha_\beta(\|y_s^0(k) - y_s^*\|)}\right)}\right)}{\alpha_O(\|y_s^0(k) - y_s^*\|)}.$$

Consequently, for $\gamma \leq \gamma_3$, and considering the Lipschitz continuity of $g_x(\cdot)$, (2.34) can be rewritten as

$$\begin{aligned}
&V_{N_c,N_p}(\hat{x}^0(1|k), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1)) - V^0_{N_c,N_p}(x(k), y_t)\\
&\leq -\frac{1}{2}\alpha_\beta(\|y_s^0(k) - y_s^*\|) - \alpha_\ell(\|x(k) - x_s^0(k)\|)\\
&\leq -\frac{1}{2}\alpha_\beta(\frac{1}{L_g}\|x_s^0(k) - x_s^*\|) - \alpha_\ell(\|x(k) - x_s^0(k)\|).
\end{aligned} \tag{2.35}$$

We now exploit a $\mathcal{K}_\infty$ function

$$\alpha_{\beta\ell}(\|x\|) := \min\left\{\frac{1}{2}\alpha_\beta\left(\frac{1}{L_g}\|x\|\right), \alpha_\ell(\|x\|)\right\}$$

to further develop (2.35) as

$$\begin{aligned}
&V_{N_c,N_p}(\hat{x}^0(1|k), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1)) - V^0_{N_c,N_p}(x(k), y_t)\\
&\leq -\alpha_{\beta\ell}(\|x_s^0(k) - x_s^*\|) - \alpha_{\beta\ell}(\|x(k) - x_s^0(k)\|)\\
&\leq -\alpha_{\beta\ell}\left(\frac{1}{2}\|x_s^0(k) - x_s^*\| + \frac{1}{2}\|x(k) - x_s^0(k)\|\right)\\
&\leq -\alpha_{\beta\ell}\left(\frac{1}{2}\|x(k) - x_s^*\|\right).
\end{aligned}$$

Exploiting a $\mathcal{K}_\infty$ function $\alpha_2(\|x\|) \leq \alpha_{\beta\ell}\left(\frac{1}{2}\|x\|\right)$, we finally obtain

$$V_{N_c,N_p}(\hat{x}^0(1|k), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1)) - V^0_{N_c,N_p}(x(k), y_t) \leq -\alpha_2\left(\|x(k) - x_s^*\|\right), \tag{2.36}$$

which proves the nominal descent of $W(\cdot, y_t)$. We now need to analyze $V_{N_c,N_p}(x(k+1), y_t, \tilde{\mathbf{v}}(k+1),$ $\tilde{y}_s(k+1)) - V_{N_c,N_p}(\hat{x}^0(1|k), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1))$, namely the contribution of the perturbation $w(k)$. First, we prove that the $(\tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1))$ is a feasible solution for $P_{N_c,N_p}(x(k+1), y_t)$. As $\hat{x}^0(1|k) \in \mathcal{X}_f(\tilde{y}_s(k+1))$ and $\tilde{\mathbf{v}}(k+1) = \{v_f(\tilde{y}_s(k+1)), \ldots, v_f(\tilde{y}_s(k+1))\}$, we have that $\phi_\pi(j; \tilde{y}_s(k+1), \hat{x}^0(1|k), \tilde{\mathbf{v}}(k+1), \mathbf{0}) \in \mathcal{X}_f(\tilde{y}_s(k+1))$, for all $j = 1, \ldots, N_p$. Therefore, given the definition of the set sequence $\{\mathcal{F}(j)\}_{j\geq 0}$, we have that $\phi_\pi(j; \tilde{y}_s(k+1), x(k+1), \tilde{\mathbf{v}}(k+1), \mathbf{0}) \in \mathcal{X}_f(\tilde{y}_s(k+1)) \oplus \mathcal{F}(j)$. The monotonicity property of the sequence $\{\mathcal{R}(j)\}_{j\geq 0}$, which is used to build the restricted constraints $\mathcal{Z}_\pi(j)$, guarantees that $(\mathcal{X}_f(\tilde{y}_s(k+1)) \oplus \mathcal{F}(j) \times \{0\}) \subseteq \mathcal{Z}_\pi(j)$, for all $j = 1, \ldots, N_p$. Moreover, given that $\mathcal{X}_f(\tilde{y}_s(k+1)) \oplus \mathcal{F}(N_p - 1) \subseteq \Omega(\tilde{y}_s(k+1))$, it also holds true that $\phi_\pi(N_p; \tilde{y}_s(k+1), x(k+1), \tilde{\mathbf{v}}(k+1), \mathbf{0}) \in \mathcal{X}_f(\tilde{y}_s)$, which proves that the constraints are verified and the last prediction, together with $\tilde{y}_s(k+1)$, is inside the robust invariant set for tracking $\mathcal{T}$.

Analyzing $V_{N_c,N_p}(x(k+1), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1)) - V_{N_c,N_p}(\hat{x}^0(1|k), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1))$, defining $\hat{x}^+(j|k+1) := \phi_\pi(j; \tilde{y}_s(k+1), x(k+1), \tilde{\mathbf{v}}(k+1), \mathbf{0})$ and $\hat{x}_n^+(j|k+1) := \phi_\pi(j; \tilde{y}_s(k+1), \hat{x}^0(1|k), \tilde{\mathbf{v}}(k+1), \mathbf{0})$, then

$$
\begin{aligned}
V_{N_c,N_p}(x(k+1), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1)) - V_{N_c,N_p}(\hat{x}^0(1|k), y_t, \tilde{\mathbf{v}}(k+1), \tilde{y}_s(k+1))& \\
\leq \sum_{j=0}^{N_p-1} \sigma_{\ell,x}\left(\|\hat{x}^+(j|k+1) - \hat{x}_n^+(j|k+1)\|\right)& \\
+ \sigma_f\left(\|\hat{x}^+(N_p|k+1) - \hat{x}_n^+(N_p|k+1)\|\right)& \\
\leq \sum_{j=0}^{N_p-1} \sigma_{\ell,x} \circ r_j(w(k)) + \sigma_f \circ r_{N_p}(w(k)).&
\end{aligned}
$$

Using a $\mathcal{K}$ function $\lambda_2(\|w\|) \geq \sum_{j=0}^{N_p-1} \sigma_{\ell,x} \circ r_j(w) + \sigma_f \circ r_{N_p}(w)$ in $\mathcal{W}$, the last result and (2.36) prove that

$$\Delta W \leq -\alpha_2(\|x(k) - x_s^*\|) + \lambda_2(\|w(k)\|). \tag{2.37}$$

Defining the $\mathcal{K}$ function $\lambda(\|w\|) := \max\{\lambda_1(\|w\|), \lambda_2(\|w\|)\}$ and the $\mathcal{K}_\infty$ function $\alpha(\cdot) := \min\left\{\underline{\alpha}(\cdot), \alpha_2(\cdot)\right\}$, with $\gamma = \min\{\gamma_1, \gamma_2, \gamma_3\}$, (2.22) and (2.37) result in

$$\Delta W \leq \lambda(\|w(k)\|) - \alpha(\|x(k) - x_s^*\|),$$

which proves that $W(\cdot, y_t)$ is an ISS-Lyapunov function with respect to x_s^*, and that x_s^* is input-to-state stable for the closed-loop system.

Input-to-state stability of x_s^* and boundedness of $\mathcal{W}$ imply that, for any $x(0) \in \mathcal{X}_{N_c,N_p}$ and for any perturbation signal $\mathbf{w} \in \mathcal{W}^j$, there exist a $\mathcal{K}$ function $\gamma_a(\cdot)$ and a constant $\delta > 0$ such that

$$\lim_{k\to\infty} \sup \|x(k) - x_s^*\| \leq \gamma_a\left(\lim_{k\to\infty} \|w(k)\|\right) \leq \delta,$$

which proves that the closed-loop system state x converges to the set $x_s^* \oplus \{x : \|x\| \leq \delta\}$. Given the uniform continuity of the control law $\pi(y_s, x, v)$ with respect to (x, v), and the continuity of the output function $h(x, u)$ in all its arguments, this last finding implies that also the system output y converges to a bounded set containing y_s^*, the best admissible steady output, which ends the proof. ■

Remark 2.8

For large values of $w(k)$ it might still happen that the optimal artificial reference at instant k is exactly y_s^* and temporarily changes to a value $y_s^0(k+1) \neq y_s^*$ at instant $k+1$. Given that the feedback policies $\pi(y_s, x, v)$ are not required to be uniformly continuous with respect to y_s, this consideration implies that the set where the system output converges is bounded but might *be non-connected. Choosing control laws $\pi(y_s, x, v)$ that are uniformly continuous also with respect to y_s would ensure that the set where the system output converges is also connected.*

2.6.1 Proof of the existence of the $\mathcal{K}_\infty$ function $\alpha_\beta(\cdot)$

We want to prove that it is always possible to choose a feasible $\beta \in (\max\{\beta_1, \beta_2\}, 1)$ such that the function $g(\beta, s)$ can be lower-bounded by a $\mathcal{K}_\infty$ function $\alpha_\beta(s)$. In order for this to be true, it must hold true that, for $\beta \in (\max\{\beta_1, \beta_2\}, 1)$, $g(\beta, 0) = 0$, $g(\beta, s) > 0$ for all $s > 0$, and $g(\beta, s_1) < g(\beta, s_2)$ if $s_1 < s_2$.

The first condition is easily proved by noting that in $s = 0$, $s^2 = 0$ and $\alpha_O(s) = 0$. Since $0 < \beta < 1$, the second condition rewrites as

$$\alpha_O(s) - (1 - \beta)4\bar{\alpha}_f L_g^2 s_2 > 0, \ \forall s > 0.$$

This condition is verified if β is such that $\beta > 1 - \frac{\alpha_O(s)}{4\bar{\alpha}_f L_g^2 s^2}$. Given that $\beta > 1 - \mathfrak{b}_1$, this condition is verified for all $s \in \mathcal{S} \setminus \{0\}$.

Since $0 < \beta < 1$, the third condition requires to check that, for $s_2 > s_1$,

$$\alpha_O(s_1) - (1 - \beta)4\bar{\alpha}_f L_g^2 s_1^2 - \alpha_O(s_2) + (1 - \beta)4\bar{\alpha}_f L_g^2 s_2^2 < 0,$$

which reduces to

$$\beta > 1 - \frac{\alpha_O(s_2) - \alpha_O(s_1)}{4\bar{\alpha}_f s_2^2 - 4\bar{\alpha}_f s_1^2}.$$

As $\beta > 1 - \mathfrak{b}_2$, also this condition is verified for all $s_1, s_2 \in \mathcal{S}$ such that $s_2 > s_1$. In order to prove that $g(\beta, s)$ is lower-bounded by a $\mathcal{K}_\infty$ function, we must also ensure that $g(\beta, s) \to \infty$ for $s \to \infty$. However, as $\mathcal{S}$ is an interval, an extension of $g(\beta, s)$ can be chosen to grow arbitrarily fast for $s \in \mathbb{R} \setminus \mathcal{S}$. As a consequence, there exists a $\mathcal{K}_\infty$ function $\alpha_\beta(\cdot)$ such that $g(\beta, s) \geq \alpha_\beta(s)$.

2.7 Supporting algorithms for the design of the controller

The computation of all the ingredients of the proposed MPC is not trivial and has to be carried out iteratively. This is because there is a strong interdependence between all the ingredients that does not allow us to build a simple sequence of steps to be followed. Indeed, to give an example, the computation of the terminal ingredients requires to know $\mathcal{F}(N_p - 1)$, but the computation of $\mathcal{F}(N_p - 1)$ requires to know the control laws $\pi(\cdot, \cdot, \cdot)$, that should be designed together with the terminal ingredients, given that $V_f(\cdot, \cdot)$ must behave as a CLF under the control law $\pi(y_s, x, v_f(y_s))$. Therefore, in the following we provide several algorithms that are needed to compute some of the ingredients and give some remarks that are needed to support the designer in the most critical steps. In any case, we suggest to follow this simplified process:

1. Design the pre-stabilizing control laws $\pi(\cdot,\cdot,\cdot)$ and the terminal ingredients using a candidate $\mathcal{Y}_t$ set.

2. Compute the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ related to the designed laws $\pi(\cdot,\cdot,\cdot)$.

3. Based on the found set sequences, choose a prediction horizon N_p verifying all the assumptions and check that the terminal ingredients and the set $\mathcal{Y}_t$ computed at point 1 verify the assumptions, too. If not, go back to point 1.

2.7.1 Computation of the pre-stabilizing control laws and the terminal ingredients

In this subsection, we briefly discuss different approaches that can be employed to design the pre-stabilizing control laws and the terminal ingredients of our robust tracking MPC for nonlinear systems. In particular, we mention four different approaches that require both the stage cost and terminal cost functions to be quadratic and the pre-stabilizing control laws to be linear, then we discuss in detail two of them:

1. If the system and the input constraints are such that feedback linearization techniques can be employed, design $\pi(\cdot,\cdot,\cdot)$ including one term that feedback-linearizes the system and one term that is a simple LQ regulator. Then, design the weight matrix P to be employed in the terminal cost so as to verify the CLF condition.

2. If the nonlinear system under control can be described with an exact quasi-LPV realization based on some scheduling parameters $\xi(t)$, then control laws $\pi(\cdot,\cdot,\cdot)$ and terminal ingredients $\big(V_f(\cdot,\cdot),\mathcal{T}\big)$ verifying the assumptions can be designed following LMI-based methods like those used in [75, 108].

3. Design the pre-stabilizing control laws and the terminal ingredients with LMI methods that resemble those that are exploited in the case of linear systems, as in [79]. This method is briefly introduced later in the text.

4. Locally approximate the nonlinear system to a Linear Time-Varying (LTV) system and design both the pre-stabilizing control laws and the terminal ingredients using concepts from [142, 143]. Also this method is introduced later in the text.

Computation method based on LTI approximation

This method is based on the same LMIs presented in (1.12), therefore we here focus on the steps to be followed to build the needed LMIs.

We suppose to employ a quadratic stage cost function and a quadratic terminal cost function, and assume the set $\mathcal{Y}_t$ to be available. We aim at finding an admissible robust invariant set for tracking $\mathcal{T} = \left(\bigcup_{y_s \in \mathcal{Y}_t} \mathcal{X}_f(y_s)\right) \times \mathcal{Y}_t$.

A process that can be followed is this:

1. Divide the set $\mathcal{Y}_t$ into a sensible number of subsets $\mathcal{Y}_{t,j}$.

2. For each subset $\mathcal{Y}_{t,j}$:

   a) Get a reasonable number of samples $y_s \in \mathcal{Y}_{t,j}$.

   b) Solve a system of LMIs where each LMI is structured as in (1.12) and employs the matrices A and B obtained by linearizing the nominal system model in (x_s, u_s), where $x_s = g_x(y_s)$ and $u_s = g_u(y_s)$. In this way, the feedback policies can be set as $\pi(y_s, x, v) = K(\mathcal{Y}_{t,j}) \cdot (x - x_s) + v$ and a weight matrix $P(\mathcal{Y}_{t,j})$ satisfying (1.9) $\forall y_s \in \mathcal{Y}_{t,j}$ is obtained. Consequently, due to the choice of such linear controllers, $v_f(y_s) = 0$, $\forall y_s \in \mathcal{Y}_{t,j}$.

   c) Find the values $\omega(y_s) > 0$ such that the sets $\Omega(y_s) = \{x : V_f(x - x_s, y_s) \leq \omega(y_s)\}$ verify (1.10) and $\Omega(y_s) \times \{v_f(y_s)\} \in \mathcal{Z}_\pi(N_p - 1)$.

   d) Find the largest values $\rho(y_s) > 0$ such that the sets $\mathcal{X}_f(y_s) = \{x : V_f(x - x_s, y_s) \leq \rho(y_s)\}$ meet the conditions $\mathcal{X}_f(y_s) \subseteq \Omega(y_s) \ominus \mathcal{F}(N_p - 1)$ and $\mathcal{X}_f(y_s) \times \{v_f(y_s)\} \in \mathcal{Z}_\pi(N_p)$.

   e) Check that for all $x \in \Omega(y_s)$, $f(x, \pi(y_s, x, v_f(y_s), 0)) \in \mathcal{X}_f(y_s)$.

Remark 2.9

Finding control laws $\pi(y_s, x, v) = K(\mathcal{Y}_{t,j}) \cdot (x - x_s) + v$ that make the sets $\Omega(y_s)$ sufficiently contracting, namely that are able to bring any state $x \in \Omega(y_s)$ to $\mathcal{X}_f(y_s)$ in one step, is challenging. This is particularly true when the set $\mathcal{F}(N_p - 1)$ is large. Therefore, the theoretical results of this chapter are sound, but are employable in practice only when the perturbation set $\mathcal{W}$ is small.

Computation method based on LTV approximation

The method proposed next for the computation of the control laws $\pi(y_s, x, v)$ and the terminal ingredients of the robust MPC for tracking is a slightly modified version of the method presented in [90, Appendix B], and based on [142, 143], for the computation of the terminal ingredients of a non-robust MPC for tracking. In particular, we consider the case of quadratic stage cost and terminal cost functions, for the sake of simplicity.

The basic idea behind the method proposed in [90, Appendix B] can be summed up in the following steps:

- The chosen set of admissible setpoints $\mathcal{Y}_t$ is divided into smaller subsets $\mathcal{Y}_{s_i}$.
- Given n_i samples $y_{s_j} \in \mathcal{Y}_{s_i}$, within the regions given by (x_{s_j}, v_{s_j}) such that $x_{s_j} = g_x(y_{s_j})$ and $v_{s_j} = g_v(y_{s_j})$, for all $j \in [1, n_i]$, the nonlinear system is approximated as an LTV system, i.e.

$$\begin{aligned} f(x,u) &= f(x_s(y_s), u_s(y_s)) \\ &\quad + \sum_{j=1}^{n_i} \lambda_j \left[A_{i,j}(x - g_x(y_s)) + B_{i,j}(u - g_u(y_s)) \right], \end{aligned}$$

  where $[A_{i,j}, B_{i,j}] \in \{[A_{i,1}, B_{i,1}], \ldots, [A_{i,n_i}, B_{i,n_i}]\}$, $\lambda_j \in [0,1]$, and $\sum_{j=1}^{n_i} \lambda_j = 1$.
- Matrices K_i and P_i verifying the Lyapunov inequality

$$A_{K,i,j}^\top P_i A_{K,i,j} - P_i \leq -Q - K_i^\top R K_i, \quad \forall j,$$

  where $A_{K,i,j} = A_{i,j} + B_{i,j} K_i$, are found by solving LMIs.
- Invariant sets Ω_i based on the control laws $\kappa(x, y_s) = K_i(x - g_x(y_s)) + g_v(y_s)$ are built for each subset $\mathcal{Y}_{s_i}$.
- An invariant set for tracking is obtained as

$$\Gamma = \bigcup_i \Gamma_i, \quad \Gamma_i = \left\{ (x, y_s) : x \in g_x(y_s) \oplus \Omega_i, y_s \in \mathcal{Y}_{s_i} \right\}.$$

In the robust context of the current work, two modifications to this method are made:

1. We set the pre-stabilizing control laws as $\pi(y_s, x, v) = K_i(x - g_x(y_s)) + v$, $\forall y_s \in \mathcal{Y}_{s_i}$ and employ $v_f(y_s) = g_v(y_s)$ as terminal control inputs.
2. Given that we need to find regions $\Omega(y_s)$ and $\mathcal{X}_f(y_s)$ such that $f\big(\hat{x}, \pi(y_s, \hat{x}, v_f(y_s)), 0\big) \in \mathcal{X}_f(y_s)$, for all $\hat{x} \in \Omega(y_s)$, we require the found matrices K_i and P_i to cause the contraction of the terminal cost on the nominal system, i.e.

$$A_{K,i,j}^\top P_i A_{K,i,j} \leq \zeta P_i,$$

   with $\zeta \in (0,1)$. Then, for all $y_s \in \mathcal{Y}_{s_i}$ and for all $x \in \Omega(y_s)$, it is verified that $f\left(\hat{x}, \pi\left(y_s, \hat{x}, v_f(y_s)\right), 0\right) \in \mathcal{X}_f(y_s)$. If the last condition is not satisfied, we try again with a new contraction factor ζ.

2.7.2 Computation of the constants $L_{x,ia}$, $L_{v,ib}$ and $L_{w,ic}$ for component-wise Lipschitz continuous systems

In this subsection, a method to compute constants $L_{x,ia}$, $L_{v,ib}$ and $L_{w,ic}$ for component-wise Lipschitz continuous systems is presented such that, for all $i \in [1, n]$,

$$\begin{aligned} |f_i(x, \pi(y_s, x, v), w) - f_i(\check{x}, \pi(y_s, \check{x}, \check{v}), \check{w})| \leq \sum_{a=1}^{n} L_{x,ia}|x_a - \check{x}_a| + \sum_{b=1}^{m} L_{v,ib}|v_b - \check{v}_b| \\ + \sum_{c=1}^{r} L_{w,ic}|w_c - \check{w}_c|. \end{aligned} \tag{2.38}$$

is verified.

For each $i \in [1, n]$, provided a candidate tuple $(L_{x,i1}, \ldots, L_{x,in}, L_{v,i1}, \ldots, L_{v,im}, L_{w,i1}, \ldots, L_{w,ir})$, solve the optimization problem

$$\begin{aligned} \max_{(x,v,w),(\check{x},\check{v},\check{w}),y_s} \quad & e_i := |f_i(x, \pi(y_s, x, v), w) - f_i(\check{x}, \pi(y_s, \check{x}, \check{v}), \check{w})| \\ & - \sum_{a=1}^{n} L_{x,ia}|x_a - \check{x}_a| - \sum_{b=1}^{m} L_{v,ib}|v_b - \check{v}_b| - \sum_{c=1}^{r} L_{w,ic}|w_c - \check{w}_c| \\ & \text{s.t.}: \\ & (x, \pi(y_s, x, v), w), (\check{x}, \pi(y_s, \check{x}, \check{v}), \check{w}) \in \mathcal{Z} \times \mathcal{W}, \\ & y_s \in \mathcal{Y}_t. \end{aligned}$$

If $e_i \leq 0$, then the candidate tuple verifies (2.38). Otherwise, choose a different candidate tuple and solve the previous optimization problem again.

Remark 2.10
Given that it is important to find small component-wise Lipschitz constants, we suggest to start from large values for all of them, so as to verify (2.38)*, and then start gradually decreasing the value of each constant till the condition $e_i \leq 0$ is lost.*

2.7.3 Computation of $\mathcal{F}(j)$ and $\mathcal{R}(j)$

In this subsection, a method to compute the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ for component-wise Lipschitz systems is presented, namely systems such that there exist non-negative constants $L_{x,ia}$, $L_{v,ib}$, and $L_{w,ic}$ that verify inequality (2.38).

In the component-wise Lipschitz case, the previously introduced functions $c_{i,j}(w)$ can be redefined as follows. For all $i = 1, \ldots, n$, define $c_{i,0}(w) := \sum_{c=1}^{r} L_{w,ic} \cdot |w_c|$. Then, for $j > 0$, $c_{i,j}(w) := \sum_{a=1}^{n} L_{x,ia} \cdot c_{a,j-1}(w)$.

Since $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ have to take into account all the possible realizations of $w \in \mathcal{W}$, we consider the worst case, i.e. a realization $\breve{w}$ such that $|\breve{w}_i| = \bar{w}_i$, for all $i \in [1, r]$. Therefore, for all $j \geq 0$, $\mathcal{F}(j)$ can be obtained as

$$\mathcal{F}(j) = \left\{x \in \mathbb{R}^n :\ |x_i| \leq c_{i,j}(\breve{w}),\ \forall i \in [1, n]\right\}. \tag{2.39}$$

Now we introduce the functions $d_{i,j}(\cdot) \coloneqq \sum_{k=0}^{j-1} c_{i,k}(\cdot)$, which can be used to compute $\mathcal{R}(j)$ as

$$\mathcal{R}(j) = \left\{x \in \mathbb{R}^n :\ |x_i| \leq d_{i,j}(\breve{w}),\ \forall i \in [1, n]\right\}. \tag{2.40}$$

Next, we present the lemma thanks to which the defined sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ ensure that Assumptions 2.5 and 2.6 are verified.

Lemma 2.2: Consistency of the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ with Assumptions 2.5 and 2.6

Given a system that verifies (2.38)*, the set sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ computed as in* (2.39) *and* (2.40) *verify all the conditions of Assumptions 2.5 and 2.6.*

Proof. This proof follows similar arguments as in [89] and [102].

Sequence $\{\mathcal{F}(j)\}_{j\geq 0}$: Given that in the component-wise Lipschitz case, $\sigma_{w,ia}(w_a) = L_{w,ia} \cdot |w_a|$, defining

$$\mathcal{F}(0) = \left\{x \in \mathbb{R}^n :\ |x_i| \leq \sum_{a=1}^{r} L_{w,ia}|\bar{w}_a|, \forall i \in [1, n]\right\}$$

is consistent with the definition of $\mathcal{F}(0)$ given in Assumption 2.5. Analyzing the gap between two nominal trajectories obtained starting from two initial conditions $x, \breve{x}$ such that $x - \breve{x} \in \mathcal{F}(0)$ and employing a feasible control sequence $\mathbf{v}$, it can be noted that, for all $i \in [1, n]$,

$$\begin{aligned}
|\phi_{\pi,i}(1; x, y_s, \mathbf{v}, \mathbf{0}) - \phi_{\pi,i}(1; \breve{x}, y_s, \mathbf{v}, \mathbf{0})| &\leq \sum_{a=1}^{n} L_{x,ia}|x_a - \breve{x}_a| \\
&\leq \sum_{a=1}^{n} L_{x,ia} c_{a,0}(\breve{w}) = c_{i,1}(\breve{w}),
\end{aligned}$$

which proves that $\phi_\pi(1; \breve{x}, y_s, \mathbf{v}, \mathbf{0}) \in \phi_\pi(1; x, y_s, \mathbf{v}, \mathbf{0}) \oplus \mathcal{F}(1)$.

We now analyze what happens for $j = 2$, i.e.

$$\begin{aligned}
&|\phi_{\pi,i}(2; x, y_s, \mathbf{v}, \mathbf{0}) - \phi_{\pi,i}(2; \breve{x}, y_s, \mathbf{v}, \mathbf{0})| \\
&\qquad\leq \sum_{a=1}^{n} L_{x,ia}|\phi_{\pi,a}(1; x, y_s, \mathbf{v}, \mathbf{0}) - \phi_{\pi,a}(1; \breve{x}, y_s, \mathbf{v}, \mathbf{0})| \\
&\qquad\leq \sum_{a=1}^{n} L_{x,ia} c_{a,1}(\breve{w}) = c_{i,2}(\breve{w}),
\end{aligned}$$

which proves that $\phi_\pi(2;\check{x}, y_s, \mathbf{v}, \mathbf{0}) \in \phi_\pi(2; x, y_s, \mathbf{v}, \mathbf{0}) \oplus \mathcal{F}(2)$.

Generalizing for all $j > 1$, we get that

$$\begin{aligned}|\phi_{\pi,i}(j; x, y_s, \mathbf{v}, \mathbf{0}) - \phi_{\pi,i}(j; \check{x}, y_s, \mathbf{v}, \mathbf{0})| \\ &\leq \sum_{a=1}^{n} L_{x,ia} |\phi_{\pi,a}(j-1; x, y_s, \mathbf{v}, \mathbf{0}) - \phi_{\pi,a}(j-1; \check{x}, y_s, \mathbf{v}, \mathbf{0})| \\ &\leq \sum_{a=1}^{n} L_{x,ia} c_{a,j-1}(\check{w}) = c_{i,j}(\check{w}),\end{aligned}$$

allowing us to conclude that the sequence $\{\mathcal{F}(j)\}_{j\geq 0}$ verifies all the conditions of Assumption 2.5.

Sequence $\{\mathcal{R}(j)\}_{j\geq 0}$: Analyzing the gap between a perturbation-free trajectory and a perturbed trajectory obtained starting from the same initial condition x and employing the same feasible control sequence $\mathbf{v}$, it can be noted that, for all $i \in [1, n]$,

$$\begin{aligned}|\phi_{\pi,i}(1; x, y_s, \mathbf{v}, \mathbf{w}) - \phi_{\pi,i}(1; x, y_s, \mathbf{v}, \mathbf{0})| &\leq \sum_{a=1}^{r} L_{w,ia} |\bar{w}_a| \\ &= c_{i,0}(\check{w}) = d_{i,1}(\check{w}),\end{aligned}$$

which proves that $\phi_\pi(1; x, y_s, \mathbf{v}, \mathbf{w}) \in \phi_\pi(1; x, y_s, \mathbf{v}, \mathbf{0}) \oplus \mathcal{R}(1)$. Then we analyze the gap after 2 steps, namely

$$\begin{aligned}|\phi_{\pi,i}(2; x, y_s, \mathbf{v}, \mathbf{w}) - \phi_{\pi,i}(2; x, y_s, \mathbf{v}, \mathbf{0})| &\leq \sum_{a=1}^{r} L_{w,ia} |\bar{w}_a| \\ &+ \sum_{b=1}^{n} L_{x,ib} |\phi_{\pi,b}(1; x, y_s, \mathbf{v}, \mathbf{w}) - \phi_{\pi,b}(1; x, y_s, \mathbf{v}, \mathbf{0})| \\ &\leq c_{i,0}(\check{w}) + \sum_{b=1}^{n} L_{x,ib} c_{b,0}(\check{w}) \\ &= c_{i,0}(\check{w}) + c_{i,1}(\check{w}) = d_{i,2}(\check{w}),\end{aligned}$$

which proves that $\phi_\pi(2; x, y_s, \mathbf{v}, \mathbf{w}) \in \phi_\pi(2; x, y_s, \mathbf{v}, \mathbf{0}) \oplus \mathcal{R}(2)$. Finally, generalizing $\forall j > 1$,

$$\begin{aligned}|\phi_{\pi,i}(j; x, y_s, \mathbf{v}, \mathbf{w}) - \phi_{\pi,i}(j; x, y_s, \mathbf{v}, \mathbf{0})| &\leq \sum_{a=1}^{r} L_{w,ia} |\bar{w}_a| \\ &+ \sum_{b=1}^{n} L_{x,ib} |\phi_{\pi,b}(j-1; x, y_s, \mathbf{v}, \mathbf{w}) - \phi_{\pi,b}(j-1; x, y_s, \mathbf{v}, \mathbf{0})| \\ &\leq \sum_{k=0}^{j-1} c_{i,k}(\check{w}) = d_{i,j}(\check{w}),\end{aligned}$$

which allows us to state that the first condition of Assumption 2.6 is verified. We now turn our attention to the monotonicity of the sequence $\{\mathcal{R}(j)\}_{j\geq 0}$. Note that, as the sets $\mathcal{R}(j)$ and $\mathcal{F}(j)$ are box-shaped, we get that

$$\mathcal{R}(j) \oplus \mathcal{F}(j) = \left\{x \in \mathbb{R}^n : \; |x_i| \leq c_{i,j}(\check{w}) + d_{i,j}(\check{w}), \forall i \in [1, n]\right\},$$

but $c_{i,j}(\check{w}) + d_{i,j}(\check{w}) = d_{i,j+1}(\check{w})$. Consequently, $\mathcal{R}(j) \oplus \mathcal{F}(j) = \mathcal{R}(j+1)$. ■

Remark 2.11

We would like to point out that the choice of showing a method to compute the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ that relies on component-wise Lipschitz constants does not lead to any loss of generality for the theory presented in the previous sections. In general, for many subfamilies of component-wise uniformly continuous systems, it is possible to compute the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ by means of toolboxes like CORA, which was presented in [7].

2.8 Case studies

In this section, the here-proposed controller is tested on two different systems, a nonlinear mass-spring-damper and a quadruple-tank system. In particular, these two examples show how a thoughtful design of the pre-stabilizing control laws can ease the design of the MPC in certain situations and two different algorithms for the design of the terminal ingredients are used.

2.8.1 Control of an uncertain nonlinear mass-spring-damper system

In this subsection, the here-proposed controller is tested on a mass-spring-damper system inspired by [96]. In particular, a carriage of mass M moves on a plane and is attached to the wall by means of a spring with nonlinear elasticity k that is given by $k = k_0 e^{-|x_1|}$, where x_1 [m] is the displacement of the carriage from the equilibrium position that is related to a null external force, i.e. $F_{\text{ext}} = 0\text{N}$. Furthermore, the damping factor h_d of the damper is considered to be uncertain, i.e. $h_d = \bar{h}_d + \Delta h_d$. The carriage displacement is assumed to be the output of the system, whose state is supposed to be fully accessible. The continuous-time state-space model of the system is

$$
\begin{aligned}
\dot{x}_1(t) &= x_2(t),\\
\dot{x}_2(t) &= -\frac{k_0}{M} e^{-|x_1(t)|} x_1(t) - \frac{\bar{h}_d + \Delta h_d}{M} x_2(t) + \frac{F_{\text{ext}}(t)}{M},\\
y(t) &= x_1(t),
\end{aligned}
\tag{2.41}
$$

where the external force $F_{\text{ext}}(t)$ is the controlled input. The parameters of the system are $M = 1\text{kg}$, $k_0 = 0.33$, and $\bar{h}_d = 1.1$. The damping factor uncertainty is considered to be bounded and such that $\Delta h_d \in [-0.125, 0.125]$.

For simulation purposes, a forward Euler discretization of (2.41) is considered, using a sampling time $T_s = 0.4$s. The resulting discrete-time system is provided in (2.42).

$$\begin{aligned} x_1(k+1) &= x_1(k) + T_s x_2(k) \\ x_2(k+1) &= x_2(k) - T_s \frac{k_0}{M} e^{-|x_1(k)|} x_1(k) - T_s \frac{\bar{h}_d + \Delta h_d}{M} x_2(k) + T_s \frac{F_{\text{ext}}(k)}{M}, \\ y(k) &= x_1(k). \end{aligned} \tag{2.42}$$

The control goal is to design a controller that allows the closed-loop system to track a piece-wise constant output reference signal that takes values 2.6m, -1m, 3.8m, and 0m, considering that the state and input constraints

$$\begin{gathered} -1.5\text{m} \le x_1(k) \le 4.5\text{m} \\ -1.5\text{m/s} \le x_1(k) \le 1.5\text{m/s} \\ -3.5\text{N} \le F_{\text{ext}}(k) \le 3.5\text{N} \end{gathered}$$

have to be verified for all $k \ge 0$, and for every possible damping uncertainty value $\Delta h_d \in [-0.125, 0.125]$. The initial condition of the system state is the equilibrium $x(0) = [0, 0]^\top$.

To track the previously defined output reference signal, the first approach proposed in Section 2.7.1 is exploited to get the pre-stabilizing control law $\pi(y_s, x(k), v(k)) = k_0 e^{-|x_1(k)|} x_1(k) + K_{\text{LQ}}(x(k) - g_x(y_s)) + v(k)$, where

$$K_{\text{LQ}} = \begin{bmatrix} -1.5837 & -1.6823 \end{bmatrix}$$

is the LQR that minimizes the infinite-horizon quadratic cost given by the weighting terms $Q = 10 I_2$ and $R = 1$. The usage of $\pi(y_s, x(k), v(k))$ turns the closed-loop discrete-time system into a linear system described by the following equations:

$$\begin{aligned} x_1(k+1) &= x_1(k) + T_s x_2(k) \\ x_2(k+1) &= x_2(k) - T_s \frac{\bar{h}_d + \Delta h_d}{M} x_2(k) + T_s \frac{K_{\text{LQ}}(x - g_x(y_s))}{M} + T_s \frac{v(k)}{M}, \\ y(k) &= x_1(k). \end{aligned}$$

Defining the uncertainty signal $w(k) := \Delta h_d x_2(k)$, where $\Delta h_d \in [-0.125, 0.125]$ and $x_2(k) \in [-1.5, 1.5]$m/s, the closed-loop state equations rewrite as

$$\begin{aligned} x_1(k+1) &= x_1(k) + T_s x_2(k) \\ x_2(k+1) &= x_2(k) - T_s \frac{\bar{h}_d}{M} x_2(k) + T_s \frac{K_{\text{LQ}}(x - g_x(y_s))}{M} + T_s \frac{v(k)}{M} + T_s \frac{w(k)}{M}, \\ y(k) &= x_1(k), \end{aligned} \tag{2.43}$$

where $w(k) \in \mathcal{W} := [-0.1875, 0.1875]$.

$L_{x,11}$	$L_{x,12}$	$L_{v,11}$	$L_{w,11}$	$L_{x,21}$	$L_{x,22}$	$L_{v,21}$	$L_{w,21}$
1	0.4	0	0	0.6335	0.1129	0.4	0.4

Table 1: Component-wise Lipschitz constants of the system.

	$\mathcal{F}(j)$		$\mathcal{R}(j)$	
j	$\bar{x}_{1,\mathcal{F}(j)}$	$\bar{x}_{2,\mathcal{F}(j)}$	$\bar{x}_{1,\mathcal{R}(j)}$	$\bar{x}_{2,\mathcal{R}(j)}$
0	0	0.0750	0	0
1	0.0300	0.0085	0	0.0750
2	0.0334	0.0200	0.0300	0.0835
3	0.0414	0.0234	0.0634	0.1034
4	0.0507	0.0289	0.1048	0.1268
5	0.0623	0.0354	0.1555	0.1557
6	0.0764	0.0434	0.2178	0.1911
7	0.0938	0.0533	0.2942	0.2345
8	0.1151	0.0655	0.3880	0.2879

Table 2: Description of the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ that were obtained by means of the algorithm shown in Section 2.7.3.

The linearity of the closed-loop system with respect to $x(k)$, $v(k)$, and $w(k)$ implies that the state equations (2.43) are component-wise Lipschitz continuous, namely there exist constants $L_{x,ia}$, $L_{v,ia}$, and $L_{w,ia}$ that verify the inequality (2.38). The found component-wise Lipschitz constants are reported in Table 1. We now turn our attention to the set sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$. As the computed sets are box-shaped, i.e.

$$\mathcal{F}(j) = \{x \in \mathbb{R}^3 : \ |x_i| \leq \bar{x}_{i,\mathcal{F}(j)}, \forall i \in \{1,2\}\}$$

and

$$\mathcal{R}(j) = \{x \in \mathbb{R}^3 : \ |x_i| \leq \bar{x}_{i,\mathcal{R}(j)}, \forall i \in \{1,2\}\},$$

we provide the computed $\bar{x}_{i,\mathcal{F}(j)}$ and $\bar{x}_{i,\mathcal{R}(j)}$ in Table 2.

In the context of this work, $v(k)$ is computed by the robust MPC for tracking by solving $P_{N_c,N_p}(x(k), y_t)$, with $N_c = N_p = 8$. A quadratic stage cost function $\ell(x - x_s, v - v_s) = (x - x_s)^\top Q(x - x_s) + (v - v_s)^\top R(v - v_s)$ is chosen, where $Q = 10I_2$ and $R = 1$. The terminal cost function is chosen as $V_f(x - x_s, y_s) = (x - x_s)^\top P(x - x_s)$, where P is designed to verify all the conditions of Assumption 2.7, considering the local stabilizing input $v_f(y_s) = v_s$. In detail, we have that

$$P = \begin{bmatrix} 43.9220 & 15.7860 \\ 15.7860 & 18.6697 \end{bmatrix}.$$

The employed offset cost function is $V_O(y_s - y_t) = S|y_s - y_y|$, with $S = 10^3$, which verifies Assumptions 2.9 and 2.10.

The sequences $\{\mathcal{F}(j)\}_{j\geq 0}$, $\{\mathcal{R}(j)\}_{j\geq 0}$, and $\{\mathcal{Z}_\pi(j)\}_{j\geq 0}$ are computed for all $j = 0, \ldots, N_p$ following the algorithm described in Section 2.7.3, and the found $\mathcal{Z}_\pi(N_p)$ is exploited to set $\mathcal{Y}_t = [-0.42, 3.42]$m. A robust invariant set for tracking $\mathcal{T}$ verifying all the conditions of Assumptions 2.7 is found as

$$\mathcal{T} = \left\{x : \; V_f(x - x_s, y_s) \leq \rho(y_s)\right\} \times \mathcal{Y}_t,$$

where

$$\rho(y_s) \coloneqq \begin{cases} 14.35, & y_s \in [-0.4, 3.4] \\ 13.06, & y_s \in [-0.42, -0.4) \cup (3.4, 3.42] \end{cases}$$

The projections of $\mathcal{T}$ and $\mathcal{Z}_\pi(N_p)$ on the state dimensions are shown in Figure 3. The operation of

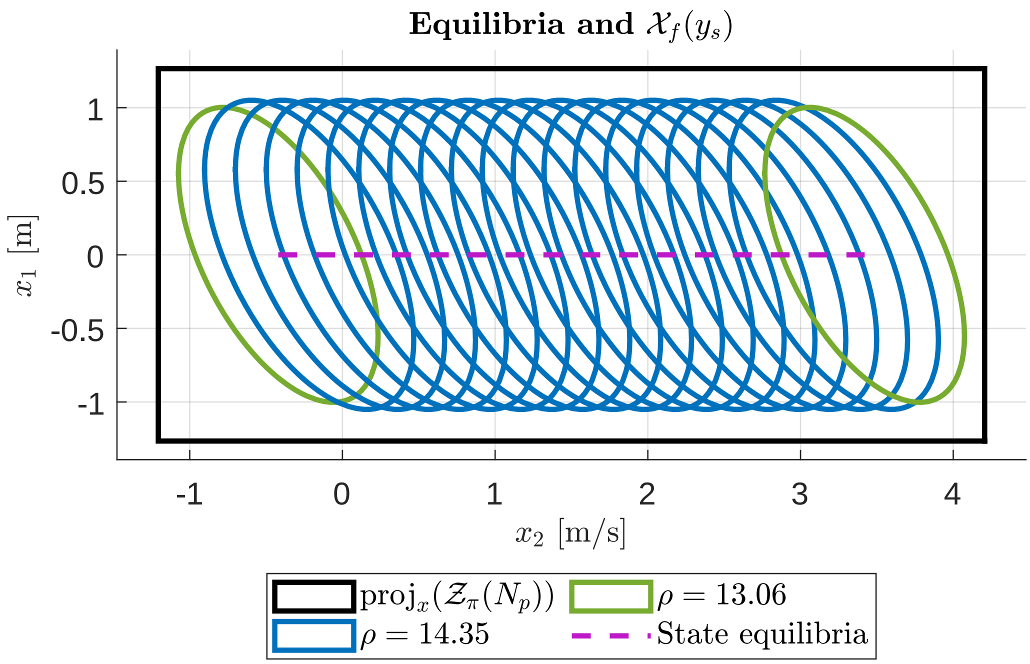


Figure 3: $\mathcal{X}_f(y_s)$ for $y_s \in [-0.59, 3.59]$m and projection of $\mathcal{Z}_\pi(N_p)$ on the state dimensions. It is possible to see that, as expected, the smaller sublevel sets $\mathcal{X}_f(y_s)$ are obtained for equilibria that are close to the boundaries of $\mathcal{Z}_\pi(N_p)$.

the closed-loop system is simulated in 10 different scenarios, which correspond to different values of the damping uncertainty, i.e.

$$\Delta h_d \in \{-0.125, -0.0972, -0.0694, -0.0417, -0.0139,$$
$$0.0139, 0.0417, 0.0694, 0.0972, 0.125\}.$$

The simulation time is always equal to 32s. The obtained state and input trajectories are shown in Figure 4, and prove how the here-proposed robust MPC for tracking is able to ensure input-to-state stability and constraint satisfaction of the closed-loop system, and to steer the system output to a neighborhood of the best admissible steady output, even in case of abruptly changing and infeasible setpoints.

2.8.2 Control of an uncertain quadruple-tank system

In the following, the here-proposed controller is tested on the quadruple-tank system with uncertain parameters presented in the Appendix C. In detail, considering a sampling time $T_s = 15$s, the system under control is its fourth-order Runge Kutta discretization.
For the sake of clarity, recall that:

- h_1, h_2, h_3, h_4 [m] are the water levels in the four tanks, which constitute the state of the system;
- q_a, q_b [m^3/h] are the flows of the two pumps of the system and are the system inputs;
- γ_a, γ_b [/] are the parameters that define how the two three-way valves operate;
- a_1, a_2, a_3, a_4 [m^2] are the sections of the tank holes, which determine the discharge speed of the four tanks;
- S [m^2] is the cross-section of all the tanks;
- $g = 9.81$ [m/s^2] is the standard acceleration of gravity.

The output of the system is $y(t) = (h_1(t), h_2(t))$. In the context of our work, we consider the parameters γ_a, γ_b to be uncertain, namely

$$\gamma_a = \bar{\gamma}_a + w_1, \tag{2.44a}$$

$$\gamma_b = \bar{\gamma}_b + w_2, \tag{2.44b}$$

where $\bar{\gamma}_a = 0.3$ and $\bar{\gamma}_b = 0.4$ are the nominal values of the parameters γ_a and γ_b, while w_1 and w_2 describe the uncertainties of such parameters and are supposed to be bounded. In detail,

$$|w_1| \leq \bar{w}_1 = 0.005, \tag{2.45a}$$

$$|w_2| \leq \bar{w}_2 = 0.005. \tag{2.45b}$$

The system inputs and states have to verify the following constraints:

$$h_{i,\min} \leq h_i \leq h_{i,\max},\ i \in [1,4], \tag{2.46a}$$

$$q_{j,\min} \leq q_j \leq q_{j,\max},\ j \in \{a,b\}. \tag{2.46b}$$

The input and state bounds are reported in the Appendix C. The control goal is to track a piece-wise constant reference signal, starting from the initial condition $(h_1(0), h_2(0), h_3(0), h_4(0)) = (0.2837, 0.2943, 0.2168, 0.2864)$m.

<table>
<tr><th colspan="2" rowspan="2">$L_{x,ia}$</th><th colspan="4">a</th></tr>
<tr><th>1</th><th>2</th><th>3</th><th>4</th></tr>
<tr><td rowspan="4">i</td><td>1</td><td>0.9388</td><td>0.0250</td><td>0.1375</td><td>0.0500</td></tr>
<tr><td>2</td><td>0.0850</td><td>0.9388</td><td>0.0795</td><td>0.1600</td></tr>
<tr><td>3</td><td>0.1225</td><td>0.0145</td><td>0.8375</td><td>0.0750</td></tr>
<tr><td>4</td><td>0.0125</td><td>0.0600</td><td>0.0600</td><td>0.8500</td></tr>
</table>

Table 3: Values of the constants $L_{x,ia}$ of the nonholonomic system.

In the following, we revert to a more generic notation and define the states and inputs of the discretized system as $x(k) = (x_1(k), x_2(k), x_3(k), x_4(k)) \coloneqq (h_1(k \cdot T_s), h_2(k \cdot T_s), h_3(k \cdot T_s), h_4(k \cdot T_s))$ and $u(k) = (u_1(k), u_2(k)) \coloneqq (q_a(k \cdot T_s), q_a(k \cdot T_s))$. Moreover, we define the sets $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{W}$ as sets such that $x(k) \in \mathcal{X}$, $u(k) \in \mathcal{U}$ and $w(k) \coloneqq (w_1(k \cdot T_s), w_2(k \cdot T_s)) \in \mathcal{W}$ imply the verification of the constraints (2.45) and (2.46). To reach the previously stated control goal, we employ a control law

$$\pi\left(y_s^0(k), x(k), v^0(0|k)\right) = K(x(k) - g_x(y_s^0(k))) + v^0(0|k), \tag{2.47}$$

where $y_s^0(k)$ and $v^0(0|k)$ are obtained by solving $P_{N_p}(x(k), y_t)$, with $N_p = 4$, employing a quadratic stage cost function $\ell(x - xs, v - v_s) = (x - x_s)^\top Q(x - x_s) + (v - v_s)^\top R(v - v_s)$, a quadratic terminal cost function $V_f(x - x_s, y_s) = (x - x_s)^\top P(x - x_s)$, and a quadratic offset cost function $V_O(y_s - y_t) = (y_s - y_t)^\top T(y_s - y_t)$. In detail, $Q = \mathrm{diag}(5, 2.5, 1, 1)$, $R = 0.01 I_2$ and $T = 10^4 I_2$.
The matrices K and P are obtained using the LTV approximation method presented in Section 2.7.1. The obtained matrices are:

$$K = \begin{bmatrix} -0.2117 & -1.0663 & 1.0632 & -2.2400 \\ -2.9453 & 0.3358 & -2.9568 & 1.5166 \end{bmatrix},$$

$$P = \begin{bmatrix} 23.7301 & -4.6424 & 10.2060 & -8.5938 \\ -4.6425 & 8.8324 & -4.7139 & 3.8709 \\ 10.2060 & -4.7139 & 12.6887 & -7.8934 \\ -8.5938 & 3.8709 & -7.8934 & 9.4075 \end{bmatrix}.$$

Given that $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{W}$ are compact and that the system under control is Lipschitz in $\mathcal{X} \times \mathcal{U} \times \mathcal{W}$, the system equations are also component-wise Lipschitz continuous in such set. Therefore, we first compute the component-wise Lipschitz continuity constants of the discretized system and then compute the sequences $\{\mathcal{F}(j)\}_{\geq 0}$ and $\{\mathcal{R}(j)\}_{\geq 0}$ following the algorithms presented in Section 2.7. The found component-wise Lipschitz constants are reported in Tables 3, 4 and 5.
Furthermore, as the computed sets $\mathcal{F}(j)$ and $\mathcal{R}(j)$ are box-shaped, i.e.

$$\mathcal{F}(j) = \{x \in \mathbb{R}^4 : \ |x_i| \leq \bar{x}_{i,\mathcal{F}(j)}, \forall i \in [1, 4]\}$$

$L_{v,ib}$		b = 1	b = 2
i	1	0.0225	0.0250
	2	0.0500	0.0350
	3	0.0100	0.1700
	4	0.1500	0.0100

Table 4: Values of the constants $L_{v,ib}$ of the nonholonomic system.

$L_{w,ic}$		c = 1	c = 2
i	1	0.2400	0.0125
	2	0.0088	0.2638
	3	0.00006	0.2675
	4	0.2450	0.0125

Table 5: Values of the constants $L_{w,ic}$ of the nonholonomic system.

j		0	1	2	3	4
$\mathcal{F}(j)$	$\bar{x}_{1,\mathcal{F}(j)}$	0.0019	0.0022	0.0025	0.0028	0.0032
	$\bar{x}_{2,\mathcal{F}(j)}$	0.0020	0.0025	0.0031	0.0036	0.0041
	$\bar{x}_{3,\mathcal{F}(j)}$	0.0020	0.0021	0.0022	0.0023	0.0025
	$\bar{x}_{4,\mathcal{F}(j)}$	0.0019	0.0019	0.0020	0.0020	0.0021
$\mathcal{R}(j)$	$\bar{x}_{1,\mathcal{R}(j)}$	0	0.0019	0.0041	0.0066	0.0094
	$\bar{x}_{2,\mathcal{R}(j)}$	0	0.0020	0.0046	0.0076	0.0112
	$\bar{x}_{3,\mathcal{R}(j)}$	0	0.0020	0.0041	0.0063	0.0086
	$\bar{x}_{4,\mathcal{R}(j)}$	0	0.0019	0.0038	0.0058	0.0078

Table 6: Description of the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$.

and

$$\mathcal{R}(j) = \{x \in \mathbb{R}^4 : \ |x_i| \leq \bar{x}_{i,\mathcal{R}(j)}, \forall i \in [1,4]\},$$

we provide the computed $\bar{x}_{i,\mathcal{F}(j)}$ and $\bar{x}_{i,\mathcal{R}(j)}$ in Table 6.

The chosen bounded subset $\mathcal{Y}_t$ of feasible setpoints with inactive constraints is provided in Figure 5, while a robust invariant set for tracking $\mathcal{T}$ verifying all the conditions of Assumption 2.7 is found as

$$\mathcal{T} = \left\{x : V_f(x - x_s, y_s) \leq \rho\right\} \times \mathcal{Y}_t,$$

where $\rho = 0.1273$.

We run 100 simulations with different values for w_1 and w_2. In detail, we test all the combinations that are obtained by using values for w_1 and w_2 in

$$\{-0.00500, -0.00390, -0.00280, -0.00170, -0.00055, 0.00055, 0.0017, 0.00280, 0.00390, 0.00500\},$$

to cover the entire set $\mathcal{W}$. The simulation time is always equal to 100min and the reference y_t is considered to take the values defined in Table 7.

Start time [s]	y_t [m]
0	$(1.10, 1.10)$
1500	$(0.35, 0.35)$
3000	$(0.60, 0.75)$
4500	$(0.90, 0.75)$

Table 7: Values of y_t.

The obtained state and input trajectories are shown in Figures 6 and 7 respectively, and prove how the here-proposed robust MPC for tracking is able to ensure input-to-state stability and constraint satisfaction of the closed-loop system, and to steer the system output to a neighborhood of the best admissible steady output, even in case of abruptly changing and infeasible setpoints.

2.9 Conclusion

In this chapter, we presented a tube-based MPC for tracking for nonlinear systems that are subject to bounded perturbations and whose state function components are component-wise uniformly continuous. The tracking capabilities of the MPC are ensured by the addition of an artificial reference as decision variable and an offset cost function, as often done in modern tracking MPC schemes. We investigated the conditions that the system under control, the cost function and the set of feasible setpoints have to meet to ensure the input-to-state stability of the closed-loop system and the recursive feasibility of the MPC. The closed-loop system output is ensured to converge to a bounded set containing the best admissible steady output by design. As the proposed method requires the computation of pre-stabilizing control laws, set sequences that bound the effect of the perturbations on the predictions and a robust invariant set for tracking, some tips and algorithms for their computation were provided. The method was tested on two perturbed nonlinear systems, which proved the robust MPC to be effective also in practice. Nonetheless, we are aware of some limitations of the proposed method. In particular, the conditions that the required robust invariant set for tracking has to satisfy are strict, thus limiting the usage of the robust MPC for tracking to systems with small perturbations. Moreover, even though the MPC is proved to stabilize the closed-loop system under a component-wise uniform continuity assumption over the state function, a practical method to compute the set sequences bounding the effect of the perturbations was provided only for systems whose state function components are component-wise Lipschitz continuous.

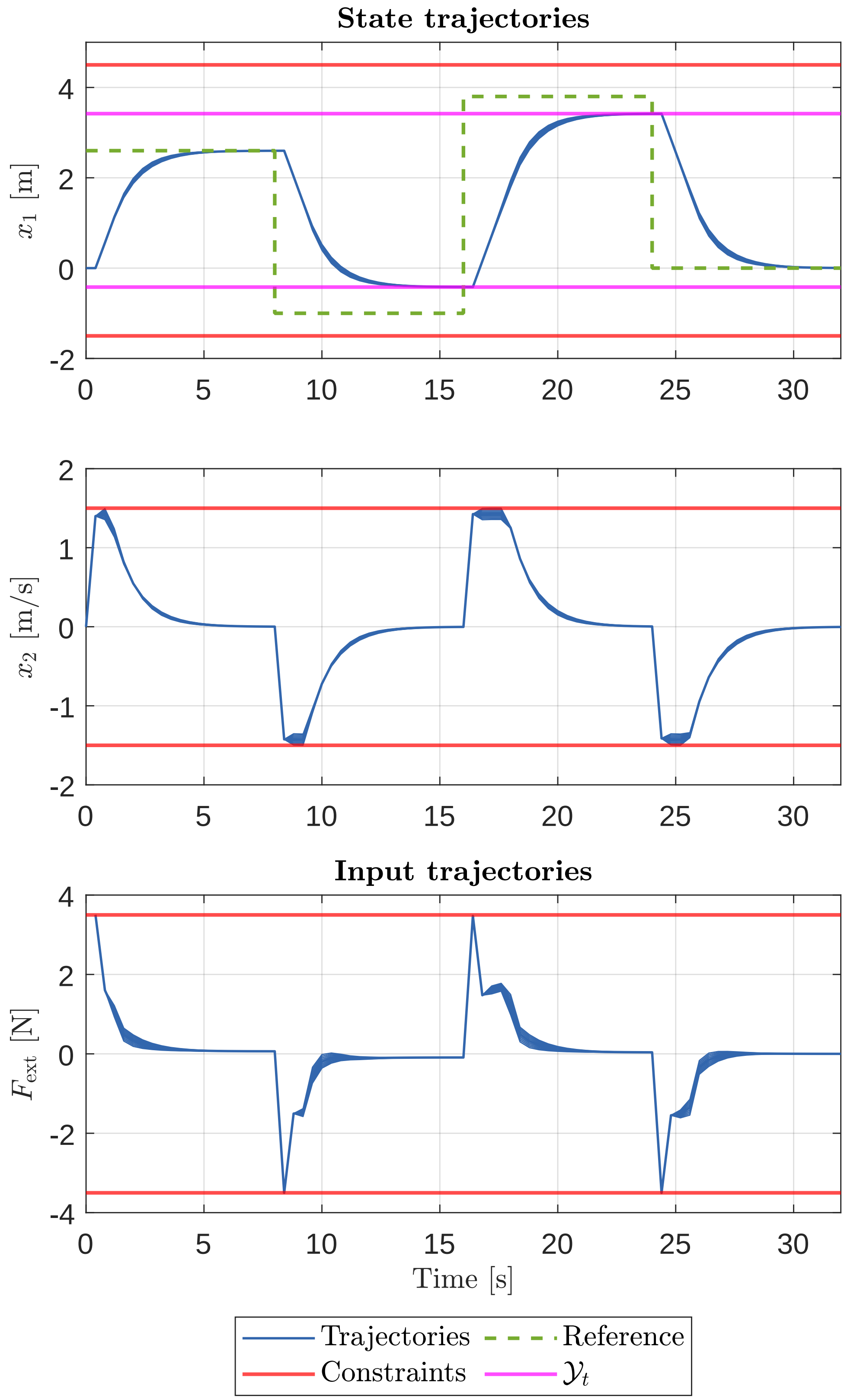


Figure 4: State and input trajectories of the closed-loop mass-spring-damper system under different Δh_d values. Note how the state constraints are verified both when the real reference lies inside the set $\mathcal{Y}_t$ of feasible setpoints and when it lies outside, regardless of the employed Δh_d value. In particular, when the reference does not belong to $\mathcal{Y}_t$, the best admissible steady output is reached, as highlighted by the first plot.

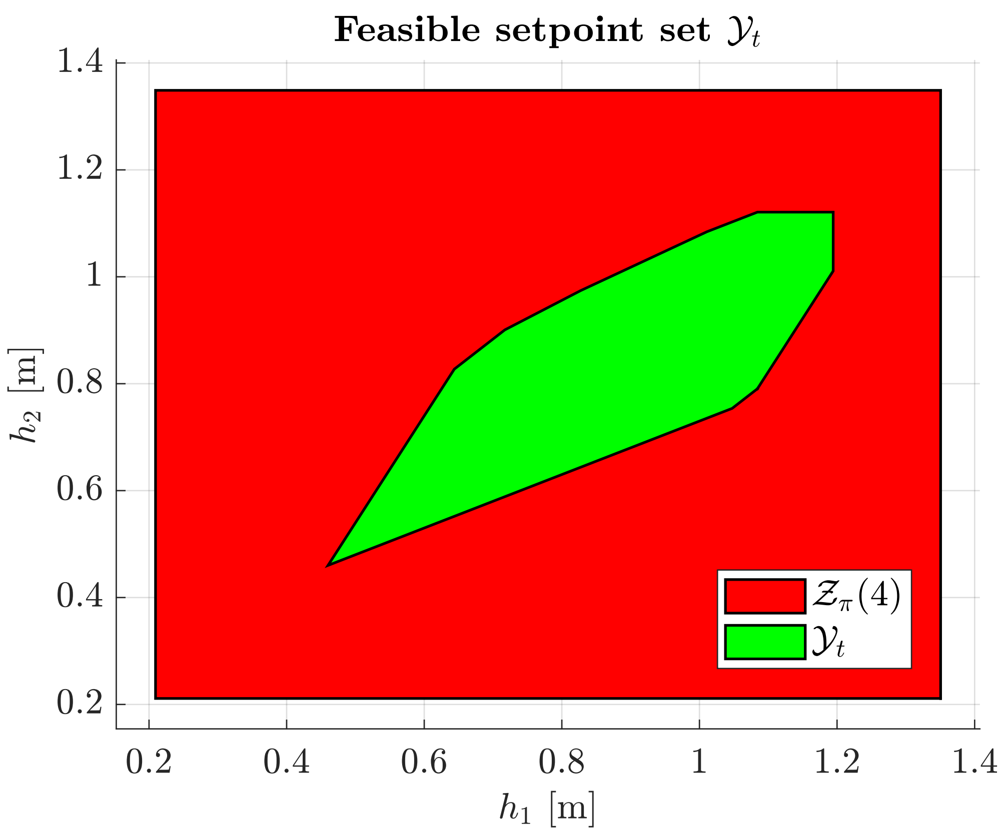


Figure 5: $\mathcal{Y}_t$ and projection of $\mathcal{Z}_\pi(4)$ on the first two state dimensions of the uncertain quadruple-tank example.

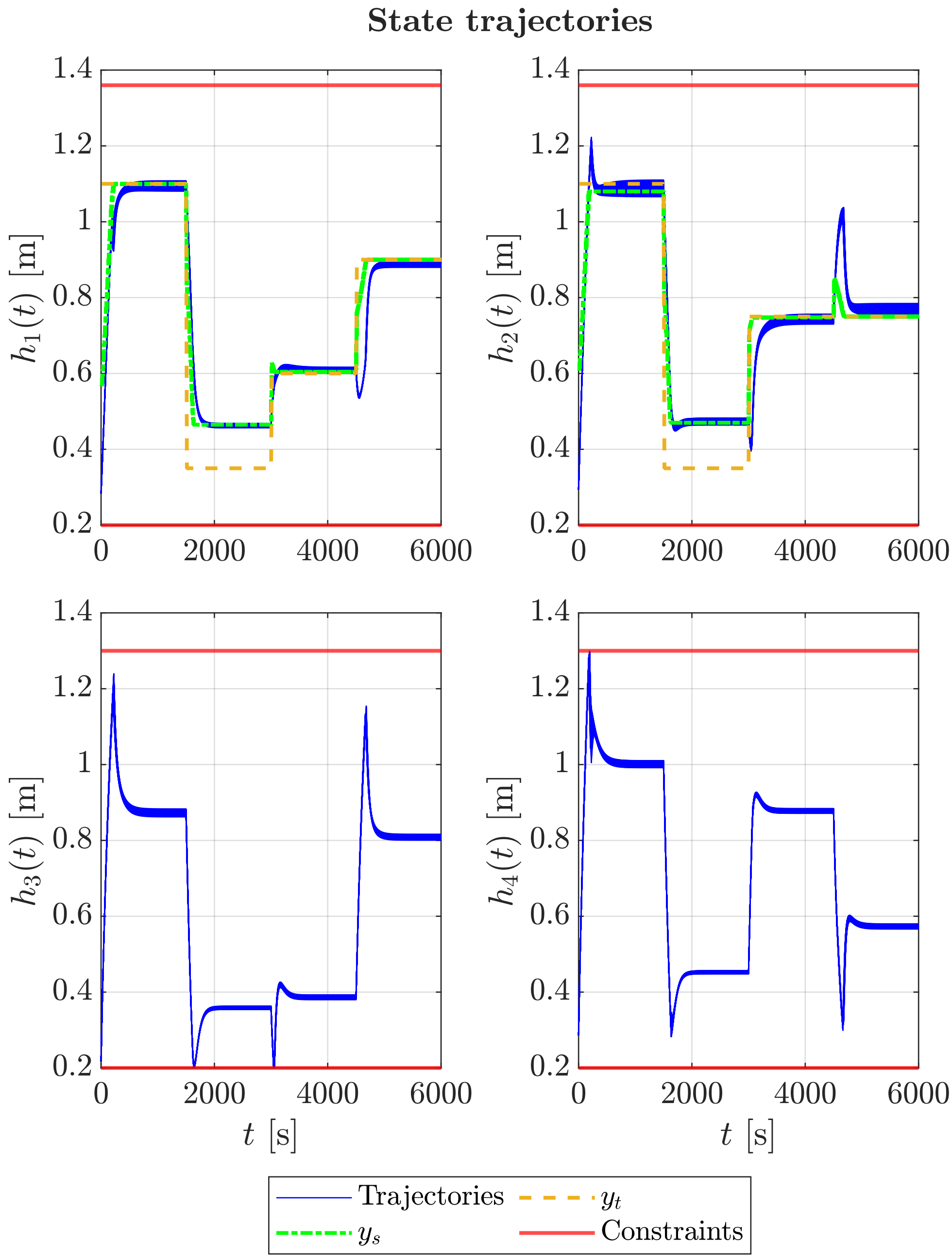


Figure 6: Trajectories of the water levels in the four tanks obtained in 100 different scenarios. The constraints are always verified, and the output of the system (in this case the first two states) converges to a neighborhood of the best admissible steady output.

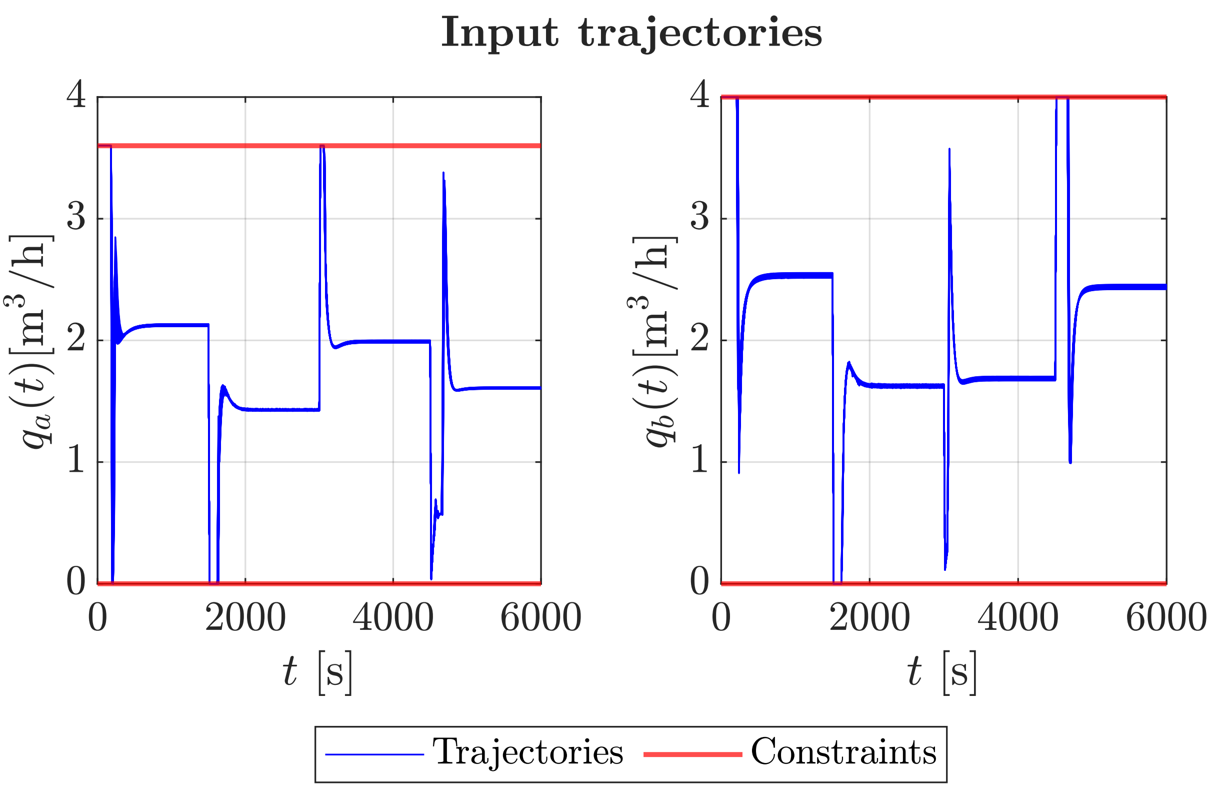


Figure 7: Input trajectories obtained in 100 different scenarios on the quadruple-tank system. As expected, also the input values always verify the constraints.

Chapter 3. Robust contraction-based model predictive control for nonlinear systems

3.1 Introduction

Chapter 2 treated a tube-based MPC for tracking for perturbed nonlinear systems and listed the conditions under which such method is recursively feasible and makes the closed-loop system ISS, ensuring also the convergence to a bounded set containing the best admissible steady output. In this chapter, we take a step back, returning to the subject of MPC schemes for regulation. As pointed out several times in Chapter 1, the main challenges that one has to face during the design of an MPC for the regulation of nonlinear systems are:

- the computation of the stability-related terminal ingredients;
- ensuring robustness with respect to known bounded perturbations.

Such issues get even more difficult to face in the case of MPC schemes for tracking changing setpoints, as they require to design (following a tube-based rationale) tightened constraints and terminal ingredients that guarantee recursive feasibility and closed-loop stability regardless of the current setpoint. This is why the contribution of the work presented in Chapter 2 is mainly related to just one of the two issues, namely robustness, as we rely on standard MPC theory for the computation of the terminal ingredients.

In this chapter, though limiting ourselves to the regulation case, we want to be more ambitious. Indeed, we want to provide a method that ensures robustness and that is able to ensure input-to-state (practical) stability of the closed-loop system employing terminal ingredients that must undergo milder conditions compared to those repeatedly exhibited up to this point. In particular, we focus on perturbed nonlinear systems that cannot be nominally controlled to the desired equilibrium within a finite number of steps, thus precluding the implementation of a TEC-MPC, and for which the computation of a terminal cost function that behaves as a CLF is difficult, e.g. systems whose linearization at the desired equilibrium is not fully controllable, hence making the design of a QIH-MPC tedious. An additional goal is that of ensuring recursive feasibility and closed-loop stability under conditions that are easier to verify than those shown in Section 1.1.4.

To do so, we rely on contractivity, a nice property of stabilizable systems that is addressed next, inspired by [4, 5]. The resulting MPC is proved to be recursively feasible and to ensure the input-to-state stability of the closed-loop system. The method is then validated in simulation on two different systems: (i) a perturbed nonholonomic system whose linearization is not fully controllable, and (ii) a perturbed quadruple-tank system.

The results presented in this chapter are available as the preprint [114].

3.2 Contractivity of stabilizable systems

We here provide a definition of stabilizability for generic nonlinear time-invariant discrete-time systems of the form

$$x(k+1) = f\left(x(k), u(k)\right), \tag{3.1}$$

where $x(k) \in \mathbb{R}^n$ is the system state and $u(k) \in \mathbb{R}^m$ is the system input. The following definition is a discrete-time adaptation of the stabilizability definition provided in [4, Chapter 4].

Definition 3.1: Stabilizability

Given the system (3.1)*, if there exists a control law* $u(k) = \kappa\left(x(k)\right)$ *such that the closed-loop system* $x(k+1) = f\left(x(k), \kappa\left(x(k)\right)\right)$ *admits* $x = 0$ *as globally asymptotically stable (GAS) equilibrium with some continuous Lyapunov function* $V(x)$*, i.e.*

$$V\left(f\left(x, \kappa\left(x\right)\right)\right) - V\left(x\right) \leq -\alpha_V\left(x\right), \tag{3.2}$$

where $\alpha_V(\cdot)$ *is a positive definite function of the state, then the system* (3.1) *is stabilizable.*

The last definition simply states that a system is said to be stabilizable if there exists at least one control law that renders the origin a GAS equilibrium. It is not necessary to know the control law $\kappa(\cdot)$ and the related Lyapunov function $V(\cdot)$.

The connection between stabilizability and contractivity is explained in [4, Proposition 4.1] and here reported for the sake of completeness.

Proposition 3.1: Contractivity of stabilizable systems

If system (3.1) *is stabilizable, then for all the compact subsets* $\mathcal{X}_{\text{sub}} \subset \mathbb{R}^n$ *and for all the positive definite functions* $\Gamma(\cdot)$*, there exist a contraction factor* $\gamma \in (0, 1)$ *and a finite* $N_p \in \mathbb{N} \setminus \{0\}$ *such that, for all* $x \in \mathcal{X}_{\text{sub}}$*, the condition*

$$\min_{\mathbf{u}(k)} \min_{q \in \{1,\ldots,N_p\}} \Gamma\left(\phi\left(q; x, \mathbf{u}(k)\right)\right) \leq \gamma \Gamma(x), \tag{3.3}$$

where $\mathbf{u}(k) \coloneqq \{u(0|k), \ldots, u(q-1|k)\}$*, is verified.*

As will become clear in the following sections, this connection between stabilizability and contractivity can turn out to be useful in the design of predictive controllers with short prediction horizons and without terminal constraints.

3.3 Problem formulation

We consider a perturbed nonlinear discrete-time time-invariant system described by the equation

$$x(k+1) = f\big(x(k), u(k), w(k)\big) \tag{3.4}$$

where $x(k) \in \mathbb{R}^n$ is the system state, $u(k) \in \mathbb{R}^m$ is the controlled input, and $w(k) \in \mathcal{W} \subset \mathbb{R}^r$ is an unknown but bounded perturbation that lies in a known hypercube $\mathcal{W} := \{w : |w_i| \leq \bar{w}_i, \ \forall i \in [1, r]\}$. The system state x is assumed to be measured at each sampling instant $k \geq 0$.
The system is subject to hard state and input constraints, namely

$$x(k) \in \mathcal{X}, \ u(k) \in \mathcal{U}, \quad \forall k \geq 0. \tag{3.5}$$

Next, the needed continuity assumption for $f(\cdot,\cdot,\cdot)$ and the conditions that the sets $\mathcal{X}$ and $\mathcal{U}$ must meet to ensure the recursive feasibility and stability of the proposed robust contraction-based MPC are provided.

Assumption 3.1: Continuity of the state function

The state function $f(\cdot,\cdot,\cdot)$ is such that each component $f_i(\cdot,\cdot,\cdot)$, $\forall i \in [1,n]$, is component-wise uniformly continuous. Consequently, there exist functions $\sigma_{x,ia}(\cdot), \sigma_{u,ib}(\cdot), \sigma_{w,ic}(\cdot) \in \mathcal{K}$ such that, for all $i \in [1,n]$,

$$\begin{aligned} |f_i(x,u,w) - f_i(\check{x},\check{u},\check{w})| \leq \sum_{a=1}^{n} \sigma_{x,ia}\left(x_a - \check{x}_a\right) + \sum_{b=1}^{m} \sigma_{u,ib}\left(|u_b - \check{u}_b|\right) \\ + \sum_{c=1}^{r} \sigma_{w,ic}\left(|w_c - \check{w}_c|\right), \end{aligned} \tag{3.6}$$

for all (x,u,w) and $(\check{x},\check{u},\check{w})$ in $\mathcal{X} \times \mathcal{U} \times \mathcal{W}$.

Assumption 3.2: Properties of the constraint sets

The sets $\mathcal{X}$ and $\mathcal{U}$ are compact and their interior is non-empty.

Assumption 3.1 poses conditions on the state function that are identical to those posed by Assumption 2.1 for our robust tracking MPC for nonlinear systems. However, Assumption 3.2 is stronger than the assumptions on the constraint sets that have been presented up to this point, as it requires the sets $\mathcal{X}$ and $\mathcal{U}$ to be compact, i.e. closed and bounded.

In the previous chapters, different MPC formulations were addressed keeping the same conceptual order when discussing the assumptions that are needed to prove their feasibility and stability properties. Given that the theoretical framework that is used to prove the properties of the here-proposed contraction-based MPC differs consistently from the framework that was used in the previous chapters, the succession of assumptions that follows deviates from the previous ones.

Given that the robust MPC for regulation discussed here is based on the contractivity of the system, we provide a contractivity assumption.

Assumption 3.3: Contractivity of the nominal system

There exist a positive definite function $\Gamma(x)$, a contraction factor $\gamma \in (0, 1)$, and a prediction horizon $N_p \in \mathbb{N} \setminus \{0\}$ such that, for all the initial conditions $x \in \mathcal{X}$, there exists a control sequence $\mathbf{u} \in \mathcal{U}^{N_p}$ that verifies:

$$\underline{\Gamma}(x, \mathbf{u}, N_p) := \min_{j=1}^{N_p} \Gamma\left(\phi(j; x, \mathbf{u}, \mathbf{0})\right) \leq \gamma\Gamma(x). \tag{3.7}$$

Moreover, the contractive function $\Gamma(x)$ is uniformly continuous in $\mathcal{X}$, i.e. there exists a $\mathcal{K}$ function $\sigma_\Gamma(\cdot)$ such that

$$|\Gamma(x) - \Gamma(\check{x})| \leq \sigma_\Gamma\left(\|x - \check{x}\|\right), \ \forall x, \check{x} \in \mathcal{X}. \tag{3.8}$$

Finally, the function $\Gamma(\cdot)$ is upper-bounded by a $\mathcal{K}_\infty$ function $\alpha_\Gamma(\cdot)$, such that

$$\Gamma(x) \leq \alpha_\Gamma(\|x\|), \ \forall x \in \mathcal{X}. \tag{3.9}$$

Henceforward, the minimizer of (3.7) over an horizon of length N_p and given the initial condition x is denoted $j_{\text{opt}}(x, \mathbf{u}, N_p)$.

Note that Assumption 3.3 does not require all the trajectories providing the best contraction factor to lie inside $\mathcal{X}$, which was instead the case of [5]. This is because [5] requires the exact knowledge of the compact set that ensures the desired contraction, which coincides with the domain of attraction of their contraction-based MPC. As the computation of such set is challenging, we do not require to analytically find the region of attraction of the MPC and we rely on recursive feasibility to ensure that, if the initial condition renders the optimal control problem feasible, feasibility will be guaranteed also afterwards.

Remark 3.1

The choice of the contractive function $\Gamma(\cdot)$ and the computation of γ can be carried out in different ways. In the case study presented in [5], an analytical way of finding γ for a given contractive function and a particular system is reported. However, also other methods can be considered, as the one we propose in Section 3.7.

The aim is to design an MPC that can drive the system state to an equilibrium $x_s = f(x_s, u_s, 0)$. Without loss of generality, in the next sections the system state is regulated to the origin.

In the next sections, the further properties that the system under control must have to enable the robust design of the contraction-based MPC are first addressed. Then, two formulations of the proposed MPC are presented, proving the stability and feasibility properties of both.

3.4 Robust design of the controller

The MPC belongs to the family of open-loop tube-based predictive controllers, hence its recursive feasibility is guaranteed by imposing that the nominal system trajectory stays inside a sequence of tightened constraints [87, 89], and no feedback control laws are exploited to reduce the effect of the perturbations on the state predictions.

Remark 3.2

It is still possible to employ the theoretical concepts introduced in this chapter in the feedback MPC framework. The main difference in such case is that a contractive function $\Gamma(\cdot)$ should be found for the nominal closed-loop system under the chosen feedback control law. Then, the presented contraction-based MPC should be modified in order to have the free parameters of the chosen control law as decision variables.

The computation of the needed tightened constraints requires the existence of the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$, whose properties are outlined in the following two assumptions.

Assumption 3.4: Bounds on the effect of the initial condition on the state trajectory

The sequence $\{\mathcal{F}(j)\}_{j\geq 0}$ is such that:

- $\mathcal{F}(0) := \{x \in \mathbb{R}^n : |x_i| \leq \sum_{a=1}^{r} \sigma_{w,ia}(\bar{w}_a), \forall i \in [1, n]\}$;

- $\forall j > 0$, *the sets* $\mathcal{F}(j)$ *ensure that, for every feasible* $(x, \mathbf{u})$*, and for every* $\check{x}$ *such that* $\check{x} - x \in \mathcal{F}(0)$*, the condition*

$$\phi\big(j; \check{x}, \mathbf{u}, \mathbf{0}\big) \in \phi\big(j; x, \mathbf{u}, \mathbf{0}\big) \oplus \mathcal{F}(j)$$

is verified.

Assumption 3.5: Bounds on the effect of the perturbations on the state trajectory

The set sequence $\{\mathcal{R}(j)\}_{j\geq 0}$ *is such that, defining* $\mathcal{R}(0) \coloneqq \{0\}$*, it verifies the following conditions:*

1. *for every feasible* $(x, \mathbf{u})$*, the sets* $\mathcal{R}(j)$*, for* $j > 0$*, ensure* $\phi(j; x, \mathbf{u}, \mathbf{w}) \in \phi(j; x, \mathbf{u}, \mathbf{0}) \oplus \mathcal{R}(j)\ \forall \mathbf{w} \in \mathcal{W}^j$*;*
2. *the set* $\mathcal{X} \ominus \mathcal{R}(N_p)$ *is non-empty and contains the origin;*
3. $\mathcal{F}(j) \oplus \mathcal{R}(j) \subseteq \mathcal{R}(j+1)$*,* $\forall j \geq 0$.

The sequence $\{\mathcal{F}(j)\}_{j\geq 0}$ bounds the gap between the perturbation-free state trajectories obtained with the same control sequence starting from close enough states x and $\check{x}$, while the sequence $\{\mathcal{R}(j)\}_{j\geq 0}$ bounds the gap between the perturbed and the perturbation-free trajectories obtained starting from a common state x and applying the same control sequence. Thus, Assumptions 3.4 and 3.5 play the same role of Assumptions 2.5 and 2.6, that were introduced in Chapter 2 to guarantee the robustness of our robust MPC for tracking.

Another ingredient that is needed to ensure the stability of the closed-loop system under the MPC is the existence of an RPI that verifies the conditions listed in the following assumption.

Assumption 3.6: Existence of an RPI

There exists a set $\Omega \subseteq \mathcal{X}$ *that verifies the following conditions simultaneously:*

1. *for all* $x \in \Omega$*,* $\exists u \in \mathcal{U}$ *such that* $f(x, u, w) \in \Omega$*, for any* $w \in \mathcal{W}$*;*
2. $\Omega \subseteq \mathcal{Q}\left(\mathcal{X} \ominus \mathcal{R}(1)\right)$ *for the nominal system* $\hat{x}(k+1) = f(\hat{x}(k), u(k), 0)$*, where* $\mathcal{Q}\left(\mathcal{X} \ominus \mathcal{R}(1)\right)$ *denotes the one-step controllable set to* $\mathcal{X} \ominus \mathcal{R}(1)$*;*
3. $\Omega \ominus \mathcal{R}(1)$ *is non-empty and contains the origin.*

Remark 3.3

Standard MPC schemes require terminal ingredients that meet several conditions simultaneously. In detail, it is necessary to find a positively invariant set (or an RPI in the robust case) generated by a certain control law $\kappa_f(\cdot)$, which must also be stabilizing and cause the terminal cost function $V_f(\cdot)$ to be a CLF for the closed-loop system.

Assumption 3.6 still requires to compute an RPI, but this does not have to be related to a particular (stabilizing) control law. Moreover, it is not required to design a CLF.

3.5 Nominal robust contraction-based MPC for regulation

Now that all the conditions that the system under control has to verify to enable the implementation of our controller have been explained, we are ready to introduce the robust contraction-based MPC, explain its design parameters and prove its stability and feasibility properties.

The control objective is assumed to be expressed by means of a stage cost function $\ell(x, u)$. Considering a generic prediction horizon q starting from a time instant k, the total cost related to the control objective is given by

$$J\big(x(k), \mathbf{u}(k), q(k)\big) \coloneqq \sum_{j=0}^{q(k)-1} \ell(\hat{x}(j|k), u(j|k)), \tag{3.10}$$

where $\mathbf{u}(k) \coloneqq \{u(0|k), \ldots, u(q(k) - 1|k)\}$ and $\hat{x}(j|k)$ denotes the nominal j-step-ahead state prediction computed starting from the initial condition $x(k)$, namely $\hat{x}(j|k) \coloneqq \phi(j; x(k), \mathbf{u}(k), \mathbf{0})$. The stage cost function is required to fulfill the conditions listed in the next assumption.

Assumption 3.7: Stage cost function properties

There exist a constant $\bar{\ell} > 0$ and a $\mathcal{K}$ function $\alpha_\ell(\cdot)$ such that

$$\forall (x, u) \in \mathcal{X} \times \mathcal{U}, \quad 0 \leq \ell(x, u) \leq \bar{\ell}, \tag{3.11}$$

and $\ell(x, u) \geq \alpha_\ell(\|x\|)$, $\forall u \in \mathcal{U}$.

Moreover, the stage cost function $\ell(x, u)$ is uniformly continuous with respect to all its arguments; therefore, there exist $\mathcal{K}$ functions $\sigma_{\ell,x}(\cdot)$ and $\sigma_{\ell,u}(\cdot)$ such that, $\forall x, \check{x} \in \mathcal{X},\ u, \check{u} \in \mathcal{U}$:

$$|\ell(x, u) - \ell(\check{x}, \check{u})| \leq \sigma_{\ell,x}(\|x - \check{x}\|) + \sigma_{\ell,u}(\|u - \check{u}\|)\,. \tag{3.12}$$

The optimization problem $P_{N_p}(x(k), \theta(k))$ that has to be solved at each MPC step is

$$\min_{(\mathbf{u}(k), q(k))} V_{N_p}(x(k), \theta(k), \mathbf{u}(k), q(k)) \coloneqq \theta(k) \cdot J(x(k), \mathbf{u}(k), q(k)) + \xi \cdot \underline{\Gamma}(x(k), \mathbf{u}(k), q(k))$$

subject to :

$$\hat{x}(0|k) = x(k), \tag{3.13a}$$

$$\hat{x}(j|k) = f\left(\hat{x}(j-1|k), u(j-1|k), 0\right), \ \forall j \in [1, q(k)], \tag{3.13b}$$

$$\hat{x}(j|k) \in \mathcal{X} \ominus \mathcal{R}(j), \ \forall j \in [0, q(k)], \tag{3.13c}$$

$$(\mathbf{u}(k), q(k)) \in \mathcal{U}^{q(k)} \times \left\{1, \dots, N_p\right\} \tag{3.13d}$$

where:

- $\mathbf{u}(k) \coloneqq \{u(0|k), \dots, u(q(k)-1|k)\}$;
- $q(k)$ is the prediction horizon of the controller, here considered as a decision variable that can take values in $[1, N_p]$;
- $\theta(k)$ is an internal state of the MPC controller and is needed to guarantee the stability properties of the closed-loop system and the dynamics of which is defined later.

If multiple solutions providing the same cost exist, the one having the smallest q is kept. The optimal solutions of $P_{N_p}(x(k), \theta(k))$ are denoted $\mathbf{u}^*(x(k), \theta(k))$ and $q^*(x(k), \theta(k))$ (or $\mathbf{u}^*(k)$ and $q^*(k)$ for brevity) and the corresponding values $\underline{\Gamma}$, J, V_{N_p} and j_{opt} are:

$$\underline{\Gamma}^*(x(k), \theta(k)) \coloneqq \underline{\Gamma}(x(k), \mathbf{u}^*(k), q^*(k)),$$

$$J^*(x(k), \theta(k)) \coloneqq J(x(k), \mathbf{u}^*(k), q^*(k)),$$

$$V^*_{N_p}(x(k), \theta(k)) \coloneqq V_{N_p}(x(k), \theta(k), \mathbf{u}^*(k), q^*(k)),$$

$$j^*(x(k), \theta(k)) \coloneqq j_{\text{opt}}(x(k), \mathbf{u}^*(k), q^*(k)).$$

We also define $\hat{\mathbf{x}}^*(k) \coloneqq \{\hat{x}^*(0|k), \dots, \hat{x}^*(q^*(k)|k)\}$ as the nominal state sequence provided by the optimal control sequence $\mathbf{u}^*(x(k), \theta(k))$, i.e. $\hat{x}^*(j|k) = \phi(j; x(k), \mathbf{u}^*(k), \mathbf{0})$. The control law of the proposed MPC is the following: $\kappa_{N_p}(x(k), \theta(k)) \coloneqq u^*(0|k)$, where $u^*(0|k)$ is the first element of the optimal control sequence $\mathbf{u}^*(k)$ obtained by solving $P_{N_p}(x(k), \theta(k))$.
The initial condition of the controller state θ is set to $\theta(0) = \max\{\epsilon, \nu\Gamma(x(0))\}$, where $\epsilon > 0$ and $\nu \in (0, 1)$ are design parameters, and is updated at every time instant k by using the following policy:

$$\theta(k) = f_\theta\left(x(k), \theta(k-1)\right) \coloneqq \begin{cases} \theta(k-1), & \Gamma\left(x(k)\right) > \theta(k-1) \\ \max\{\epsilon, \nu\Gamma\left(x(k)\right)\}, & \text{otherwise} \end{cases} \tag{3.14}$$

Next, we want to prove that the equilibrium $x = 0$ is input-to-state practically stable (ISpS) for the closed-loop system under our robust contraction-based MPC. In order to do so, we demonstrate that $V^*_{N_p}(x(k), \theta(k))$ is an ISpS-Lyapunov function. The first step in this direction is proving that $V^*_{N_p}(x(k), \theta(k))$ is upper-bounded by a positive definite function of $x(k)$, as proved by Lemma 3.2, which is based on the result of Lemma 3.1.

Lemma 3.1: Generic upper bound on the cost function value

If Assumptions 3.1-3.7 are satisfied, then for all the feasible initial conditions $x(k) \in \mathcal{X}$, we have that

$$V^*_{N_p}(x(k), \theta(k)) \leq \theta(k) N_p \bar{\ell} + \xi \gamma \Gamma(x(k)), \tag{3.15}$$

and the minimum value of the map $\Gamma(\cdot)$ is obtained at the end of the trajectory, that is

$$j^*(x(k), \theta(k)) = q^*(x(k), \theta(k)), \tag{3.16}$$

where j^* is the horizon providing the maximum contraction.

Proof. Given that $P_{N_p}(x(k), \theta(k))$ is feasible, the stage cost is surely lower than $N_p\bar{\ell}$ thanks to Assumption 3.7, while $\underline{\Gamma}\big(x(k), \mathbf{u}(k), N_p\big) \leq \gamma\Gamma\big(x(k)\big)$ as a result of Assumption 3.3. This proves (3.15).

Concerning (3.16), it can be proved by contradiction. Suppose $j^*\big(x(k), \theta(k)\big) < q^*\big(x(k), \theta(k)\big)$; then, the candidate solution $\big(\mathbf{u}^c(k), q^c(k)\big) := \big(\big\{u^*(0|k), \ldots, u^*(j^*-1|k)\big\}, j^*\big(x(k), \theta(k)\big)\big)$, together with the related state trajectory $\hat{\mathbf{x}}^c(k) := \{\hat{x}^c(0|k), \ldots, \hat{x}^c(q^c|k)\}$, where $\hat{x}^c(j|k) := \phi(j; x(k), \mathbf{u}^c(k), \mathbf{0})$, would correspond to a cost function value satisfying

$$V_{N_p}(x(k), \theta(k), \mathbf{u}^c(k), q^c(k)) = \theta(k) \cdot \sum_{j=0}^{q^c(k)-1} \ell(\hat{x}^c(j|k), u^c(j|k)) + \xi \min_{j=1}^{q^c(k)} \Gamma(\hat{x}^c(j|k)).$$

Considering that

$$V^*_{N_p}(x(k), \theta(k)) = \theta(k) \cdot \sum_{j=0}^{q^*(k)-1} \ell(\hat{x}^*(j|k), u^*(j|k)) + \xi \min_{j=1}^{q^*(k)} \Gamma(\hat{x}^*(j|k)),$$

by optimality it holds that

$$\begin{aligned}
0 &\leq V_{N_p}(x(k), \theta(k), \mathbf{u}^c(k), q^c(k)) - V^*_{N_p}(x(k), \theta(k)) \\
&= \theta(k) \cdot \sum_{j=0}^{q^c(k)-1} \ell(\hat{x}^c(j|k), u^c(j|k)) + \xi \min_{j=1}^{q^c(k)} \Gamma(\hat{x}^c(j|k)) \\
&\quad - \theta(k) \cdot \sum_{j=0}^{q^*(k)-1} \ell(\hat{x}^*(j|k), u^*(j|k)) - \xi \min_{j=1}^{q^*(k)} \Gamma(\hat{x}^*(j|k)).
\end{aligned} \tag{3.17}$$

Provided that the instant of maximum contraction is $j^* \equiv q^c(k)$, we get that

$$\min_{j=1}^{q^c(k)} \Gamma\left(\hat{x}^c(j|k)\right) = \min_{j=1}^{q^*(k)} \Gamma\left(\hat{x}^*(j|k)\right).$$

Therefore, (3.17) reduces to

$$\begin{aligned} 0 &\leq V_{N_p}\left(x(k), \theta(k), \mathbf{u}^c(k), q^c(k)\right) - V^*_{N_p}\left(x(k), \theta(k)\right) \\ &= -\theta(k) \cdot \sum_{j=q^c(k)}^{q^*(k)-1} \ell\left(\hat{x}^*(j|k), u^*(j|k)\right) \leq 0, \end{aligned}$$

which implies

$$\sum_{j=q^c(k)}^{q^*(k)-1} \ell\left(\hat{x}^*(j|k), u^*(j|k)\right) = 0,$$

meaning that the desired equilibrium is reached after $q^c(k)$ prediction steps. This contradicts the optimality of $q^*(k)$. ■

Lemma 3.2: Existence of an upper-bounding function for the cost function

Under Assumptions 3.1-3.7, given a feasible initial condition $x(k) \in \mathcal{X}$, if the following conditions hold simultaneously:

1. $\theta(k) = f_\theta\left(x(k), \theta(k-1)\right)$,
2. $\xi \geq 2N_p\bar{\ell}/(1-\gamma)$,

then the optimal solution to $P_{N_p}(x(k), \theta(k))$ satisfies the inequality

$$V^*_{N_p}\left(x(k), \theta(k)\right) \leq \left[\frac{1+\gamma}{2}\right] \xi\Gamma\left(x(k)\right) + \epsilon N_p \bar{\ell}. \tag{3.18}$$

Proof. The conditions of Lemma 3.1 are satisfied. Therefore,

$$V^*_{N_p}\left(x(k), \theta(k)\right) \leq \theta(k) N_p \bar{\ell} + \xi\gamma\Gamma\left(x(k)\right).$$

To continue the proof, given that the update law $f_\theta(x(k), \theta(k-1))$ ensures that $\Gamma\left(x(k)\right) > \theta(k)$ for all $x \in \mathcal{X}$ such that $\Gamma(x) > \epsilon$, we need to discern two cases:

Case $\Gamma\left(x(k)\right) > \theta(k)$: Given that also $\Gamma\left(x(k)\right) > \theta(k)$ is satisfied, we get that

$$V^*_{N_p}\left(x(k), \theta(k)\right) \leq (N_p\bar{\ell} + \xi\gamma)\Gamma\left(x(k)\right).$$

Using the assumption $\xi \geq 2N_p\bar{\ell}/(1-\gamma)$, we obtain

$$V^*_{N_p}\left(x(k), \theta(k)\right) \leq \left[\frac{\gamma+1}{2}\right] \xi\Gamma\left(x(k)\right). \tag{3.19}$$

Case $\Gamma(x(k)) \leq \theta(k) = \epsilon$: Given that $\theta(k) = \epsilon$, (3.15) rewrites as

$$V^*_{N_p}(x(k), \theta(k)) \leq \epsilon N_p \bar{\ell} + \gamma \xi \Gamma(x(k)). \tag{3.20}$$

Unifying (3.19) and (3.20), we obtain

$$\begin{aligned} V^*_{N_p}(x(k), \theta(k)) &\leq \epsilon N_p \bar{\ell} + \xi \max\left\{\gamma, \frac{\gamma+1}{2}\right\} \Gamma(x(k)) \\ &\leq \epsilon N_p \bar{\ell} + \xi \left\lceil \frac{\gamma+1}{2} \right\rceil \Gamma(x(k)), \end{aligned}$$

which ends the proof. ■

Next, we prove the nominal descent property of $V^*_{N_p}(x(k), \theta(k))$ and the boundedness of its increase due to the uncertainty $w(k)$, discerning the cases where $q^*(x(k), \theta(k)) > 1$ (Lemma 3.3) and $q^*(x(k), \theta(k)) = 1$ (Lemma 3.4).

Lemma 3.3: Recursive feasibility and cost descent for $q^* > 1$

Given a feasible initial condition $x(k) \in \mathcal{X}$, under Assumptions 3.1-3.7, if the following conditions hold simultaneously:

1. $q^*(x(k), \theta(k)) > 1$,
2. $\theta(k) = f_\theta\big(x(k), \theta(k-1)\big)$,

then $P_{N_p}(x(k+1), \theta(k+1))$ is feasible, and there exists a $\mathcal{K}$ function $\lambda_1(\cdot)$ such that

$$\begin{aligned} &V_{N_p}\big(x(k+1), \theta(k+1), \mathbf{u}^+(k+1), q^+(k+1)\big) - V^*_{N_p}(x(k), \theta(k)) \\ &\leq -\theta(k)\alpha_\ell(\|x(k)\|) + (\theta(k) + \xi)\, \lambda_1(\|w(k)\|), \end{aligned} \tag{3.21}$$

where $q^+(k+1) := q^*(k) - 1$ and $\mathbf{u}^+(k+1) := \big\{u^*(1|k), \ldots, u^*(q^*(k) - 1|k)\big\}$.

Proof. **Recursive feasibility:** Given that $q^*(k) > 1$, $|x_i(k+1) - \hat{x}^*_i(1|k)| \leq \bar{w}_i$, $\forall i \in [1, n]$, that $u^+(j-1|k+1) = u^*(j|k)$, and that Assumptions 3.4 and 3.5 hold, we get that, $\forall j \in [1, q^+(k+1)]$,

$$\begin{aligned} \phi\big(j; x(k+1), \mathbf{u}^+(k+1), \mathbf{0}\big) &\in \phi\big(j; \hat{x}^*(1|k), \mathbf{u}^+(k+1), \mathbf{0}\big) \oplus \mathcal{F}(j) \\ &\subseteq \phi\big(j; \hat{x}^*(1|k), \mathbf{u}^+(k+1), \mathbf{0}\big) \oplus \mathcal{R}(j+1) \ominus \mathcal{R}(j). \end{aligned}$$

As $\phi(j; \hat{x}^*(1|k), \mathbf{u}^+(k+1), \mathbf{0}) \in \mathcal{X} \ominus \mathcal{R}(j+1)$, we eventually obtain that

$$\begin{aligned} \phi\big(j; x(k+1), \mathbf{u}^+(k+1), \mathbf{0}\big) &\in \phi\big(j; \hat{x}^*(1|k), \mathbf{u}^+(k+1), \mathbf{0}\big) \oplus \mathcal{R}(j+1) \ominus \mathcal{R}(j) \\ &\subseteq (\mathcal{X} \ominus \mathcal{R}(j+1)) \oplus \mathcal{R}(j+1) \ominus \mathcal{R}(j) \\ &\subseteq \mathcal{X} \ominus \mathcal{R}(j), \end{aligned}$$

which proves that the couple $(\mathbf{u}^+(k+1), q^+(k+1))$ is feasible.

Cost descent: The cost related to the admissible pair $(\mathbf{u}^+(k+1), q^+(k+1))$ is given by

$$\begin{aligned}
&V_{N_p}\left(x(k+1), \theta(k+1), \mathbf{u}^+(k+1), q^+(k+1)\right) \\
&= \theta(k+1) \cdot \sum_{j=0}^{q^+(k+1)-1} \ell\left(\hat{x}^+(j|k+1), u^+(j|k+1)\right) \\
&+ \xi \underline{\Gamma}\left(x(k+1), \mathbf{u}^+(k+1), q^+(k+1)\right).
\end{aligned}$$

Since $\underline{\Gamma}\left(x(k+1), \mathbf{u}^+(k+1), q^+(k+1)\right) \le \Gamma\left(\hat{x}^+(j|k+1)\right)$ for all $j \in [1, q^+(k+1)]$, we get:

$$\begin{aligned}
&V_{N_p}\left(x(k+1), \theta(k+1), \mathbf{u}^+(k+1), q^+(k+1)\right) \\
&\le \theta(k+1) \cdot \sum_{j=0}^{q^+(k+1)-1} \ell\left(\hat{x}^+(j|k+1), u^+(j|k+1)\right) + \xi\Gamma\left(\hat{x}^+(q^+|k+1)\right).
\end{aligned}$$

Evaluating $V_{N_p}\left(x(k+1), \theta(k+1), \mathbf{u}^+(k+1), q^+(k+1)\right) - V^*_{N_p}\left(x(k), \theta(k)\right)$. Taking into account that, in virtue of Lemma 3.2, $q^*(x(k), \theta(k)) = j^*(x(k), \theta(k))$, we also have that $\underline{\Gamma}\left(x(k), \mathbf{u}^*(k), q^*(k)\right) = \Gamma\left(\hat{x}^*(q^*(k)|k)\right)$, therefore:

$$\begin{aligned}
&V_{N_p}\left(x(k+1), \theta(k+1), \mathbf{u}^+(k+1), q^+(k+1)\right) - V^*_{N_p}\left(x(k), \theta(k)\right) \\
&\le \theta(k+1) \cdot \sum_{j=0}^{q^+(k+1)-1} \ell\left(\hat{x}^+(j|k+1), u^+(j|k+1)\right) + \xi\Gamma\left(\hat{x}^+(q^+(k+1)|k+1)\right) \\
&- \theta(k) \cdot \sum_{j=0}^{q^*(k)-1} \ell\left(\hat{x}^*(j|k), u^*(j|k)\right) - \xi\Gamma\left(\hat{x}^*(q^*(k)|k)\right).
\end{aligned}$$

Since $\theta(k+1) = f_\theta\left(x(k+1), \theta(k)\right)$ and $\theta(k) > 0$, we have that $\theta(k+1) \le \theta(k)$; for this reason, we obtain

$$\begin{aligned}
&V_{N_p}\big(x(k+1), \theta(k+1), \mathbf{u}^+(k+1), q^+(k+1)\big) - V^*_{N_p}\big(x(k), \theta(k)\big) \\
&\le \theta(k) \cdot \sum_{j=0}^{q^+(k+1)-1} \ell\big(\hat{x}^+(j|k+1), u^+(j|k+1)\big) + \xi\Gamma\big(\hat{x}^+(q^+(k+1)|k+1)\big) \\
&- \theta(k) \cdot \sum_{j=0}^{q^*(k)-1} \ell\left(\hat{x}^*(j|k), u^*(j|k)\right) - \xi\Gamma\left(\hat{x}^*(q^*(k)|k)\right) \\
&= -\theta(k)\ell\left(x(k), u^*(k)\right) + \xi\Big[\Gamma\left(\hat{x}^+(q^+(k+1)|k+1)\right) - \Gamma\left(\hat{x}^*(q^*(k)|k)\right)\Big] \\
&+ \theta(k) \cdot \sum_{j=1}^{q^*(k)-1} \left[\ell\left(\hat{x}^+(j-1|k+1), u^+(j-1|k+1)\right) - \ell\big(\hat{x}^*(j|k), u^*(j|k)\big)\right].
\end{aligned}$$

Because of Assumptions 3.3 and 3.7, and due to the fact that $u^+(j-1|k+1) \equiv u^*(j|k)$, the last inequality becomes

$$\begin{aligned}
&V_{N_p}\left(x(k+1), \theta(k+1), \mathbf{u}^+, q^+\right) - V^*_{N_p}\left(x(k), \theta(k)\right) \\
&\leq -\theta(k)\alpha_\ell\left(\|x(k)\|\right) + \xi\sigma_\Gamma\left(\left\|\hat{x}^+(q^*-1|k+1) - \hat{x}^*(q^*|k)\right\|\right) \\
&+ \theta(k) \cdot \sum_{j=1}^{q^*-1} \sigma_{\ell,x}\left(\left\|\hat{x}^+(j-1|k+1) - \hat{x}^*(j|k)\right\|\right).
\end{aligned}$$

Analyzing the term $\|\hat{x}^+(j-1|k+1) - \hat{x}^*(j|k)\|$. In particular, note that

$$\|\hat{x}^+(j-1|k+1) - \hat{x}^*(j|k)\| = \left\|\begin{matrix} \hat{x}_1^+(j-1|k+1) - \hat{x}_1^*(j|k) \\ \vdots \\ \hat{x}_n^+(j-1|k+1) - \hat{x}_n^*(j|k) \end{matrix}\right\|.$$

From the component-wise uniform continuity of the model, we have that, for all $i \in [1, n]$,

$$|\hat{x}_i^+(0|k+1) - \hat{x}_i^*(1|k)| \leq \sum_{a=1}^{r} \sigma_{w,ia}(w_a(k)) =: c_{i,0}(w(k)).$$

In the same way,

$$\begin{aligned}
|\hat{x}_i^+(1|k+1) - \hat{x}_i^*(2|k)| &\leq \sum_{b=1}^{n} \sigma_{x,ib}\left(\sum_{a=1}^{r} \sigma_{w,ba}(w_a(k))\right) \\
&= \sum_{b=1}^{n} \sigma_{x,ib}(c_{b,0}(w(k))) =: c_{i,1}(w(k)).
\end{aligned}$$

Generalizing for all $j > 0$,

$$|\hat{x}_i^+(j|k+1) - \hat{x}_i^*(j+1|k)| \leq \sum_{b=1}^{n} \sigma_{x,ib}\left(c_{b,j-1}(w(k))\right) =: c_{i,j}(w(k)).$$

Therefore,

$$\|\hat{x}^+(j|k+1) - \hat{x}^*(j+1|k)\| \leq \left\|\begin{matrix} c_{1,j}(w(k)) \\ \vdots \\ c_{n,j}(w(k)) \end{matrix}\right\| =: r_j(w(k)). \tag{3.22}$$

Note that the functions $r_j(w(k))$ are positive definite, strictly increasing with respect to $\|w(k)\|$, and they tend to infinity for $\|w(k)\| \to \infty$; for this reason, it can be concluded that they are $\mathcal{K}_\infty$ functions. Consequently, we obtain

$$\begin{aligned}
&V_{N_p}\left(x(k+1), \theta(k+1), \mathbf{u}^+(k+1), q^+(k+1)\right) - V^*_{N_p}\left(x(k), \theta(k)\right) \\
&\leq -\theta(k)\alpha_\ell\left(\|x(k)\|\right) + \xi\sigma_\Gamma\left(r_{q^*(k)-1}\left(w(k)\right)\right) + \theta(k) \cdot \sum_{j=1}^{q^*(k)-1} \sigma_{\ell,x}\left(r_{j-1}\left(w(k)\right)\right).
\end{aligned}$$

Provided that there exists a $\mathcal{K}$ function $\lambda_1(\cdot)$ such that the following conditions hold simultaneously:

1. $\lambda_1(\|w\|) \geq \sum_{j=1}^{q^*(k)-1} \sigma_{\ell,x}\left(r_{j-1}(w)\right)$,

2. $\lambda_1(\|w\|) \geq \sigma_\Gamma\left(r_{q^*(k)-1}(w)\right)$,

the last inequality becomes

$$\begin{aligned}&V_{N_p}\left(x(k+1), \theta(k+1), \mathbf{u}^+(k+1), q^+(k+1)\right) - V^*_{N_p}\left(x(k), \theta(k)\right)\\ &\leq -\theta(k)\alpha_\ell\left(\|x(k)\|\right) + \left(\theta(k)+\xi\right)\lambda_1\left(\|w(k)\|\right).\end{aligned}$$

■

Before introducing the next lemma, we define $\omega > 0$ as a constant such that $\{x : \Gamma(x) \leq \omega\} \subseteq \Omega \ominus \mathcal{R}(1)$.

Lemma 3.4: Recursive feasibility and cost descent for $q^* = 1$

Under Assumptions 3.1-3.7, given a feasible initial condition $x(k) \in \mathcal{X}$, if the following conditions hold simultaneously:

1. $q^*\left(x(k), \theta(k)\right) = 1$,

2. $\theta(k) = f_\theta\big(x(k), \theta(k-1)\big)$,

3. $\xi \geq 2N_p\bar{\ell}/(1-\gamma)$,

4. $\gamma \leq \frac{\omega}{\Gamma_{\max}}$, *where* $\Gamma_{\max} := \max_{x\in\mathcal{X}} \Gamma(x)$,

then $P_{N_p}\left(x(k+1), \theta(k+1)\right)$ is feasible, and there exists a $\mathcal{K}$ function $\lambda_2(\cdot)$ such that the following inequality holds true:

$$\begin{aligned}&V^*_{N_p}\left(x(k+1), \theta(k+1)\right) - V^*_{N_p}\left(x(k), \theta(k)\right) \leq -\theta(k)\alpha_\ell\left(\|x(k)\|\right)\\ &+ \lambda_2\left(\|w(k)\|\right) + \epsilon N_p\bar{\ell}.\end{aligned} \tag{3.23}$$

Proof. **Recursive feasibility:** Given that $q^* = 1$, we have that

$$\Gamma(\hat{x}^*(1|k)) \leq \gamma\Gamma(x(k)) \leq \omega\frac{\Gamma(x(k))}{\Gamma_{\max}} \leq \omega.$$

Therefore, $\hat{x}^*(1|k) \in \Omega \ominus \mathcal{R}(1)$, which implies $x(k+1) \in \hat{x}^*(1|k) \oplus \mathcal{R}(1) \subset \Omega$.

Since, thanks to Assumption 3.6, $\Omega \subseteq \mathcal{Q}\left(\mathcal{X} \ominus \mathcal{R}(1)\right)$ for the nominal system, there exists $u^+(0|k+1) \in \mathcal{U}$ that is able to drive the nominal system state to $\mathcal{X} \ominus \mathcal{R}(1)$. Therefore, a feasible solution to $P_{N_p}(x(k+1), \theta(k+1))$ is $(q^+(k+1), \mathbf{u}^+(k+1)) = (1, \{u^+(0|k+1)\})$.

Cost descent: Since $q^*(k) = 1$ and by virtue of the receding horizon implementation, we have

$$\begin{aligned} V^*_{N_p}\left(x(k), \theta(k)\right) &= \theta(k)\ell\left(x(k), u^*(k)\right) + \xi\Gamma\left(\hat{x}^*(1|k)\right) \\ &\geq \theta(k)\alpha_\ell\left(\|x(k)\|\right) + \xi\Gamma\left(\hat{x}^*(1|k)\right). \end{aligned}$$

Since the conditions of Lemma 3.2 are satisfied at time instant $k+1$, we have

$$\begin{aligned} V^*_{N_p}\left(x(k+1), \theta(k+1)\right) &\leq \left[\frac{1+\gamma}{2}\right]\xi\Gamma\left(x(k+1)\right) + \epsilon N_p\bar{\ell} \\ &< \xi\Gamma\left(x(k+1)\right) + \epsilon N_p\bar{\ell}. \end{aligned}$$

Putting together the last two inequalities, we get

$$\begin{aligned} V^*_{N_p}\left(x(k+1), \theta(k+1)\right) - V^*_{N_p}\left(x(k), \theta(k)\right) &\leq \xi\Gamma\left(x(k+1)\right) - \theta(k)\alpha_\ell\left(\|x(k)\|\right) \\ &\quad - \xi\Gamma\left(\hat{x}^*(1|k)\right) + \epsilon N_p\bar{\ell}. \end{aligned}$$

Due to the uniform continuity of $\Gamma(\cdot)$ over $\mathcal{X}$, this becomes:

$$\begin{aligned} V^*_{N_p}\left(x(k+1), \theta(k+1)\right) - V^*_{N_p}\left(x(k), \theta(k)\right) &\leq -\theta(k)\alpha_\ell\left(\|x(k)\|\right) \\ &\quad + \xi\sigma_\Gamma\left(\|x(k+1) - \hat{x}^*(1|k)\|\right) + \epsilon N_p\bar{\ell}. \end{aligned}$$

Given that, as stated in the proof of Lemma 3.3, $\|x(k+1) - \hat{x}^*(1|k)\| \leq r_0(w(k))$, we obtain

$$\begin{aligned} &V^*_{N_p}\left(x(k+1), \theta(k+1)\right) - V^*_{N_p}\left(x(k), \theta(k)\right) \leq -\theta(k)\alpha_\ell\left(\|x(k)\|\right) \\ &+ \xi\sigma_\Gamma\left(r_0\left(w(k)\right)\right) + \epsilon N_p\bar{\ell}. \end{aligned}$$

We now exploit a $\mathcal{K}$ function $\lambda_2(\cdot)$ such that $\lambda_2(\|w\|) \geq \xi\sigma_\Gamma\left(r_0\left(w\right)\right)$ to obtain the final inequality:

$$\begin{aligned} V^*_{N_p}\left(x(k+1), \theta(k+1)\right) - V^*_{N_p}\left(x(k), \theta(k)\right) &\leq -\theta(k)\alpha_\ell\left(\|x(k)\|\right) \\ &\quad + \lambda_2\left(\|w(k)\|\right) + \epsilon N_p\bar{\ell}. \end{aligned}$$

■

We are now ready to present the main theorem of the work, which proves input-to-state practical stability of the closed loop and recursive feasibility of the robust contraction-based MPC.

Theorem 3.1: Stability of the robust contraction-based MPC for the regulation of nonlinear systems

Under Assumptions 3.1-3.7, if the following conditions hold simultaneously:

1. $\xi \geq 2N_p\bar{\ell}/(1-\gamma)$,

2. $\theta(k) = f_\theta\left(x(k), \theta(k-1)\right)$,

3. $\gamma \leq \frac{\omega}{\Gamma_{\max}}$,

then, given a feasible initial condition $x \in \mathcal{X}$, $P_{N_p}(x, \theta)$ is recursively feasible and the equilibrium $x = 0$ is ISpS for the closed-loop system.

Proof. : The proof is divided in two parts; we first prove recursive feasibility and then prove input-to-state practical stability.

Recursive feasibility: Lemmas 3.3 and 3.4, prove recursive feasibility for all $q^*(k) \in [1, N_p]$.

Input-to-state stability: Lemma 3.2 proves that

$$V^*_{N_p}(x(k), \theta(k)) \leq \xi \Gamma(x(k)) \leq \xi \alpha_\Gamma(\|x\|) + \epsilon N_p \bar{\ell},$$

and, due to the structure of the cost function, we have that

$$V^*_{N_p}(x(k), \theta(k)) \geq \theta(k)\ell(x, u) \geq \theta(k)\alpha_\ell(\|x(k)\|) \geq \epsilon \alpha_\ell(\|x(k)\|),$$

as $\theta(k) \geq \epsilon$.

In the following, we prove that $V^*_{N_p}(x(k), \theta(k))$ has the nominal descent property of typical ISpS-Lyapunov functions, and that the possible rises in its value due to the uncertainty $w(k)$ are bounded. To this end, note that, starting from a feasible initial condition $x(0)$ and imposing $\theta(0) = \max\{\epsilon, \nu\Gamma(x(0))\}$, employing the update rule $\theta(k) = f_\theta(x(k), \theta(k-1))$ and setting $\xi \geq 2N_p\bar{\ell}/(1-\gamma)$ ensure that, under Assumptions 3.3-3.6, all the conditions of Lemmas 3.3 and 3.4 are met.

Lemma 3.3 states that, if $q^*(k) > 1$, then the following inequality holds:

$$\begin{aligned} &V_{N_p}\left(x(k+1), \theta(k+1), \mathbf{u}^+(k+1), q^+(k+1)\right) - V^*_{N_p}(x(k), \theta(k)) \\ &\leq -\theta(k)\alpha_\ell(\|x(k)\|) + (\theta(k) + \xi)\lambda_1(\|w(k)\|), \end{aligned}$$

where $q^+(k+1) := q^*(k) - 1$, $\mathbf{u}^+(k+1) := \left\{u^*(1|k), \ldots, u^*(q^*(k)-1|k)\right\}$ and $\lambda_1(\cdot)$ is a $\mathcal{K}$ function. Because of the suboptimality of $V_{N_p}\left(x(k+1), \theta(k+1), \mathbf{u}^+(k+1), q^+(k+1)\right)$, it also holds true that

$$\begin{aligned} V^*_{N_p}(x(k+1), \theta(k+1)) - V^*_{N_p}(x(k), \theta(k)) &\leq -\theta(k)\alpha_\ell(\|x(k)\|) \\ &+ (\theta(k) + \xi)\lambda_1(\|w(k)\|). \end{aligned} \tag{3.24}$$

Lemma 3.4 states that, if $q^*(k) = 1$, then

$$\begin{aligned} V^*_{N_p}(x(k+1), \theta(k+1)) - V^*_{N_p}(x(k), \theta(k)) &\leq -\theta(k)\alpha_\ell(\|x(k)\|) \\ &+ \lambda_2(\|w(k)\|) + \epsilon N_p \bar{\ell}. \end{aligned} \tag{3.25}$$

In order to combine (3.24) and (3.25), we make the following consideration. Given that $\theta(0) = \max\{\epsilon, \Gamma(x(0))\}$ and $\theta(k) = f_\theta(x(k), \theta(k-1))$, we have that $\theta(k) \leq \theta(0)$. For this reason,

$$(\theta(k) + \xi)\,\lambda_1\,(\|w(k)\|) \leq (\theta(0) + \xi)\,\lambda_1\,(\|w(k)\|)\,.$$

We now employ a $\mathcal{K}$ function $\lambda(\cdot)$ such that $\lambda(\|w\|) \geq (\theta(0) + \xi) \cdot \lambda_1(\|w\|)$ and $\lambda(\|w\|) \geq \lambda_2(\|w\|)$, so as to obtain that, for $q^*(k) \geq 1$,

$$\begin{aligned} V^*_{N_p}\,(x(k+1), \theta(k+1)) - V^*_{N_p}\,(x(k), \theta(k)) \leq -\theta(k)\alpha_\ell\,(\|x(k)\|) \\ + \lambda\,(\|w(k)\|) + \epsilon N_p \bar{\ell}. \end{aligned} \tag{3.26}$$

Given that $\theta(k) > 0$ for all $k \geq 0$, (3.26), together with the lower and upper bounds of $V^*_{N_p}(x(k), \theta(k))$, proves that $V^*_{N_p}(x(k), \theta(k))$ is a time-varying ISpS-Lyapunov function, whose variability over time is given by $\theta(k)$. The existence of a time-varying ISpS-Lyapunov function allows us to state that the equilibrium $x = 0$ is ISpS for the closed-loop system. ■

Remark 3.4

It is important to note that the input-to-state practical stability property ensured by Theorem 3.1 is very close to input-to-state stability. Indeed, $V^*_{N_p}(\cdot,\cdot)$ is an ISpS-Lyapunov function because in its upper bound function and in its descent inequality some positive constant terms appear, but these constant terms all depend on ϵ, which can be arbitrarily set to very small values. Therefore, setting ϵ to values that are close to zero, $V^*_{N_p}(\cdot,\cdot)$ practically behaves as an ISS-Lyapunov function.

3.6 A robust contraction-based MPC formulation without integer decision variables

The MPC formulation introduced in (3.13) involves the integer variable q, which could make the resolution of the problem computationally demanding. In this section, an alternative two-stage algorithm without integer decision variables is introduced.

The two steps of the newly proposed algorithm are the following:

1. First, the following fixed-horizon problem is solved:

$$\begin{aligned}
&\min_{\mathbf{u}(k)} \underline{\Gamma}(x(k), \mathbf{u}(k), N_p) \\
&\text{ s.t. } \hat{x}(0|k) = x(k), \\
&\qquad \hat{x}(j|k) = f\left(\hat{x}(j-1|k), u(j-1|k), 0\right), \forall j \in [1, N_p], \\
&\qquad \hat{x}(j|k) \in \mathcal{X} \ominus \mathcal{R}(j),\ \forall j \in [0, N_p], \\
&\qquad \mathbf{u}(k) \in \mathcal{U}^{N_p}
\end{aligned} \tag{3.27}$$

and the instant $j_{N_p} \in [1, N_p]$ when the maximum contraction occurs is memorized.

2. A second optimization problem with prediction horizon equal to j_{N_p} is solved to get the desired optimal control sequence:

$$\begin{aligned}
&\min_{\mathbf{u}(k)} V_{N_p}\left(x(k), \theta(k), \mathbf{u}(k), j_{N_p}\right) \\
&\text{ s.t. } \hat{x}(0|k) = x(k), \\
&\qquad \hat{x}(j|k) = f\left(\hat{x}(j-1|k), u(j-1|k), 0\right), \forall j \in [1, j_{N_p}], \\
&\qquad \hat{x}(j|k) \in \mathcal{X} \ominus \mathcal{R}(j),\ \forall j \in [0, j_{N_p}], \\
&\qquad \mathbf{u}(k) \in \mathcal{U}^{j_{N_p}}.
\end{aligned} \tag{3.28}$$

Remark 3.5

In order to limit the increment in the computational cost provided by this formulation, the first j_{N_p} elements of the solution of problem (3.27) *could be used as feasible initial solution for problem* (3.28).

This formulation inherits all the stability and robustness properties of the one introduced before, as proved by the next theorem.

Theorem 3.2: Stability of the two-stage formulation of the robust contraction-based MPC for the regulation of nonlinear systems

If the following conditions hold:

1. *Assumptions 3.1-3.7 are satisfied;*
2. *the penalty ξ involved in the second optimization problem satisfies $\xi \geq 2N_p\bar{\ell}/(1-\gamma)$;*
3. *the rule $\theta(k) = f_\theta(x(k), \theta(k-1))$ is used to update the value of θ,*
4. $\gamma \leq \frac{\omega}{\Gamma_{\max}}$,

then, given a feasible initial condition $x \in \mathcal{X}$, the 2-stage version of $P_{N_p}(x, \theta)$ is recursively feasible and the equilibrium $x = 0$ is ISpS for the closed-loop system.

Proof. At each decision instant k, the first stage of the algorithm computes the prediction horizon $q^*(k)$ that corresponds to the maximum contraction instant, and the control sequence that leads to such contraction remains available for the second stage. For this reason, every candidate sequence that gives the maximum contraction in the two-stage formulation also verifies all the conditions of Theorem 3.1, which implies that the equilibrium $x = 0$ is ISpS for the closed-loop system also under the two-stage formulation of the robust contraction-based MPC. ∎

Remark 3.6

It is important to note that the two-stage formulation of the MPC proposed in this chapter can provide sub-optimal solutions. Nonetheless, in contexts where computational complexity is more important than optimality, the two-stage formulation represents the best choice.

3.7 Supporting algorithms for the design of the controller

In the current section, useful suggestions and algorithms for the computation of the ingredients of the robust contraction-based MPC are provided.

In case the system under control were component-wise Lipschitz continuous, the computation of the component-wise Lipschitz constants can be carried out relying on the method proposed in Section 2.7.2, with the only differences residing in the fact that the optimization problem to be solved does not have y_s as a decision variable and does not consider any pre-stabilizing control laws $\pi(\cdot, \cdot, \cdot)$. Once the constants are computed, the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ can be computed as in Section 2.7.3.

Nonetheless, differently from the MPC presented in Chapter 2, the currently analyzed robust contraction-based MPC requires also to choose a candidate contractive function and to find the contraction factor γ and the maximum prediction horizon N_p. A method for designing all these ingredients is provided next.

3.7.1 Choice of the contractive function and computation of the maximum prediction horizon

While from a theoretical point of view the choice of the best contractive function candidate is still a subject to be investigated, what we intuitively suggest to do is to choose a positive definite function that could maximize the volume of the sublevel set $\{\Gamma(x) \leq \omega\}$, which must lie inside the set

$\Omega \ominus \mathcal{R}(1) \equiv \mathcal{X} \ominus \mathcal{R}(1)$. The intuition behind this idea is that by maximizing the volume of such set we could potentially obtain large values of ω, and consequently get a larger upper bound for the contraction factor γ within the horizon $[1, N_p^*]$, where N_p^* is the maximum prediction horizon value that the designer is willing to accept.

Remark 3.7

The last consideration does not need to be true in every situation. In fact, given that the upper bound of γ is computed as $\omega/\Gamma_{\max}$, where $\Gamma_{\max}$ is the maximum value that is taken by $\Gamma(x)$ in $\mathcal{X}$, the shape and the volume of the sets $\mathcal{X}$ and Ω have to be taken into account to choose a suitable contractive function $\Gamma(x)$. In addition, as the contractivity of a candidate function depends also on the dynamics of the system, it might also happen that the candidate function $\Gamma(x)$ maximizing the volume of $\{\Gamma(x) \leq \omega\}$ actually results in a smaller upper bound for γ, due to the dynamics of the system not favouring the descent of $\Gamma(x)$.

In any case, should a contractive function $\Gamma(x)$ lead to an unsatisfying contraction factor γ within the horizon $[1, N_p^*]$, it is always possible to use a state constraint set $\hat{\mathcal{X}} \subset \mathcal{X}$ that can be shaped to reduce the value of $\Gamma_{\max}$. This would reduce the space where the system state is allowed to move, but would keep the stabilizing properties of the here-proposed robust contraction-based MPC.

After choosing a candidate contractive function, the real contraction factor γ can be computed as follows:

1. a grid of points in $\mathcal{X}$ is created;

2. for each $\hat{N}_p \in [1, N_p^*]$ and for each sample $x \in \mathcal{X}$, the punctual contraction factor $\gamma(\hat{N}_p, x)$ is computed by solving the following optimization problem:

$$\begin{aligned}
&\min_{\mathbf{u}} \Gamma\left(\hat{x}(\hat{N}_p)\right) / \Gamma\left(\hat{x}(0)\right) \\
&\text{s.t. } \hat{x}(0) = x, \\
&\quad \hat{x}(j) = f\left(\hat{x}(j-1), u(j-1), 0\right), \forall j \in [1, \hat{N}_p], \\
&\quad \mathbf{u} \in \mathcal{U}^{\hat{N}_p}
\end{aligned}$$

3. for each $\hat{N}_p \in [1, N_p^*]$, the respective contraction factor $\gamma(\hat{N}_p)$ is computed as

$$\gamma(\hat{N}_p) = \max_{x \in \mathcal{X}} \gamma(\hat{N}_p, x);$$

4. the maximum prediction horizon of the MPC is chosen as

$$N_p = \min_{\hat{N}_p \in [1, N_p^*]} \hat{N}_p \text{ s.t.} \gamma(\hat{N}_p) \leq \omega / \Gamma_{\max};$$

5. the final contraction factor is set as $\gamma = \gamma(N_p)$.

Remark 3.8

Note that the optimization problem that is solved for each prediction horizon and for each sample x does not impose the predicted states to belong to the set $\mathcal{X}$. This is because the theory presented in this chapter does not require such condition to hold, as it ensures recursive feasibility starting from a feasible initial condition and does not try to give an analytical formulation of the feasibility set. Of course, this means that it is possible to find a contraction factor γ verifying all the conditions listed in the previous sections and have no initial conditions that make the robust contraction-based MPC feasible.

3.8 Case studies

In order to validate the robust contraction-based MPC, we employ it to regulate two different systems and check its effectiveness. In particular, the aims of the following case studies are two: (i) proving how the MPC can control systems for which common methodologies for designing the terminal ingredients are ineffective, and (ii) proving how the MPC can also be implemented starting from terminal ingredients that are designed following standard state-of-the-art techniques.

In detail, we first control a perturbed nonholonomic system, and then control an uncertain quadruple-tank system.

3.8.1 Control of a perturbed nonholonomic system

We here consider a modified version of the illustrative example provided in [5]. In detail, the aim is to control the following perturbed discrete-time nonholonomic system to the origin starting from the initial condition $x(0) = (-4, 10, 4)$:

$$x_1(k+1) = x_1(k) + (1 + w(k))\, u_1(k), \tag{3.29a}$$

$$x_2(k+1) = x_2(k) + u_2(k), \tag{3.29b}$$

$$x_3(k+1) = x_3(k) + x_1(k) u_2(k). \tag{3.29c}$$

In detail, $x := (x_1, x_2, x_3) \in \mathbb{R}^3$ is the state of the system, $u := (u_1, u_2) \in \mathbb{R}^2$ is the input, and w is a perturbation that is assumed to be bounded. In particular, $w \in \mathcal{W} := [-0.025, 0.025]$. In the following, we often refer to this system using the generic notation $x(k+1) = f(x(k), u(k), w(k))$.

The state and the input of the system are subject to the following constraints:

$$x(k) \in \mathcal{X} \coloneqq \left\{x \in \mathbb{R}^3 :\ |x_1| \leq 4, |x_2| \leq 10,\ |x_3| \leq 10\right\},\ \forall k \geq 0,$$

$$u(k) \in \mathcal{U} \coloneqq \left\{u \in \mathbb{R}^2 :\ |u_1| \leq 8, |u_2| \leq 0.5\right\},\ \forall k \geq 0.$$

Before proceeding with the design of the robust contraction-based MPC, consider why the design of a standard MPC equipped with a terminal cost function and a terminal inequality constraint would be challenging. As the classical methods to design such terminal ingredients require to linearize the nominal system, i.e. setting $w = 0$, at the reference equilibrium, we proceed by linearizing the system at the origin, obtaining the following state and input matrices:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

The previous matrices clearly show that the last state of the linearized system is uncontrollable and such that it remains constantly equal to its initial value. Therefore, it is impossible to resort to the method presented in Section 1.1.1 for the computation of a control law $\kappa_f(\cdot)$, a terminal cost function $V_f(\cdot)$ and a terminal region $\mathcal{X}_f$ fulfilling all the conditions that are needed for building a recursively feasible and stabilizing QIH-MPC.

In cases like this, one could consider to act differently and opt for a regulation TEC-MPC, which makes it possible to ensure the stability of the closed loop also without knowing a locally stabilizing control law and a local CLF. Nonetheless, as outlined in Section 1.1.2, in order for a TEC-MPC to be correctly implemented the system under control must be controllable to the origin in a finite number of steps. Moreover, implementing a tube-based MPC with terminal equality constraint is quite unusual. In fact, ensuring the feasibility of a TEC-MPC often requires to have longer prediction horizons, but at the same time tube-based MPCs need short prediction horizons, as the sequence of tightened constraints that they employ to ensure robustness could lead to an empty feasible set for longer prediction horizons.

For these reasons, the robust contraction-based MPC is here employed to show its effectiveness. In the following, (i) the computation of the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$, (ii) the computation of the RPI Ω, (iii) the consequent choice of the contractive function and the maximum prediction horizon, and (iv) the choice of the MPC settings are discussed. Finally, the simulation results are shown.

Computation of the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$

Before delving into the technical details behind the computation of the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$, we first note that system (3.29) is Lipschitz in the compact set $\mathcal{X} \times \mathcal{U} \times \mathcal{W}$. However, as reported in [102], the Lipschitz continuity in a compact set implies also component-wise Lipschitz continuity in the same set, namely there exist non-negative real constants $L_{x,ia}$, $L_{u,ib}$ and $L_{w,i}$, where $i, a \in \{1, 2, 3\}$, $b \in \{1, 2\}$, such that

$$|f_i(x, u, w) - f_i(\check{x}, \check{u}, \check{w})| \leq \sum_{a=1}^{3} L_{x,ia}|x_a - \check{x}_a| + \sum_{b=1}^{2} L_{u,ib}|u_b - \check{u}_b| + L_{w,i}|w - \check{w}|. \tag{3.30}$$

In particular, the values of the constants $L_{x,ia}$, $L_{u,ib}$ and $L_{w,i}$ are respectively reported in Tables 8, 9 and 10.

$L_{x,ia}$		a		
		1	*2*	*3*
i	*1*	1	0	0
	2	0	1	0
	3	0.5	0	1

Table 8: Values of the constants $L_{x,ia}$ of the nonholonomic system.

$L_{u,ib}$		b	
		1	*2*
i	*1*	1.025	0
	2	0	1
	3	0	4

Table 9: Values of the constants $L_{u,ib}$ of the nonholonomic system.

$L_{w,1}$	$L_{w,2}$	$L_{w,3}$
8	0	0

Table 10: Values of the constants $L_{w,i}$ of the nonholonomic system.

Provided that component-wise Lipschitz continuity is a stronger condition than component-wise uniform continuity, the robust contraction-based MPC can be applied to system (3.29). Moreover, the component-wise Lipschitz continuity of the system makes it possible to exploit the algorithm presented in Section 2.7.3 for the computation of the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$. As the computed sets are box-shaped, i.e.

$$\mathcal{F}(j) = \{x \in \mathbb{R}^3 : \ |x_i| \leq \bar{x}_{i,\mathcal{F}(j)}, \forall i \in \{1, 2, 3\}\}$$

and

$$\mathcal{R}(j) = \{x \in \mathbb{R}^3 : \ |x_i| \leq \bar{x}_{i,\mathcal{R}(j)}, \forall i \in \{1, 2, 3\}\},$$

	j	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
$\mathcal{F}(j)$	$\bar{x}_{1,\mathcal{F}(j)}$	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
	$\bar{x}_{2,\mathcal{F}(j)}$	0	0	0	0	0	0	0	0	0	0	0
	$\bar{x}_{3,\mathcal{F}(j)}$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$\mathcal{R}(j)$	$\bar{x}_{1,\mathcal{R}(j)}$	0	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
	$\bar{x}_{2,\mathcal{R}(j)}$	0	0	0	0	0	0	0	0	0	0	0
	$\bar{x}_{3,\mathcal{R}(j)}$	0	0	0.1	0.3	0.6	1.0	1.5	2.1	2.8	3.6	4.5

Table 11: Description of the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ that were obtained by means of the algorithm presented in Section 2.7.3.

we provide the computed $\bar{x}_{i,\mathcal{F}(j)}$ and $\bar{x}_{i,\mathcal{R}(j)}$ in Table 11.

Given that the tightened constraint sets employed in the MPC are of the kind $\mathcal{X} \ominus \mathcal{R}(j)$, it is evident from Table 11 that the state that suffers most from the propagation of uncertainty is x_3, and that a maximum prediction horizon $N_p > 10$ would lead to an empty feasible set.

Computation of an RPI

As clarified in Assumption 3.6, the knowledge of an RPI Ω is fundamental to guarantee the recursive feasibility of the robust contraction-based MPC. In detail, it is needed to compute the maximum value that the contraction factor γ can take to guarantee the recursive feasibility of the MPC when the chosen prediction horizon is $q = 1$.

In the case of system (3.29), the computation of an RPI is straightforward. Note, indeed, that for any $x \in \mathcal{X}$ the input $u = (0, 0)$ is feasible and such that $f(x, u, w) = x$, $\forall w \in \mathcal{W}$. Therefore, as Ω is only required to be robustly invariant, without the need of finding a locally stabilizing control law, the last finding is sufficient to conclude that the entire state constraint set $\mathcal{X}$ is an RPI, i.e. $\Omega \equiv \mathcal{X}$.

Choice of the contractive function and the maximum prediction horizon

The computed sequence $\{\mathcal{R}(j)\}_{j\geq 0}$ puts an upper bound on the maximum prediction horizon that we can set. In particular, we must have that $N_p \leq 10$. Therefore, we must find a contractive positive definite function $\Gamma(x)$ that contracts sufficiently fast to ensure that the set $\Omega \ominus \mathcal{R}(1)$ is reached by the nominal system within 10 steps starting from any initial condition $x \in \mathcal{X}$. For this reason, the choice of a candidate contractive function and the computation of γ and N_p are carried out following the algorithm presented in Section 3.7.1, setting $N_p^* = 10$ and sampling 8000 points in $\mathcal{X}$.

The found contractive function is

$$\Gamma(x) = x^\top P x, \quad P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.167 & 0 \\ 0 & 0 & 0.167 \end{bmatrix}. \tag{3.31}$$

The employment of this contractive function results in $\omega = 14.44$ and $\Gamma_{\max} = 49.4$. Therefore, the upper bound of the contractivity factor is given by $\omega/\Gamma_{\max} = 0.2923$. Following the previously mentioned algorithm, a maximum prediction horizon $N_p = 10$ and a corresponding contraction factor $\gamma = 0.2487$ are obtained, which is lower than the previously computed upper bound.

Controller settings

Now that the main ingredients of the MPC have been defined, we need to choose a stage cost function and set the parameters ν, ϵ, ξ, and the initial condition of the internal controller state $\theta(0)$.
The stage cost function $\ell(x, u)$ is chosen as a quadratic cost function $\ell(x, u) = x^\top Q x + u^\top R u$, where $Q = I_3$ and $R = 0.01 I_2$, resulting in $\bar{\ell} = 216.6425$. The parameter ξ is chosen as $\xi = 5.7668 \cdot 10^3$ to verify the conditions of Theorem 3.2, while ν and ϵ, that are the parameters used in the update rule for $\theta(k)$, are set to $\nu = 0.99$ and $\epsilon = 10^{-8}$. The initial condition of the internal state θ is set to $\theta(0) = \nu\Gamma(x(0)) = 35.0183$.

Simulation results

In order to emphasize the robustness of the proposed method under different perturbation realizations, we run 100 simulations of the closed-loop system, setting the length of each simulation to 30 steps. All the simulations start from the given initial condition and use a perturbation w that is randomly sampled from $\mathcal{W}$ at each instant k. Our robust contraction-based MPC is implemented in its two-stage formulation.
The trajectories of the closed-loop system state are shown in Figure 8, which gives evidence of the stabilizing properties of the robust contraction-based MPC. Moreover, note that the state always verifies the constraints regardless of the perturbation realizations. Also the control actions provided by the MPC always verify the input constraints, as proved by Figure 9.
It is also interesting to analyze the evolution of the contractive function $\Gamma(x)$ over time, which is shown in Figure 10. Indeed, it is possible to note how the value of such contractive function actually grows sometimes. Nevertheless, this is not inconsistent with the theory. Our robust contraction-based MPC requires the nominal system, i.e. without perturbations, to have such contractivity property. It should

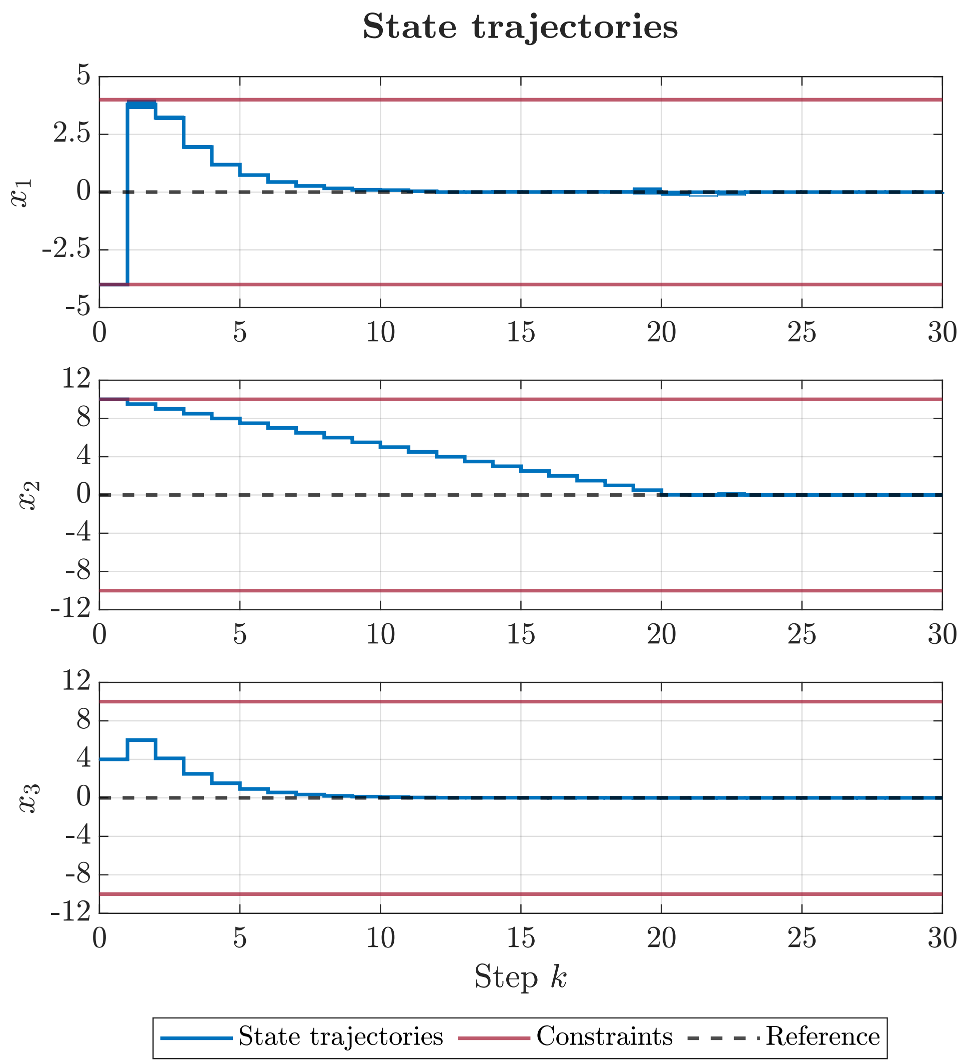


Figure 8: Closed-loop state trajectories under the robust contraction-based MPC. The nonholonomic nature of the system under control is emphasized by the first portion of the trajectories of states x_1 and x_3, which first deviate from the reference and only later approach it. Nonetheless, the overshoots that these states show verify the constraints and the state of the closed-loop system stabilizes around the origin after 20 steps.

also be noted that the sublevel set $\{x : \Gamma(x) \leq \omega\}$ is reached after only 4 steps on average. This is consistent with the fact that $\Omega \equiv \mathcal{X}$ and that the chosen contractive function maximizes the volume of the sublevel set $\{x : \Gamma(x) \leq \omega\}$.

3.8.2 Control of an uncertain quadruple-tank system

Section 3.8.1 provides an example that shows how the robust contraction-based MPC is a good alternative to QIH-MPC in those cases in which the computation of standard terminal ingredients is difficult. In the following, we instead show how the here-proposed MPC can be used also when standard terminal ingredients can be computed, and we give proof of how such ingredients can actually be exploited during the design of the robust contraction-based MPC.

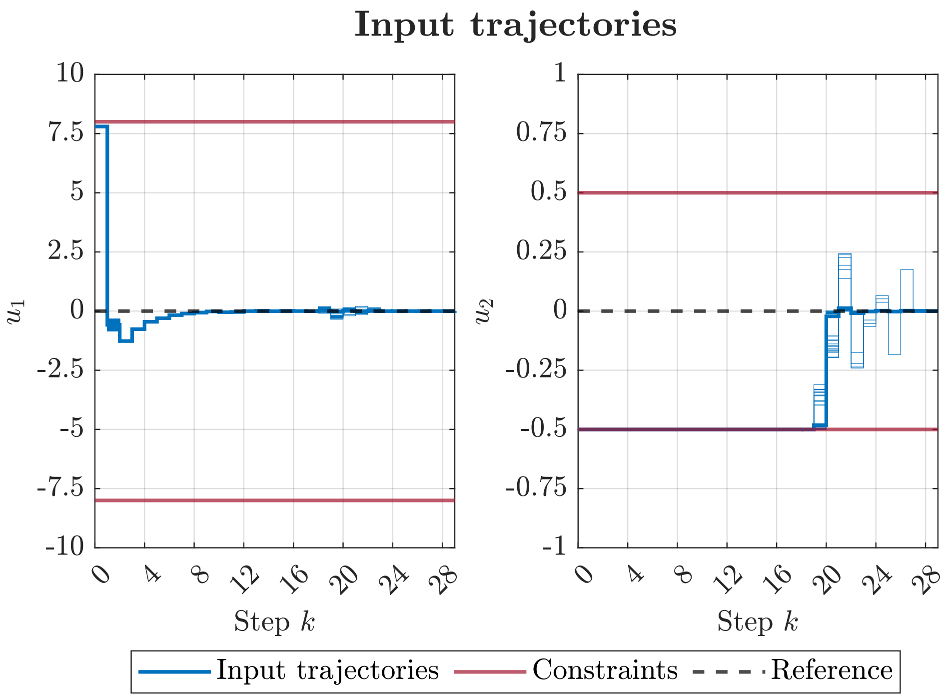


Figure 9: Control actions provided by the robust contraction-based MPC. The utility of employing an MPC can be observed for the control of a nonholonomic system like the one in (3.29)**. Indeed, the MPC required both u_1 and u_2 to reach saturation, but without ever going beyond the predefined constraints.**

The considered system is a forward Euler discretization of the uncertain quadruple-tank system provided in the Appendix C, using a sampling time $T_s = 15$s. The state of the system is composed of the levels of the quadruple tanks, which is $x := (h_1, h_2, h_3, h_4)$, while the flows of the two pumps are the controllable inputs of the system, namely $u = (q_a, q_b)$. The parameters γ_a and γ_b, which are characteristic parameters of the two three-way valves that act on the system, are considered to be perturbed by $w_1(k)$ and $w_2(k)$ respectively. In detail, $w(k) := (w_1(k), w_2(k))$ is considered to belong to a compact set, i.e.

$$w(k) \in \mathcal{W} := \{w \in \mathbb{R}^2 : \ |w_i| \leq 0.0325, \ \forall i \in \{1, 2\}\}.$$

The parameters and the bounds on the inputs and the states are given in the Appendix C. The goal is to control the system to the equilibrium (x_s, u_s) given by $x_s = (0.6702, 0.6549, 0.5435, 0.5887)$m and $u_s = (1.63, 2)\text{m}^3/\text{h}$. The initial condition is $x(0) = (1.3533, 1.1751, 1.2228, 0.8863)$m.

Given that the aim of the current subsection is to give an insight about how to adapt standard terminal ingredients so that they can be exploited during the design of the robust contraction-based MPC, the remainder of the subsection is organized as follows: (i) we first present the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$, then (ii) we present the computed terminal constraints, (iii) we show how such ingredients can be adapted for being used inside the robust contraction-based MPC, and (iv) we finally list the chosen controller settings. At the end, we also show the results of our closed-loop simulations.

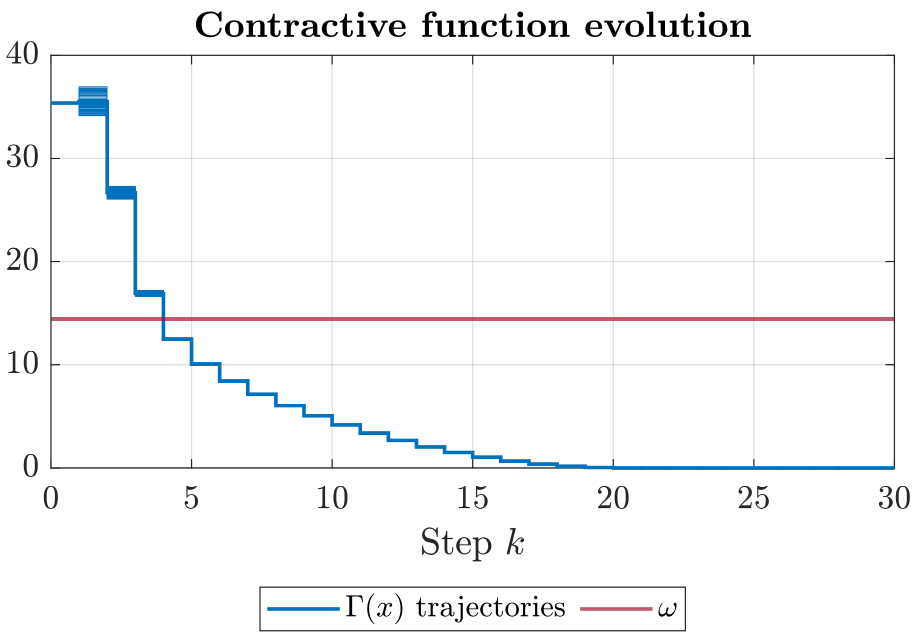


Figure 10: Evolution of the contractive function $\Gamma(x)$ during the simulations. Note that, on average, the sublevel set defined by $\Gamma(x) \leq \omega$ is reached after just 5 steps.

Computation of the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$

The computation of the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ is carried out following the method shown in Section 2.7.3, as also the quadruple-tank system is component-wise Lipschitz continuous. In detail, there exist non-negative real constants $L_{x,ia}$, $L_{u,ib}$ and $L_{w,ic}$, where $i, a \in \{1, 2, 3, 4\}$, $b, c \in \{1, 2\}$, such that

$$\begin{aligned} |f_i(x,u,w) - f_i(\check{x},\check{u},\check{w})| \leq \sum_{a=1}^{4} L_{x,ia}|x_a - \check{x}_a| + \sum_{b=1}^{2} L_{u,ib}|u_b - \check{u}_b| \\ + \sum_{c=1}^{2} L_{w,ic}|w_c - \check{w}_c|. \end{aligned} \tag{3.32}$$

The values of the constants $L_{x,ia}$, $L_{u,ib}$ and $L_{w,ic}$ are respectively reported in Tables 12, 13 and 14.

$L_{x,ia}$		a			
		1	*2*	*3*	*4*
i	*1*	0.95	0	0.18	0
	2	0	0.95	0	0.15
	3	0	0	0.96	0
	4	0	0	0	0.96

Table 12: Values of the constants $L_{x,ia}$ of the quadruple-tank system.

$L_{u,ib}$		b = 1	b = 2
i	1	0.025	0
	2	0	0.035
	3	0	0.17
	4	0.17	0

Table 13: Values of the constants $L_{u,ib}$ of the quadruple-tank system.

$L_{w,ic}$		c = 1	c = 2
i	1	0.25	0
	2	0	0.275
	3	0	0.275
	4	0.25	0

Table 14: Values of the constants $L_{w,ic}$ of the quadruple-tank system.

Remark 3.9

One might wonder why the component-wise Lipschitz constants shown in Tables 12, 13 and 14 are different than the constants shown in Tables 3, 4 and 5, given that the system under control is the same. This difference is actually motivated by the following three points:

- *In the case study in Chapter 2, a Runge Kutta discretization of the system was considered, while in the current example a simple forward Euler integrator is employed.*
- *The bounds on the uncertainty variables are different.*
- *In the case study in Chapter 2, the found component-wise Lipschitz constants are related to the closed-loop system under the pre-stabilizing control laws $\pi(\cdot,\cdot,\cdot)$, while in the current example the same constants are computed for the open-loop system.*

The computed sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ are such that

$$\mathcal{F}(j) = \{x \in \mathbb{R}^4 : \ |x_i| \leq \bar{x}_{i,\mathcal{F}(j)}, \forall i \in \{1,2,3,4\}\}$$

and

$$\mathcal{R}(j) = \{x \in \mathbb{R}^4 : \ |x_i| \leq \bar{x}_{i,\mathcal{R}(j)}, \forall i \in \{1,2,3,4\}\}.$$

Therefore, as for the example in Section 3.8.1, we provide the computed $\bar{x}_{i,\mathcal{F}(j)}$ and $\bar{x}_{i,\mathcal{R}(j)}$ in Table 15.

j	$\mathcal{F}(j)$				$\mathcal{R}(j)$			
	$\bar{x}_{1,\mathcal{F}(j)}$	$\bar{x}_{2,\mathcal{F}(j)}$	$\bar{x}_{3,\mathcal{F}(j)}$	$\bar{x}_{4,\mathcal{F}(j)}$	$\bar{x}_{1,\mathcal{R}(j)}$	$\bar{x}_{2,\mathcal{R}(j)}$	$\bar{x}_{3,\mathcal{R}(j)}$	$\bar{x}_{4,\mathcal{R}(j)}$
0	0.0081	0.0089	0.0089	0.0081	0	0	0	0
1	0.0093	0.0097	0.0086	0.0078	0.0081	0.0089	0.0089	0.0081
2	0.0104	0.0104	0.0082	0.0075	0.0175	0.0186	0.0175	0.0159
3	0.0114	0.0110	0.0079	0.0072	0.0279	0.0290	0.0258	0.0234
4	0.0122	0.0115	0.0076	0.0069	0.0392	0.0400	0.0337	0.0306
5	0.0130	0.0120	0.0073	0.0066	0.0514	0.0516	0.0413	0.0375
6	0.0136	0.0124	0.0070	0.0064	0.0644	0.0635	0.0485	0.0441
7	0.0142	0.0127	0.0067	0.0061	0.0781	0.0759	0.0555	0.0505
8	0.0147	0.0130	0.0064	0.0059	0.0923	0.0886	0.0623	0.0566
9	0.0151	0.0132	0.0062	0.0056	0.1070	0.1016	0.0687	0.0625
10	0.0155	0.0134	0.0059	0.0054	0.1221	0.1149	0.0749	0.0681
11	0.0158	0.0135	0.0057	0.0052	0.1376	0.1283	0.0808	0.0735
12	0.0160	0.0136	0.0055	0.0050	0.1534	0.1418	0.0865	0.0787
13	0.0162	0.0137	0.0053	0.0048	0.1695	0.1555	0.0920	0.0836
14	0.0163	0.0137	0.0050	0.0046	0.1857	0.1692	0.0973	0.0884
15	0.0164	0.0137	0.0048	0.0044	0.2020	0.1829	0.1023	0.0930
16	0.0165	0.0137	0.0047	0.0042	0.2185	0.1967	0.1072	0.0974
17	0.0165	0.0137	0.0045	0.0041	0.2350	0.2104	0.1118	0.1016

Table 15: Description of the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ related to the quadruple-tank system that were obtained by means of the algorithm shown in Section 2.7.3. The values for $j > 17$ are not reported because, as explained later in the text, the maximum prediction horizon of the MPC is set to $N_p = 17$.

Computation of the terminal ingredients

In order to introduce the design of the terminal ingredients of a standard MPC with terminal cost and terminal set, it is necessary to anticipate some design choices that are also explained later. Indeed, in order to design such ingredients, the knowledge of the stage cost function $\ell(x, u)$ is needed. For this reason, we choose to employ a quadratic stage cost function

$$\ell(x, u) = (x - x_s)^\top Q(x - x_s) + (u - u_s)^\top R(u - u_s),$$

where $Q = I_4$ and $R = 0.01I_2$.

Thereafter, we need to design a terminal cost function $V_f(x)$ that behaves as a CLF inside an invariant set $\mathcal{X}_f$ thanks to a locally stabilizing control law $u = \kappa_f(x)$. In order to simplify the design process, we decide to employ a quadratic terminal cost function $V_f(x) = (x-x_s)^\top P(x-x_s)$, where P is symmetric and positive definite, to use as terminal set a sublevel set of $V_f(x)$, namely $\mathcal{X}_f \coloneqq \{x \in \mathbb{R}^4 : \ V_f(x) \leq \rho\}$ for some $\rho > 0$, and to employ a locally stabilizing linear controller, i.e. $u = u_s + K(x - x_s)$, for some K. In particular, we design the terminal ingredients following the method presented in Section 1.1.1, which is based on the linearization of the system at the desired equilibrium.

The linearized system is described by the matrices

$$A = \begin{bmatrix} 0.9125 & 0 & 0.0767 & 0 \\ 0 & 0.8971 & 0 & 0.0673 \\ 0 & 0 & 0.9233 & 0 \\ 0 & 0 & 0 & 0.9327 \end{bmatrix}, B = \begin{bmatrix} 0.0208 & 0 \\ 0 & 0.0278 \\ 0 & 0.0417 \\ 0.0486 & 0 \end{bmatrix}.$$

Provided that the couple (A, B) is fully controllable, it is possible to compute P, K, and ρ as in Section 1.1.1.

Setting $\kappa = 0.025$, we get:

$$P = \begin{bmatrix} 6.0794 & -0.9107 & 1.5580 & -1.9296 \\ -0.9107 & 4.9770 & -2.1981 & 1.0145 \\ 1.5580 & -2.1981 & 4.1999 & -1.0133 \\ -1.9296 & 1.0145 & -1.0133 & 3.3115 \end{bmatrix},$$

$$K = \begin{bmatrix} -1.7914 & -1.6219 & 0.8432 & -6.9335 \\ -2.2376 & -2.5948 & -6.7541 & 0.6823 \end{bmatrix},$$

$$\rho = 0.124.$$

Adaptation of the terminal ingredients for usage in the robust contraction-based MPC

In order to recycle the terminal ingredients introduced earlier, the first thing to do is checking whether the invariant set $\mathcal{X}_f$ is also robustly invariant. Indeed, the set $\mathcal{X}_f$ being invariant for the nominal system does not ensure that the invariance property is kept also for all the possible perturbation realizations $w \in \mathcal{W}$. Here, $\mathcal{X}_f$ is also robustly invariant, therefore $\Omega = \mathcal{X}_f$ can be set.

Moreover, note that $V_f(x)$ is positive definite by design and that the interior of the set $\Omega \ominus \mathcal{R}(1)$ is non-empty. For this reason, $V_f(x)$ can be employed as contractive function, i.e. $\Gamma(x) = V_f(x)$, and it is ensured that there exists a $\omega > 0$ such that $\{x : \Gamma(x) \le \omega\} \subseteq \Omega \ominus \mathcal{R}(1)$. Here, $\omega = 0.074$.

Proceeding with computing the upper bound on the contractivity factor γ, which is computed as $\omega/\Gamma_{\max}$. Given that $\Gamma_{\max} = 7.7408$, the contraction factor must be such that $\gamma \le 0.0096$.

Remark 3.10

The found upper bound on γ clearly shows how recycling the terminal ingredients used in a standard MPC is possible but inefficient. Indeed, imposing such a low contractivity factor is very similar to imposing a terminal equality constraint, consequently leading to large prediction horizons.

The same algorithm shown in Section 3.8.1 is used to compute the lowest maximum prediction horizon N_p that ensures the needed contractivity and the corresponding γ value. The obtained results are $N_p = 17$ and $\gamma = 0.0079$, which verify all the conditions and ensure that the the sets $\mathcal{X} \ominus \mathcal{R}(j)$, $\forall j \in [1, N_p]$, are non-empty.

Controller settings

The employed stage cost function has already been anticipated in Section 3.8.2. In order for the parameter ξ to verify all the conditions of Theorem 3.1, its value is set to $\xi = 73.0013$, as the maximum value of the stage cost function over the set $\mathcal{X} \times \mathcal{U}$ is $\bar{\ell} = 2.13$.
Regarding the dynamics of the internal controller state θ, its update parameters ν and ϵ are set to $\nu = 0.99$ and $\epsilon = 10^{-8}$, while its initial condition is set to $\theta(0) = \nu\Gamma\left(x(0)\right) = 4.7323$.

Simulation results

As also done for the example in Section 3.8.1, we run 100 simulations of the closed-loop system, setting the length of each simulation to 50 steps, i.e. 750s. All the simulations start from the given initial condition and use a perturbation w that is randomly sampled from $\mathcal{W}$ at each instant k. Our robust contraction-based MPC is implemented in its two-stage formulation.
The trajectories of the system states and the control actions that are computed by the MPC at each sampling instant are respectively shown in Figure 11 and Figure 12. The evolution of the contractive function $\Gamma(x)$, which is shown in Figure 13, is totally different from the one obtained on the nonholonomic system in Section 3.8.1. In particular, while also in this case $\Gamma(x)$ decreases over time and approaches zero, the value ω is reached after a relatively large number of sampling instants. This is consistent with the design process that was followed. Indeed, for the nonholonomic system in Section 3.8.1, the state constraint set $\mathcal{X}$ coincides with the RPI Ω and the contractive function $\Gamma(x)$ is designed with the aim of guaranteeing contractivity within short horizons. In the case of the currently analyzed quadruple-tank system, the contractivity ingredients are not designed with the same goal, therefore guaranteeing all the stability and robustness properties ensured by the robust contraction-based MPC in a less efficient way.

3.9 Conclusion

In this chapter, we presented a tube-based MPC for the regulation of nonlinear systems that are subject to bounded perturbations and whose state function components are component-wise uniformly

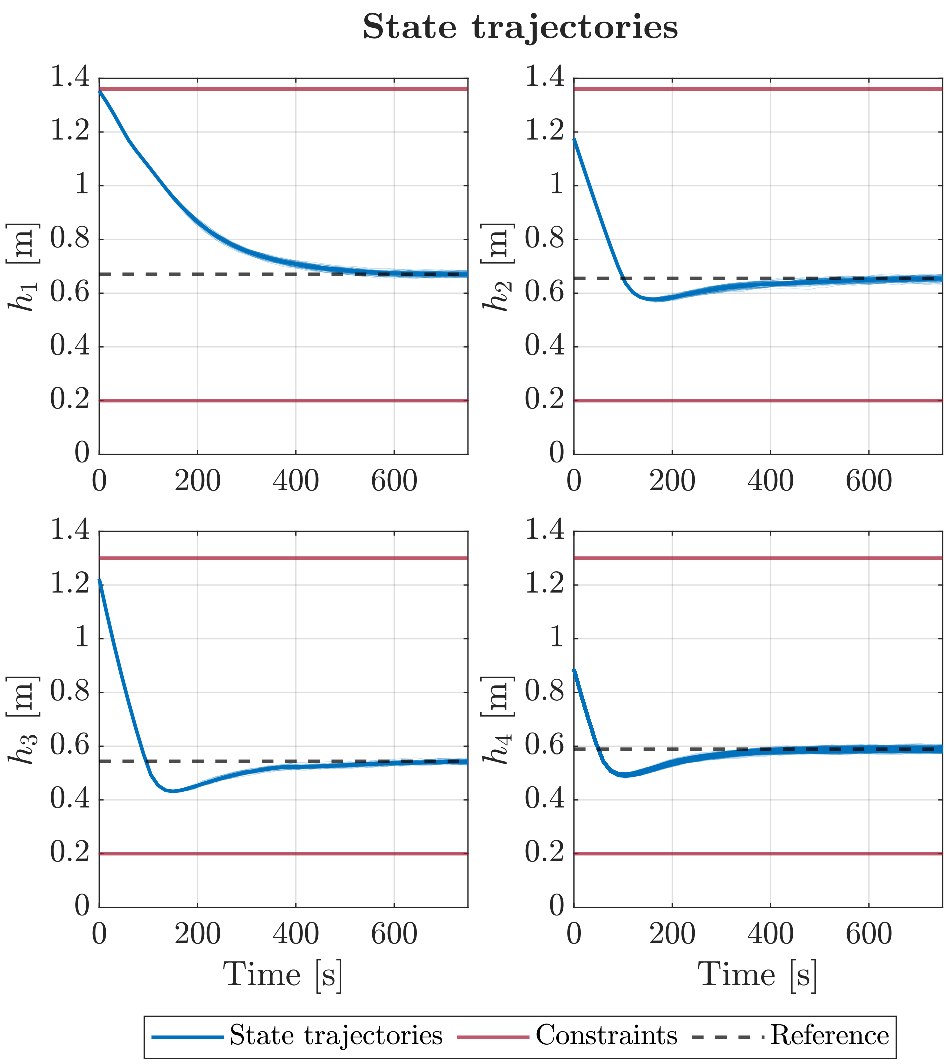


Figure 11: Closed-loop trajectories of the water levels in the four tanks under the robust contraction-based MPC. Consistently with the theory presented in this chapter, the constraints are always verified and the state stabilizes around the reference, regardless of the perturbation realizations.

continuous. In particular, the proposed method aims at stabilizing the closed-loop system and being easily implementable also in cases in which the design of standard terminal ingredients is challenging. In detail, the need of a terminal cost function that is a local CLF is overcome by employing as terminal cost function a simple positive definite function, and relying on the contractivity property of stabilizable systems to prove the input-to-state (practical) stability of the closed-loop system. Another singular property of the MPC is that the prediction horizon is a decision variable. While the latter feature is motivated by stability reasons, it renders the optimal control problem to be solved a mixed-integer problem. In order to reduce the computational burden of such problem, we show how an equivalent two-stage formulation without integer variables can provide the same results while limiting the increase in the computational complexity. The controller was tested on two different case studies, i.e. a perturbed nonholonomic system, for which the computation of standard terminal ingredients

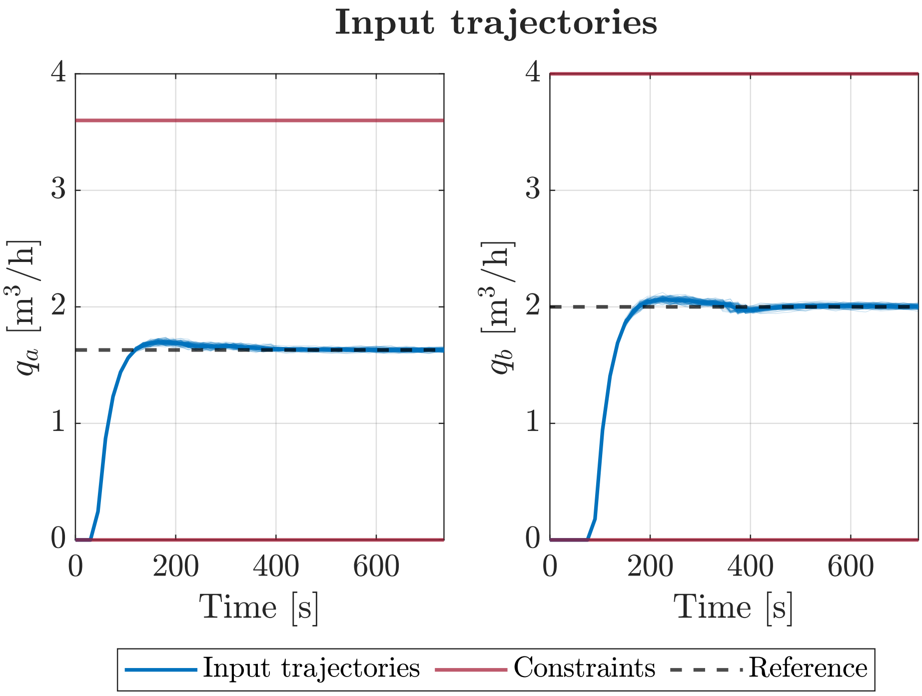


Figure 12: Control actions provided by the robust contraction-based MPC. Exactly like for the previously addressed nonholonomic system, the MPC cleverly exploits the saturation levels of the two inputs of the system.

would be challenging, and a perturbed quadruple-tank system, showing also how standard terminal ingredients can be employed in this novel framework. Despite the mathematical soundness and the practical effectiveness of our contractive MPC, we are aware of the fact that there are issues that deserve further investigation. In particular, it would be interesting to investigate how to optimally choose a contractive function that ensures contractivity within short horizons and how to further limit the increase in the computational burden provided by both the mixed-integer and the two-stage formulations of the optimal control problem.

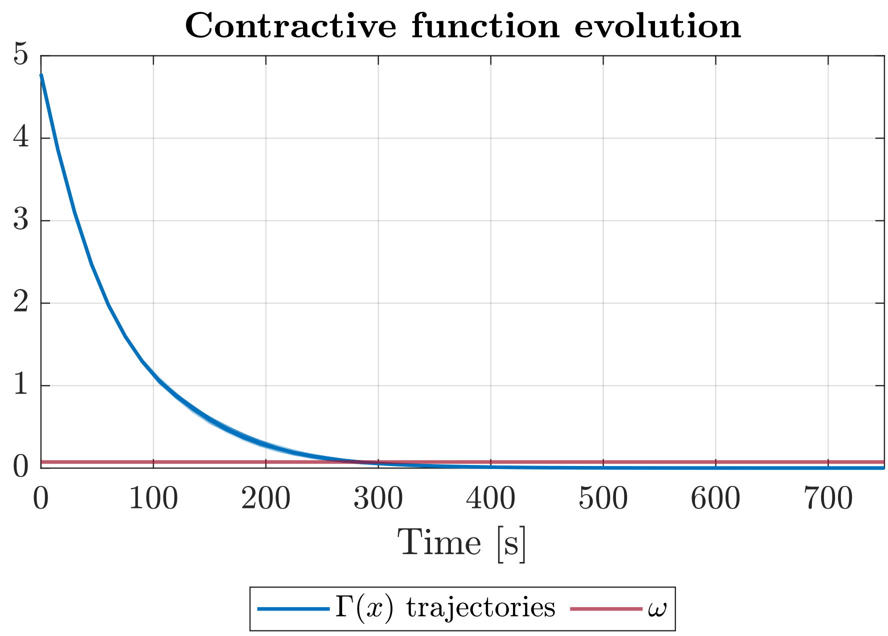


Figure 13: Evolution of the contractive function $\Gamma(x)$ during the simulations. Differently from the previously addressed nonholonomic system, in this case the sublevel set given by $\Gamma(x) \leq \omega$ is reached after a longer time. This is due to a suboptimal choice of the contractive function $\Gamma(\cdot)$.

Chapter 4. Robust contractive MPC based on Gaussian Processes

4.1 Introduction

Chapter 3 addressed an MPC scheme for regulation that cleverly exploits the stabilizability of a system and the component-wise uniform continuity of its state function to ensure recursive feasibility and input-to-state (practical) stability of the closed-loop system thanks to the employment of terminal ingredients that have to fulfill milder conditions than those of a standard MPC. Therefore, we can conclude that the original goal of designing a method that could help us face both the issues discussed at the beginning of Chapter 3, namely (i) the control of systems for which the design of standard terminal ingredients is troublesome, while (ii) ensuring robustness with respect to known bounded perturbations, was achieved.

There is actually a further issue that the control community is trying to solve. Indeed, up to this point we have first addressed MPC schemes that assume to have no model mismatches and then we have turned our attention to schemes that take into account generic perturbations acting on the system, e.g. also mismatches between the real system and the model that is employed in the MPC optimal control problem. Still, also in robust MPC schemes a model of the system is considered to be available. Then, a question that arises is whether this condition is always true. The answer to this question is not trivial. As a matter of fact, the systems to be controlled are getting more and more complex, and the design of prediction models was reported to take up to the 80% of the MPC commissioning effort, as outlined in [136]. This is why the control community has been investigating *data-driven MPC* schemes during the last two decades. In particular, since there is ambiguity regarding the definition of what a data-driven method is, in this chapter we refer to data-driven methods as generic methods that do not employ first-principle models of the system to be controlled and for which data are of major importance, i.e.:

1. methods that do not require finding any parameters of a mathematical model of the system and directly use data to make predictions;

2. methods that do require finding parameters (or hyperparameters) of a mathematical model of the system, but such model is not a first-principle model and requires the availability of a substantial amount of data to be designed.

This definition may be questionable, but it is motivated by the fact that we argue that no MPC is truly model-free. In fact, famous methods like those introduced in [15, 29, 30], which represent a significant breakthrough in the evolution of MPC, are based on the well known Willems' fundamental lemma first introduced in [144], which enables the characterization of the entire subspace of the possible trajectories of a Linear Time-Invariant (LTI) system based on raw data. Still, even though these methods do not require to find any parameters, they assume the linearity and time-invariance of the system that generates the data. Consequently, an underlying structure of the model is assumed, and due to this assumption the model can be formulated in such a way that it can be parameterized by means of raw data. The same applies to [17], which applies the same strategy to the control of continuously differentiable nonlinear systems. Hence, all the previously cited works belong to the first group of data-driven methods we mentioned earlier.

The second group of data-driven methods is that of MPC schemes that make predictions by means of black-box parametric and non-parametric methods. In particular, among the parametric machine learning methods that have been used to learn the prediction model used in the MPC context, neural networks (NN) play a crucial role. Indeed, some examples of MPC schemes based on NNs can be found in [3, 66, 78]. Among the non-parametric methods that have been used to identify the MPC prediction model, we find Gaussian Processes (GP), which have proved to be particularly suitable for their use in MPC. This is because they employ prior information about the process to be identified [67], they can model complex functions also with few data points, and Gaussian Process regression (GPR) provides prediction probability distributions whose variance can be used as a measure of the prediction uncertainty.

Several MPC formulations based on GPs (which we refer to as GP-MPC) have been proposed, and the uncertainty measures provided by GPR have been taken into account to ensure robust feasibility following both the theory of robust MPC, e.g. [99], which ignores the probabilistic nature of uncertainties, and stochastic MPC, as [21, 23, 57, 68, 69, 83], where the stochasticity of the uncertainty is explicitly considered in the design of the cost function and the state constraints. Nevertheless, many works are focused on maximizing the performances and solving all the possible implementation issues of the proposed GP-MPC schemes, without studying the stability of the closed loop from a mathematical perspective. In [23], a stochastic GP-MPC is proposed for the control of an unmanned quadrotor, focusing in particular on how to linearize the identified GP to get a convex optimization problem and how to address the main issues related to the usage of active-set solvers to solve the created optimization problem. Different ways for dealing with the problem of uncertainty propagation with GPs are presented in [57], where also a method for translating chance constraints into deterministic constraints on the mean value of the predictions is proposed, and an efficient implementation of the presented GP-MPC is shown to provide execution times that are lower than 20 [ms].

A complete investigation about the stability properties of GP-MPC schemes was proposed in [83] for the control of linear uncertain systems, while [99] proves nominal and robust stability of nonlinear systems that are controlled by a GP-MPC and provides an effective way of improving the used GP by collecting new samples online. However, the GP-MPC proposed in [99] is not ensured to be recursively feasible in case of hard state or input constraints.

Therefore, in the remainder of this chapter we first introduce GPs and discuss how they can be employed to model dynamic systems, then we present a tube-based GP-MPC for the regulation of a wide family of nonlinear systems whose dynamics is unknown and that uses simplified stability-related terminal ingredients. With this aim, we recur to the MPC presented in Chapter 3 and show under what conditions a GP modeling the system under control can be employed in such MPC scheme without affecting its recursive feasibility and stabilizing capabilities. The method is then validated in simulation on a continuous stirred-tank reactor (CSTR) modeled by means of GPs.

The results presented in this chapter are available as the preprint [115].

4.2 Fundamentals of Gaussian Process learning

This section addresses GPs, providing the basic concepts that are needed to understand the GP-MPC that is treated in this chapter. For this reason, even if GPs can be used for classification problems, we only introduce how to employ GPs to solve regression problems. For further details on GPs, we refer the reader to [125].

4.2.1 Basics of Gaussian Processes

Before discussing GPs, we first make some considerations about multivariate Gaussian distributions, inspired by [98], as their understanding will make the comprehension of the concepts behind GPs easier. For this reason, let us consider a random variable A that takes values a and whose distribution follows an $n_{\mathcal{D}}$-dimensional multivariate Gaussian $\mathcal{N}(\mu, \Sigma)$, i.e. $A \sim \mathcal{N}(\mu, \Sigma)$. This means that the probability density function (PDF) of A (evaluated in a) is such that

$$p(a) = \frac{1}{\sqrt{(2\pi)^{n_{\mathcal{D}}} \det(\Sigma)}} \exp\left[(a-\mu)^\top \Sigma^{-1} (a-\mu)\right],$$

where $\mu \in \mathbb{R}^{n_{\mathcal{D}}}$ is the mean vector and $\Sigma \in \mathbb{R}^{n_{\mathcal{D}} \times n_{\mathcal{D}}}$ is the covariance matrix. In particular, the mean vector is such that $\mu = (\mathbb{E}[A_1], \ldots, \mathbb{E}[A_{n_{\mathcal{D}}}])$, while each element of Σ is given by $\Sigma_{i,j} = \mathbb{E}\left[(A_i - \mu_i)(A_j - \mu_j)\right]$. Further analyzing Σ, we note that the elements on the diagonal, i.e. with

$i = j$, denote the variances of the elements of A, while the terms off the diagonal, namely with $i \neq j$, are the covariance terms that quantify the mutual dependencies between A_i and A_j. A particular case is that of covariance matrices Σ whose elements off the diagonal are zero. In such cases, as there is no mutual dependence between the elements of A, we get $n_{\mathcal{D}}$ univariate Gaussian distributions instead of a $n_{\mathcal{D}}$-dimensional multivariate Gaussian distribution.

A GP is a stochastic process [10] and is a generalization of the Gaussian probability distribution to functions. Indeed, just as a Gaussian distribution is completely defined by the mean vector μ and the covariance matrix Σ, a GP is completely defined by a mean function $m(\cdot)$ and a covariance function $k(\cdot,\cdot)$. Therefore, we denote a function $f : \mathbb{R}^{n_z} \longrightarrow \mathbb{R}$ that is a GP as

$$f(z) \sim \mathcal{GP}\left(m(z), k(z, \check{z})\right),$$

where $z, \check{z} \in \mathbb{R}^{n_z}$ are deterministic inputs, or *regressors*. The result of having a mean function and a covariance function is that, differently from the previously defined random variable a, $f(\cdot)$ does not have a constant mean and a constant covariance matrix. In fact, the mean and the covariance of $f(\cdot)$ depend on the regressor z. Therefore, different values of $z, \check{z} \in \mathbb{R}^{n_z}$ provide different Gaussian random variables $f(z)$ and $f(\check{z})$. In particular, the covariance function $k(\cdot,\cdot)$ is such that $k(z,z)$ returns the variance of $f(z)$, while $k(z,\check{z})$ returns the covariance between $f(z)$ and $f(\check{z})$.

The last statements are sufficient to conclude that

$$\begin{bmatrix} f(z) \\ f(\check{z}) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(z) \\ m(\check{z}) \end{bmatrix}, \begin{bmatrix} k(z,z) & k(z,\check{z}) \\ k(\check{z},z) & k(\check{z},\check{z}) \end{bmatrix}\right),$$

which means that the vector $(f(z), f(\check{z}))$ is a two-dimensional multivariate Gaussian distribution. However, this reasoning can be done for any finite number of regressors. This fact helps us understand why [125] gives the following definition of what a GP is.

Definition 4.1: Gaussian Process

A GP is a collection of random variables, any finite number of which have a joint Gaussian distribution.

4.2.2 Bayesian inference

In this subsection, we discuss how GPs can be used with a Bayesian approach to infer the Gaussian distribution $f(z_*)$ at a test point z_*.

As outlined in Section 4.2.1, a GP is completely defined by a mean function $m(\cdot)$ and a covariance function $k(\cdot,\cdot)$, which describe the way a function $f(\cdot)$ is likely to behave in the space where it is defined. This is the reason why, when we model a function $f(\cdot)$ with a GP, we place a *prior* over a space of functions. Indeed, in the machine learning context, when a GP is used to model an unknown function $f(\cdot)$, the functions $m(\cdot)$ and $k(\cdot,\cdot)$ are chosen to be consistent with the few known (or supposed) characteristics of $f(\cdot)$, e.g. mean-square continuity and differentiability, and to take into account other possible low-complexity and inaccurate models of $f(\cdot)$.

Let us explain the last concept with some examples. As confirmed also in [125], the mean function is often chosen to be $m(z) = 0$. This means that the designer has no prior information about how the function $f(\cdot)$ behaves on average in the space where it is defined. As we will see in the remainder of this subsection, this is not a limiting choice, as it does not confine the posterior distribution at a test point z_* to have zero mean. However, in case a simplified model of $f(\cdot)$ is available, it may be sensible to employ it as the GP mean function.

Turning to the relationship between the chosen covariance function $k(\cdot,\cdot)$ and the latent function $f(\cdot)$. To do so, a widely used covariance function [125] is considered, i.e. the squared exponential with automatic relevance determination (SE-ARD) covariance function, defined as

$$k(z,\check{z}) = \sigma_f^2 \exp\left(-\frac{1}{2}(z-\check{z})^\top \Lambda^{-1}(z-\check{z})\right), \tag{4.1}$$

where $\Lambda = \mathrm{diag}(l_1^2,\ldots,l_{n_z}^2)$. We refer to the elements of the vector $\zeta \coloneqq (\sigma_f, l_1, \ldots, l_{n_z})$ as *hyperparameters* and, for the moment, we consider them to be known. The SE-ARD covariance function is infinitely differentiable. Therefore, choosing to model a function $f(\cdot)$ with a GP that employs a SE-ARD covariance function results in the modeled process being infinitely mean-square differentiable, thus smooth.

Next, we analyze what happens when data are available. In particular, we suppose that a dataset $\mathcal{D}$ composed of $n_\mathcal{D}$ pairs $(z_{(j)}, y_{(j)})$ of regressors and noisy measures $y_{(j)} = f\left(z_{(j)}\right) + \eta_{(j)}$ is available. In detail, the random variables $\eta_{(j)}$ are considered to be independent and identically distributed (i.i.d.) and such that $\eta_{(j)} \sim \mathcal{N}(0,\sigma_\eta^2)$. Furthermore, let us define the vector $\mathsf{y} \coloneqq \left(y_{(1)},\ldots,y_{(n_\mathcal{D})}\right)$, the matrix $Z \coloneqq \left[z_{(1)}\ \ldots\ z_{(n_\mathcal{D})}\right]^\top$ and the covariance matrix $K \coloneqq k(Z,Z)$. Then, considering a test regressor z_*, for which the inference of $f(z_*)$ is desired, the joint prior distribution of y and $f(z_*)$ is given by

$$\begin{bmatrix} \mathsf{y} \\ f(z_*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(Z) \\ m(z_*) \end{bmatrix}, \begin{bmatrix} K + \sigma_\eta^2 I_{n_\mathcal{D}} & k(Z,z_*) \\ k(z_*,Z) & k(z_*,z_*) \end{bmatrix}\right), \tag{4.2}$$

where the covariance matrix of y is $K + \sigma_\eta^2 I_{n_\mathcal{D}}$ (and not just K) as the values $y_{(j)}$ are noisy measures. To infer the *posterior* distribution of $f(z_*)$, which is the goal of GPR, we need to condition (4.2) on the available dataset $\mathcal{D}$. However, as $(m(Z)^\top, m(z_*))$ is a Gaussian distributed random variable, exploiting the properties of Gaussian distributions, shown in the Appendix B, the desired conditional distribution can be easily obtained as

$$p\left(f(z_*)|z_*, Z, \mathsf{y}, \zeta, \sigma_\eta\right) = \mathcal{N}\left(\mu(z_*|\mathcal{D}), \sigma^2(z_*|\mathcal{D})\right), \tag{4.3}$$

where

$$\mu(z_*|\mathcal{D}) = m(z_*) + k(z_*, Z)\left(K + \sigma_\eta^2 I_{n_\mathcal{D}}\right)^{-1}(\mathsf{y} - m(Z)), \tag{4.4a}$$

$$\sigma^2(z_*|\mathcal{D}) = k(z_*, z_*) - k(z_*, Z)\left(K + \sigma_\eta^2 I_{n_\mathcal{D}}\right)^{-1} k(Z, z_*). \tag{4.4b}$$

A common way to use GPs in learning problems is to exploit the posterior mean function $\mu(\cdot|\mathcal{D})$ as an estimator of the true function $f(\cdot)$ and the posterior variance function $\sigma^2(\cdot|\mathcal{D})$ as a function providing measures of the uncertainty of the estimator.

4.2.3 Hyperparameter estimation

Up to this point, we assumed the hyperparameter vector $(\zeta^\top, \sigma_\eta^2)$ to be known. However, in most cases such hyperparameters are not known a priori and have to be estimated to "fit" the latent function at the available points $(z_{(j)}, y_{(j)})$. In other words, the choice of the mean function $m(\cdot)$ and the covariance function $k(\cdot,\cdot)$ is carried out taking into account global properties of the latent function, e.g. smoothness, and then the value of the hyperparameters $(\zeta^\top, \sigma_\eta^2)$ is chosen so as to match the behavior of the latent function where we have data.

Estimating the hyperparameters is crucial, because two GPs employing the same mean and covariance functions and different hyperparameters lead to significantly different processes. An example of this is shown in Figure 14, where two GPs are employed to model a function $f : \mathbb{R} \to \mathbb{R}$. In such one-dimensional case, a SE-ARD covariance function of the kind

$$k(z, \check{z}) = \sigma_f^2 \exp\left(-\frac{(z - \check{z})^2}{2l_1^2}\right)$$

and a zero mean function are employed. The overall variance σ_f^2 is set to $\sigma_f^2 = 1$, while the lengthscale l_1 is first given a value of 0.2 and then a value of 2. The two plots in Figure 14 show functions that were sampled from the two GPs.

The most common method for estimating the hyperparameters is described below. An intuitive idea is that of picking those hyperparameter values that maximize the probability of observing the available pairs $(z_{(j)}, y_{(j)})$. From a mathematical point of view, this translates into the maximization of the marginal likelihood $p(\mathsf{y}|Z, \zeta, \sigma_\eta^2)$, which is equivalent to maximizing the log-marginal likelihood

$$\log p(\mathsf{y}|Z, \zeta, \sigma_\eta^2) = -\frac{1}{2}\mathsf{y}^\top\left(K + \sigma_\eta^2 I_{n_\mathcal{D}}\right)^{-1}\mathsf{y} - \frac{1}{2}\log\left(\det\left(K + \sigma_\eta^2 I_{n_\mathcal{D}}\right)\right) - \frac{n_\mathcal{D}}{2}\log(2\pi) \tag{4.5}$$

with respect to $(\zeta^\top, \sigma_\eta^2)$.

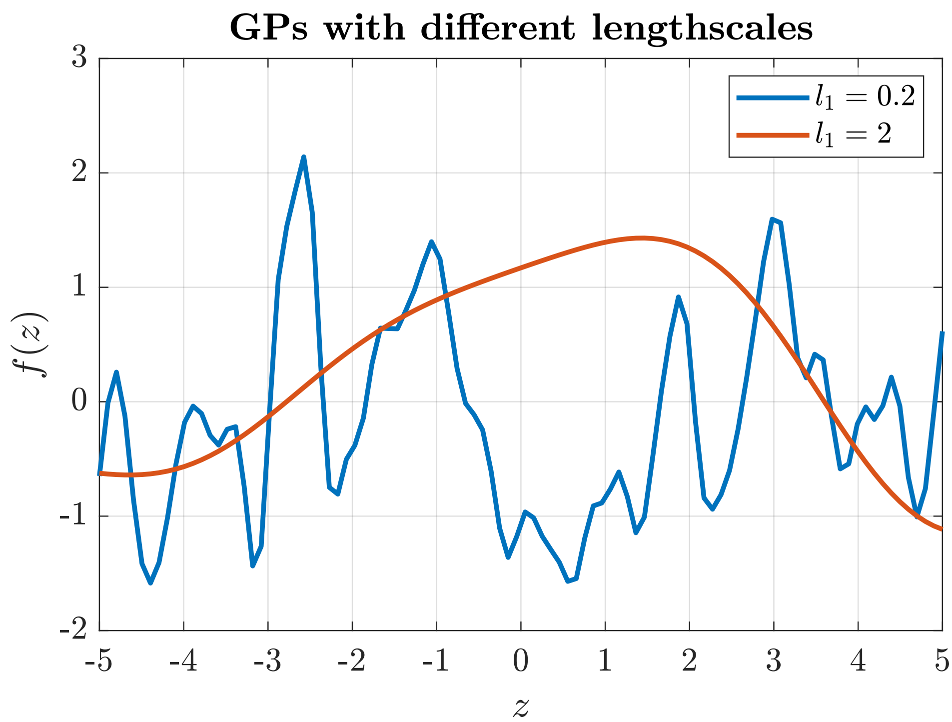


Figure 14: Functions that were sampled from two GPs employing the same zero mean function and the same SE-ARD covariance function, but using different lengthscales. It is possible to note how small lengthscales result in functions that change more rapidly.

4.2.4 Sparse Gaussian Process approximations

Compared to parametric inference methods used to model complex nonlinear functions, like NNs, training a GP requires to learn a much more limited number of (hyper)parameters. Moreover, due to their probabilistic nature, GPs are often more interpretable than NNs. However, these advantages come at a cost. Indeed, the computational burden of both hyperparameter estimation and inference (GPR) can get significantly high when the number $n_{\mathcal{D}}$ of training points is large. This is due to the fact that both these operations require to invert the matrix $(K+\sigma_\eta^2 I_{n_{\mathcal{D}}}) \in \mathbb{R}^{n_{\mathcal{D}}\times n_{\mathcal{D}}}$. Since this matrix is dense, its inversion requires $O(n_{\mathcal{D}}^3)$ operations. For this reason, several *sparse approximations* of GPs have been proposed. An interesting survey on such approximations is provided in [118]. Nonetheless, we can summarize the ways GPs are approximated in three approaches:

1. Using a subset of data (SoD). Even though dropping part of the available data means wasting information, using a reduced subset of $n_r < n_{\mathcal{D}}$ points obviously reduces the complexity of the inversion of $(K+\sigma_\eta^2 I_{n_r})$ to $O(n_r^3)$.

2. Using an approximation of the prior distribution, as in [131].

3. Using an approximation of the posterior distribution, as in [139].

There is an important difference between the last two philosophies. Approximating the prior distribution means that the assumption that $f(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$ is broken. Therefore, the goal in such approximations is not to distort the base GP prior excessively, while ensuring that the modified prior results in a simplified inference step. On the contrary, approximations of the posterior distribution keep the standard GP prior assumption and just approximate the inference step. Nonetheless, we do not delve into the details of such approximation philosophies, as their knowledge is not necessary for understanding the MPC presented in this chapter.

4.2.5 Gaussian Process learning in control

The setting in which learning methods are usually employed is when the mapping $f(\cdot)$ is static, and this applies also to GPs. Nevertheless, in the MPC context and other control frameworks, e.g. reinforcement learning, as in [36, 37], GPs are used to model a dynamic system, so as to enable the design of a controller based on the learnt model. This is because GPs can be easily adapted to the modeling of dynamic systems described by the equations

$$x(k+1) = f\left(x(k), u(k)\right),$$
$$y(k) = h\left(x(k)\right),$$

where $x(k) \in \mathbb{R}^n$ is the system state, $u(k) \in \mathbb{R}^m$ is the system input and $y(k) \in \mathbb{R}^p$ is the system output. In detail, we mainly identify two different circumstances in which GPs are employed to model dynamic systems:

1. The modeling of systems whose state $x(k)$ is known and accessible. In such cases, given that the usual control goal of an MPC is to control the system state, a GP can directly model the state function $f(\cdot, \cdot)$, employing a regressor $z(k) \coloneqq (x(k), u(k))$. This is the case of the works presented in [36, 37], even though the identified GPs are inserted in a reinforcement learning framework.

2. The modeling of systems whose state $x(k)$ is unknown, but output measures $y(k)$ are available. In such cases, a GP cannot directly model the state function $f(\cdot, \cdot)$. Therefore, a common way of addressing such issue in the MPC context is to act as follows. An equivalent *realigned system* is obtained by building an artificial state $x(k)$ composed of past output and input values, i.e. $x(k) \coloneqq (y(k), y(k-1), \ldots, y(k-n_a+1), u(k-1), \ldots, u(k-n_b))$, where n_a and n_b define

the order of the realigned system, as proposed in [93], and then GPs can be used to model the dynamics of such realigned system, as proposed in [67].

Regardless of whether the state of the system is known and accessible or it is artificially built by recurring to a realigned model, the function to be learnt is a function $f : \mathbb{R}^n \to \mathbb{R}^n$, and n is often such that $n > 1$. Therefore, another question that we need to answer is how GPs can be used to model vector-valued functions. This can be done in two ways, namely (i) training n independent GPs, each one modeling one of the functions $f_i(x(k), u(k))$, for all $i \in [1, n]$, as done in [99], or (ii) training only one multiple-output GP that directly models the entire function $f(\cdot)$, as in [9, 137]. An interesting review of kernel-based methods, e.g. GPs, used to model vector-valued functions is provided in [147]. While employing several one-dimensional GPs is usually simpler, the advantage of a multiple-output GP is that it is able to describe the coupling of the different states of the system, which is modeled by means of the covariances between the states.

4.3 Problem formulation

We consider a nonlinear time-invariant discrete-time system of the form

$$x(k+1) = f\,(x(k), u(k)) + \eta(k), \tag{4.7}$$

where $x(k) \in \mathbb{R}^n$ is the system state, $u(k) \in \mathbb{R}^m$ is the system input, and $\eta(k) := (\eta_1(k), \ldots, \eta_n(k))$, with η_i i.i.d. and such that $\eta_i(k) \sim \mathcal{N}\left(0, \sigma_{\eta_i}^2\right)$, with variances $\sigma_{\eta_i}^2 \geq 0$ and bounded supports $|\eta_i(k)| \leq \bar{\eta}_i < \infty$.

Remark 4.1

Note that $\sigma_{\eta_i}^2 = 0$ results in $\eta_i(k) = 0$, $\forall k \geq 0$, namely some state variables might not be affected by additive Gaussian noise. Moreover, even if Gaussian noise is theoretically characterized by an infinite support, from an engineering perspective it is reasonable to bound $|\eta_i(k)|$; in particular, the bounds $\bar{\eta}_i$ can be chosen to be arbitrarily large depending on the safety level that is desired.

The state function $f(x, u)$ has an equilibrium point in $(x, u) = (0, 0)$ and is considered to be unknown, while the state variables are assumed to be known and accessible.

Remark 4.2

From a mathematical perspective, the MPC formulation proposed in Section 4.5 could be used also to control systems whose state variables are not accessible, by employing a realigned model with state $x(k) = (y(k), \ldots, y(k - n_a), u(k-1), \ldots, u(k-n_b))$. Nevertheless, in the context presented here, the system is modeled by a GP; for this reason, if the system to be controlled has an accessible low-dimensional state, it is computationally convenient to directly exploit the known state variables.

The system is subject to hard state and input constraints, i.e.

$$x(k) \in \mathcal{X},\ u(k) \in \mathcal{U}, \quad \forall k \geq 0. \tag{4.8}$$

Before introducing a continuity assumption on $f(x,u)$, define the regressor $z := (x,u) \in \mathbb{R}^n \times \mathbb{R}^m$. With this notation, we can redefine the state equation as $x(k+1) = f\,(z(k)) + \eta(k)$.

Next, as done also in the previous chapters, we introduce a continuity assumption on $f : \mathbb{R}^n \times \mathbb{R}^m \longrightarrow \mathbb{R}^n$ and define the conditions that the sets of constraints $\mathcal{X}$ and $\mathcal{U}$ must verify.

Assumption 4.1: Lipschitz continuity of the latent function

The functions $f_i(z)$ are Lipschitz continuous in $\mathcal{X} \times \mathcal{U}$, with Lipschitz constants L_{f_i}, that is

$$|f_i(z) - f_i(\check{z})| \leq L_{f_i} \|z - \check{z}\|,\ \forall z, \check{z} \in \mathcal{X} \times \mathcal{U},\ \forall i \in [1,n]. \tag{4.9}$$

Remark 4.3

Considering that the functions $f_i(z)$ are required to be Lipschitz continuous only on a compact set, Assumption 4.1 is not particularly restrictive.

Assumption 4.2: Properties of the constraint sets

The sets of constraints $\mathcal{X}$ and $\mathcal{U}$ are compact and contain the origin in their interior.

A dataset $\mathcal{D}$ containing information about several operating conditions of the system is available. In particular, $\mathcal{D}$ is composed of a regressor matrix

$$Z := \begin{bmatrix} z_{(1)} & \ldots & z_{(n_{\mathcal{D}})} \end{bmatrix}^\top$$

and vectors $\mathrm{x}_i := \begin{bmatrix} x_{i,(1)} & \ldots & x_{i,(n_{\mathcal{D}})} \end{bmatrix}^\top$, $\forall i \in [1,n]$, such that $x_{i,(j)} = f_i(z_{(j)}) + \eta_{i,(j)}$, with $\eta_{i,(j)} \sim \mathcal{N}(0, \sigma^2_{\eta_i})$.

The goal is to drive the system state to the origin from a non-trivial initial condition $x_0 \neq 0$. In the next sections, we present our robust contraction-based GP-MPC. In particular, (i) we first provide the details about the conditions that the employed GPs must verify, then (ii) we present the GP-MPC and prove its stability and feasibility properties, and (iii) we finally introduce an alternative formulation of the GP-MPC that simplifies the computation of the control action to be applied.

4.4 Properties of the employed GP

We assume each state function $f_i(z)$ to be sampled from an independent $\mathcal{GP}(m_i(z), k_i(z, \check{z}))$, where $m_i(z)$ and $k_i(z, \check{z})$ denote the mean and covariance functions that define $f_i(z)$. For this reason, as the functions $f_i(z)$ are sampled from n independent GPs, and given the availability of the dataset $\mathcal{D}$, the GPs are trained using the same training regressor matrix Z, but different target vectors x_i. Moreover, the states $x_{1,(j)}, \ldots, x_{n,(j)}$ are assumed to be conditionally independent given the regressor $z_{(j)}$, that is

$$p\left(x_{i_1,(j)} | z_{(j)}, x_{i_2,(j)}, i_2 \neq i_1\right) = p\left(x_{i_1,(j)} | z_{(j)}\right).$$

By doing so, the posterior mean function $\mu(z(k)|\mathcal{D}) \coloneqq \big(\mu_1(z(k)|\mathcal{D}) \ \ldots \ \mu_n(z(k)|\mathcal{D})\big)$ can be used as predictor model of the estimated system state $\hat{x}(k+1)$. In detail,

$$\begin{aligned} \hat{x}_i(k+1) &= \hat{f}_i(x(k), u(k)) \coloneqq \mu_i(z(k)|\mathcal{D}) \\ &= m_i(z(k)) + k_i(Z, z(k))^\top \left[k_i(Z, Z) + \sigma_{\eta_i}^2 I_{n_\mathcal{D}}\right]^{-1} (\mathrm{x}_i - m_i(Z)), \end{aligned} \tag{4.10}$$

where $m_i(Z) \coloneqq (m_i(z_{(1)}), \ldots, m_i(z_{(n_\mathcal{D})}))$.
The estimated model $\hat{f}(x, u)$ allows us to define the prediction error as

$$\begin{aligned} w(k) &= \begin{bmatrix} w_1(k) & \ldots & w_n(k) \end{bmatrix}^\top \\ &\coloneqq \begin{bmatrix} f_1(x(k), u(k)) + \eta_1(k) - \hat{f}_1(x(k), u(k)) \\ \vdots \\ f_n(x(k), u(k)) + \eta_n(k) - \hat{f}_n(x(k), u(k)) \end{bmatrix}. \end{aligned} \tag{4.11}$$

Thanks to the definition of w, the state equation rewrites as

$$x(k+1) = \hat{f}(x(k), u(k)) + w(k). \tag{4.12}$$

To guarantee the consistency between (4.7) and (4.12), it is supposed that also with the new state-space formulation $\hat{f}(0, 0) = 0$.
In order to guarantee the robustness of the here-proposed MPC, we make the following assumption, which is needed to ensure that the functions $\hat{f}_i(x, u)$ are Lipschitz continuous.

Assumption 4.3: Lipschitz continuity of the mean and covariance functions

The covariance functions $k_i(z, \check{z})$ used to define the n GPs are Lipschitz continuous with Lipschitz constants L_{k_i}, and the mean functions $m_i(z)$ are Lipschitz continuous with Lipschitz constants L_{m_i}.

Before making a contractivity assumption on the nominal system, we first show how Assumptions 4.1-4.3 ensure that the prediction error w belongs to a bounded convex set and that there exist set sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ verifying Assumptions 3.4 and 3.5 from Chapter 3.

Lemma 4.1: Lipschitz continuity of the model

Under Assumption 4.3, there exist Lipschitz constants $L_{\hat{f}_i}$ such that

$$|\hat{f}_i(z) - \hat{f}_i(\check{z})| \leq L_{\hat{f}_i}\|z - \check{z}\|, \ \forall z, \check{z} \in \mathcal{X} \times \mathcal{U}, \ \forall i \in [1, n].$$

Proof. Theorem 3.1 in [81] proves that, using Lipschitz continuous covariance functions $k_i(z, \check{z})$ with Lipschitz constants L_{k_i}, and choosing as mean functions the zero-mean function $m_i(z|\mathcal{D}) = 0$, $\forall i \in [1, n]$, the posterior mean functions $\mu_i(z) \equiv \hat{f}_i(z)$ are Lipschitz continuous in $\mathcal{X} \times \mathcal{U}$ with Lipschitz constants

$$L_{\hat{f}_i,0} \leq L_{k_i}\sqrt{n_{\mathcal{D}}} \ \| \left[k_i(Z, Z) + \sigma_{\eta_i}^2 I_{n_{\mathcal{D}}}\right]^{-1} \mathrm{x}_i\|.$$

This means that, for all $i \in [1, n]$, and $\forall z, \check{z} \in \mathcal{X} \times \mathcal{U}$,

$$\left|k_i(Z, z)^\top \left[k_i(Z, Z) + \sigma_{\eta_i}^2 I_{n_{\mathcal{D}}}\right]^{-1} \mathrm{x}_i - k_i(Z, \check{z})^\top \left[k_i(Z, Z) + \sigma_{\eta_i}^2 I_{n_{\mathcal{D}}}\right]^{-1} \mathrm{x}_i\right| \leq L_{\hat{f}_i,0}\|z - \check{z}\|.$$

Furthermore, as different covariance function Lipschitz constants $L_{\hat{f}_i,0}$ can be found for different target vectors x_i, it is still possible to find finite Lipschitz constants for the target vectors $\tilde{\mathrm{x}}_i := \mathrm{x}_i - m_i(Z)$. Moreover, given that the sum of Lipschitz continuous functions is Lipschitz continuous, also in the case of generic Lipschitz continuous mean functions $m_i(z)$, there exist constants $L_{\hat{f}_i}$ such that

$$\begin{aligned}
&|m_i(z) + k_i(Z, z)^\top \left[k_i(Z, Z) + \sigma_{\eta_i}^2 I_{n_{\mathcal{D}}}\right]^{-1} (\mathrm{x}_i - m_i(Z)) \\
&-m_i(\check{z}) - k_i(Z, \check{z})^\top \left[k_i(Z, Z) + \sigma_{\eta_i}^2 I_{n_{\mathcal{D}}}\right]^{-1} (\mathrm{x}_i - m_i(Z))| \\
&\leq L_{\hat{f}_i}\|z - \check{z}\|.
\end{aligned}$$

■

Robust MPC formulations usually assume that the perturbations are bounded. The next lemma shows that Assumptions 4.1 and 4.3 are enough to ensure that the model error $w(k)$ is bounded.

Lemma 4.2: Boundedness of the prediction error

Under Assumptions 4.1 and 4.3, the prediction error $w(k)$ is bounded. Specifically, $w(k) \in \mathcal{W}$, $\forall k \geq 0$, where $\mathcal{W}$ is a compact set containing the origin in its interior and is defined as

$$\mathcal{W} := \{w \in \mathbb{R}^n : \ |w_i| \leq \bar{w}_i, \ i \in [1, n]\},$$

with $\bar{w}_i > 0$, $\forall i \in [1, n]$.

Proof. Consider the definition of the prediction error provided in (4.11), and analyze each term $f_i(x(k), u(k)) + \eta_i(k) - \hat{f}_i(x(k), u(k))$. In particular, note that

$$\begin{aligned} f_i(x(k), u(k)) + \eta_i(k) - \hat{f}_i(x(k), u(k)) &\leq |f_i(x(k), u(k)) + \eta_i(k) - \hat{f}_i(x(k), u(k))| \\ &\leq |f_i(x(k), u(k))| + |\eta_i(k)| + |\hat{f}_i(x(k), u(k))|. \end{aligned}$$

However, $|\eta_i(k)| \leq \bar{\eta}_i < \infty$. Moreover, thanks to Assumptions 4.1 and 4.3, $f_i(x, u)$ and $\hat{f}_i(x, u)$ are Lipschitz continuous. Thus, as $\mathcal{X}$ and $\mathcal{U}$ are compact, it follows that $|f_i(x, u)| < \infty$ and $|\hat{f}_i(x, u)| < \infty$ in $\mathcal{X} \times \mathcal{U}$. This allows for the conclusion that, $\forall i \in [1, n]$, there exists $\bar{w}_i > 0$ such that $|f_i(x(k), u(k)) + \eta_i(k) - \hat{f}_i(x(k), u(k))| \leq \bar{w}_i$. Therefore, there also exists a compact set $\mathcal{W}$ defined as

$$\mathcal{W} := \{w \in \mathbb{R}^n : |w_i| \leq \bar{w}_i, i \in [1, n]\},$$

such that $w(k) \in \mathcal{W}$, $\forall k \geq 0$. ■

Remark 4.4

From a practical point of view, the volume of $\mathcal{W}$ can be reduced via an appropriate choice of the mean and covariance functions, and the availability of data points that well represent the behavior of the system.

Lemma 4.1 proves that the functions $\hat{f}_i(z)$ are Lipschitz continuous in $\mathcal{X} \times \mathcal{U}$. However, as reported in [102], Lipschitz continuity is a sufficient condition to prove also component-wise Lipschitz continuity in $\mathcal{X} \times \mathcal{U}$. Moreover, given that $w(k)$ appears in (4.12) as an additive perturbation, we can state that (4.12) is component-wise Lipschitz continuous in $\mathcal{X} \times \mathcal{U} \times \mathcal{W}$. In detail, $\forall i \in [1, n]$, there exist non-negative real constants $L_{x,ia}$ and $L_{u,ib}$ such that

$$\begin{aligned} |\hat{f}_i(x, u) + w_i - \hat{f}_i(\check{x}, \check{u}) - \check{w}_i| \leq& \sum_{a=1}^{n} L_{x,ia}|x_a - \check{x}_a| \\ &+ \sum_{b=1}^{m} L_{u,ib}|u_b - \check{u}_b| + |w_i - \check{w}_i|. \end{aligned} \tag{4.13}$$

As the GP-MPC belongs to the family of tube-based MPC formulations, we now show how the component-wise Lipschitz continuity of (4.12) implies properties that can be exploited to build the tightened constraints and to prove recursive feasibility of the GP-MPC.

Lemma 4.3: Bounds on the effect of the initial condition on the model state trajectory

Under Assumptions 4.1 and 4.3, there exists a sequence of sets $\{\mathcal{F}(j)\}_{j\geq 0}$ such that:

- $\mathcal{F}(0) := \{x \in \mathbb{R}^n : |x_i| \leq \bar{w}_i, \forall i \in [1, n]\}$;
- *for every* $(x, \mathbf{u}) \in \mathcal{X} \times \mathcal{U}^j$, *the sets* $\mathcal{F}(j)$, *with* $j > 0$, *ensure that* $\phi(j; \check{x}, \mathbf{u}, \mathbf{0}) \in \phi(j; x, \mathbf{u}, \mathbf{0}) \oplus \mathcal{F}(j)$, *for all* $\check{x}$ *such that,* $\check{x} - x \in \mathcal{F}(0)$.

Proof. First, define the worst-case prediction error $\check{w} := [\bar{w}_1, \ \ldots, \ \bar{w}_n]^\top$, and define the $\mathcal{K}_\infty$ functions $c_{i,0}(\cdot)$, such that $c_{i,0}(\check{w}) = \bar{w}_i$. In this way, the inequalities $|x_i - \check{x}_i| \leq \bar{w}_i$, $\forall i \in [1, n]$, can be rewritten as $|x_i - \check{x}_i| \leq c_{i,0}(\check{w})$.

Considering a feasible control sequence $\mathbf{u} \in \mathcal{U}^j$, and due to the component-wise Lipschitz continuity of the functions $\hat{f}_i(x, u) + w$, it holds that

$$
\begin{aligned}
|\phi_i(1; x, \mathbf{u}, \mathbf{0}) - \phi_i(1; \check{x}, \mathbf{u}, \mathbf{0})| &\leq \sum_{a=1}^{n} L_{x,ia} \bar{w}_a \\
&= \sum_{a=1}^{n} L_{x,ia} c_{a,0}(\check{w}) =: c_{i,1}(\check{w}).
\end{aligned}
$$

Proceeding with what happens after two steps, we find that

$$
\begin{aligned}
|\phi_i(2; x, \mathbf{u}, \mathbf{0}) - \phi_i(2; \check{x}, \mathbf{u}, \mathbf{0})| &\leq \sum_{a=1}^{n} L_{x,ia} |\phi_a(1; x, \mathbf{u}, \mathbf{0}) - \phi_a(1; \check{x}, \mathbf{u}, \mathbf{0})| \\
&= \sum_{a=1}^{n} L_{x,ia} c_{a,1}(\check{w}) =: c_{i,2}(\check{w}).
\end{aligned}
$$

Therefore, generalizing for all $j > 0$, we obtain that

$$
|\phi_i(j; x, \mathbf{u}, \mathbf{0}) - \phi_i(j; \check{x}, \mathbf{u}, \mathbf{0})| \leq \sum_{a=1}^{n} L_{x,ia} c_{a,j-1}(\check{w}) =: c_{i,j}(\check{w}).
$$

Thus, the set sequence $\{\mathcal{F}(j)\}_{j\geq 0}$, with

$$
\mathcal{F}(j) := \{x \in \mathbb{R}^n : |x_i| \leq c_{i,j}(\check{w}), \forall i \in [1, n]\},
$$

verifies the condition $\phi(j; \check{x}, \mathbf{u}, \mathbf{0}) \in \phi(j; x, \mathbf{u}, \mathbf{0}) \oplus \mathcal{F}(j)$, for all $j \geq 0$. ■

Lemma 4.4: Bounds on the effect of the prediction error on the model state trajectory

Under Assumptions 4.1 and 4.3, there exists a sequence of sets $\{\mathcal{R}(j)\}_{j\geq 0}$, *with* $\mathcal{R}(0) := \{0\}$, *that verifies the following conditions simultaneously:*

1. *for every* $(x, \mathbf{u}) \in \mathcal{X} \times \mathcal{U}^j$, *the sets* $\mathcal{R}(j)$, *for* $j > 0$, *ensure* $\phi(j; x, \mathbf{u}, \mathbf{w}) \in \phi(j; x, \mathbf{u}, \mathbf{0}) \oplus \mathcal{R}(j)$;
2. $\mathcal{F}(j) \oplus \mathcal{R}(j) \subseteq \mathcal{R}(j+1)$, $\forall j \geq 0$.

Proof. First, define the functions $d_{i,0}(\cdot)$ as zero functions such that $d_{i,0}(w) = 0$, $\forall w \in \mathcal{W}$ and $\forall i \in [1, n]$, and the functions $d_{i,j}(\cdot) := \sum_{k=0}^{j-1} c_{i,k}(\cdot)$, $\forall j > 0$. Then, analyzing the gap between $\phi(1; x, \mathbf{u}, \mathbf{w})$ and $\phi(1; x, \mathbf{u}, \mathbf{0})$. In detail, we have

$$|\phi(1; x, \mathbf{u}, \mathbf{w}) - \phi(1; x, \mathbf{u}, \mathbf{0})| \leq \bar{w}_i \equiv c_{i,0}(\breve{w}) = d_{i,1}(\breve{w}).$$

We now analyze the gap between $\phi(2; x, \mathbf{u}, \mathbf{w})$ and $\phi(2; x, \mathbf{u}, \mathbf{0})$:

$$\begin{aligned} |\phi(2; x, \mathbf{u}, \mathbf{w}) - \phi(2; x, \mathbf{u}, \mathbf{0})| &\leq \bar{w}_i + \sum_{a=1}^{n} L_{x,ia} c_{a,0}(\breve{w}) \\ &= c_{i,0}(\breve{w}) + c_{i,1}(\breve{w}) = d_{i,2}(\breve{w}). \end{aligned}$$

Generalizing for all $j > 0$, we obtain

$$|\phi(j; x, \mathbf{u}, \mathbf{w}) - \phi(j; x, \mathbf{u}, \mathbf{0})| \leq \sum_{k=0}^{j-1} c_{i,k}(\breve{w}) = d_{i,j}(\breve{w}).$$

Therefore, the set sequence $\{\mathcal{R}(j)\}_{j\geq 0}$, where

$$\mathcal{R}(j) := \{x \in \mathbb{R}^n : |x_i| \leq d_{i,j}(\breve{w}), \forall i \in [1, n]\},$$

verifies the first condition of the lemma. We now need to verify that $\mathcal{F}(j) \oplus \mathcal{R}(j) \subseteq \mathcal{R}(j+1)$. Note that, as the sets $\mathcal{F}(j)$ and $\mathcal{R}(j)$ are both n-dimensional boxes, the sets $\mathcal{F}(j) \oplus \mathcal{R}(j)$ can be easily computed as

$$\mathcal{F}(j) \oplus \mathcal{R}(j) = \{x \in \mathbb{R}^n : |x_i| \leq c_{i,j}(\breve{w}) + d_{i,j}(\breve{w}), \forall i \in [1, n]\}.$$

As $c_{i,j}(\breve{w}) + d_{i,j}(\breve{w}) \equiv d_{i,j+1}(\breve{w})$, it follows that $\mathcal{F}(j) \oplus \mathcal{R}(j) = \mathcal{R}(j+1)$, which ends the proof. ■

The sequence $\{\mathcal{F}(j)\}_{j\geq 0}$ bounds the gap between the perturbation-free state trajectories obtained with the same control sequence starting from close enough states x and $\check{x}$, while the sequence $\{\mathcal{R}(j)\}_{j\geq 0}$ bounds the gap between the perturbed and the perturbation-free trajectories obtained starting from a common state x and applying the same control sequence.

Remark 4.5

The proofs of Lemmas 4.3 and 4.4 also provide a method to compute the set sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ *and* $\{\mathcal{R}(j)\}_{j\geq 0}$.

As the proof of stability of our robust GP-MPC is based on the contractivity of the nominal system, i.e. the employed GPs, a non-restrictive contractivity assumption is necessary.

Assumption 4.4: Contractivity of the nominal system

There exist a positive definite function $\Gamma(x)$*, a contraction factor* $\gamma \in (0,1)$*, and a prediction horizon* $N_p \in \mathbb{N} \setminus \{0\}$ *such that, for all the initial conditions* $x \in \mathcal{X}$*, there exists a control sequence* $\mathbf{u} \in \mathcal{U}^{N_p}$ *that verifies:*

$$\underline{\Gamma}(x, \mathbf{u}, N_p) \coloneqq \min_{j=1}^{N_p} \Gamma\left(\phi(j; x, \mathbf{u}, \mathbf{0})\right) \leq \gamma \Gamma(x). \tag{4.14}$$

Moreover, the contractive function $\Gamma(x)$ *is uniformly continuous in* $\mathcal{X}$*, i.e. there exists a* $\mathcal{K}$ *function* $\sigma_\Gamma(\cdot)$ *such that*

$$|\Gamma(x) - \Gamma(\check{x})| \leq \sigma_\Gamma\left(\|x - \check{x}\|\right), \ \forall x, \check{x} \in \mathcal{X}, \tag{4.15}$$

and is upper-bounded by a $\mathcal{K}_\infty$ *function* $\alpha_\Gamma(\cdot)$*, such that*

$$\Gamma(x) \leq \alpha_\Gamma(\|x\|), \ \forall x \in \mathcal{X}. \tag{4.16}$$

Finally, the set $\mathcal{X} \ominus \mathcal{R}(N_p)$ *is non-empty and contains the origin in its interior.*

Henceforward, the minimizer of (4.14) over a prediction horizon of length N_p is denoted $j_{\text{opt}}(x, \mathbf{u}, N_p)$.

Remark 4.6

The choice of the candidate $\Gamma(\cdot)$ *and the computation of* γ *and* N_p *can be carried out following the same suggestions and algorithms shown in Section 3.7.1.*

The recursive feasibility of the GP-MPC is guaranteed by the knowledge of an RPI, exactly like for the MPC presented in Chapter 3. Therefore, we make also the following assumption.

Assumption 4.5: Existence of an RPI

There exists a set $\Omega \subseteq \mathcal{X}$ *that verifies the following conditions simultaneously:*

1. *for all* $x \in \Omega$*,* $\exists u \in \mathcal{U}$ *such that* $\hat{f}(x,u) + w \in \Omega$*, for any* $w \in \mathcal{W}$*;*

2. $\Omega \subseteq \mathcal{Q}\,(\mathcal{X} \ominus \mathcal{R}(1))$ *for the nominal system* $\hat{x}(k+1) = \hat{f}(\hat{x}(k), u(k))$*, where* $\mathcal{Q}\,(\mathcal{X} \ominus \mathcal{R}(1))$ *denotes the one-step controllable set to* $\mathcal{X} \ominus \mathcal{R}(1)$*;*

3. $\Omega \ominus \mathcal{R}(1)$ *is non-empty and contains the origin.*

Remark 4.7

Assumption 4.5 requires Ω *to be an admissible RPI. The computation of such set for nonlinear systems is still a topic of discussion in the control system community; for low-dimensional nonlinear systems, an effective graph-based method to find outer approximations of such sets is presented in [34].*

4.5 Robust contractive GP-MPC

In this subsection, the proposed GP-MPC is presented, its design parameters are explained and its stability and feasibility properties are proved.

We begin by designing the cost function. We suppose to express the control objective by means of a stage cost function $\ell(x, u)$. In detail, given an initial time instant k and a prediction horizon $q(k)$, the total cost related to the control objective is given by

$$J\big(x(k), \mathbf{u}(k), q(k)\big) := \sum_{j=0}^{q(k)-1} \ell(\hat{x}(j|k), u(j|k)), \tag{4.17}$$

where $\mathbf{u}(k) := \{u(0|k), \dots, u(q(k)-1|k)\}$ and $\hat{x}(j|k)$ denotes the nominal j-step-ahead state prediction computed starting from the initial condition $x(k)$, i.e. $\hat{x}(j|k) := \phi(j; x(k), \mathbf{u}(k), \mathbf{0})$. The stage cost function $\ell(x, u)$ is also required to verify the (non-restrictive) conditions of the following assumption.

Assumption 4.6: Stage cost function properties

There exist a constant $\bar{\ell} > 0$ *and a* $\mathcal{K}$ *function* $\alpha_\ell(\cdot)$ *such that*

$$\forall (x, u) \in \mathcal{X} \times \mathcal{U}, \quad 0 \le \ell(x, u) \le \bar{\ell}, \tag{4.18}$$

and $\alpha_\ell(\|x\|) \le \ell(x, u),\ \forall u \in \mathcal{U}$*.*

Moreover, the stage cost function $\ell(x, u)$ *is uniformly continuous with respect to all its arguments; therefore, there exist* $\mathcal{K}$ *functions* $\sigma_{\ell,x}(\cdot)$ *and* $\sigma_{\ell,u}(\cdot)$ *such that,* $\forall x, \check{x} \in \mathcal{X},\ u, \check{u} \in \mathcal{U}$*:*

$$|\ell(x, u) - \ell(\check{x}, \check{u})| \le \sigma_{\ell,x}\left(\|x - \check{x}\|\right) + \sigma_{\ell,u}\left(\|u - \check{u}\|\right). \tag{4.19}$$

The optimization problem $P_{N_p}(x(k),\theta(k))$ that has to be solved at each time instant k is

$$\min_{(\mathbf{u}(k),q(k))} V_{N_p}(x(k),\theta(k),\mathbf{u}(k),q(k)) := \theta(k)\cdot J(x(k),\mathbf{u}(k),q(k)) + \xi\cdot\underline{\Gamma}(x(k),\mathbf{u}(k),q(k))$$

subject to :

$$\hat{x}(0|k) = x(k), \tag{4.20a}$$

$$\hat{x}(j|k) = \hat{f}(\hat{x}(j-1|k),u(j-1|k)),\ \forall j\in[1,q(k)], \tag{4.20b}$$

$$\hat{x}(j|k) \in \mathcal{X}\ominus\mathcal{R}(j),\ \forall j\in[0,q(k)], \tag{4.20c}$$

$$(\mathbf{u}(k),q(k)) \in \mathcal{U}^{q(k)}\times\{1,\dots,N_p\} \tag{4.20d}$$

where:

- $\mathbf{u}(k) := \{u(0|k),\dots,u(q(k)-1|k)\}$;
- $q(k)$ is the prediction horizon of the controller, which is a decision variable and can take values within the range $[1,N_p]$;
- $\theta(k)$ is an internal state of the MPC controller, which plays the same role of the internal state introduced in the contraction-based MPC presented in Chapter 3.

If multiple solutions providing the same cost exist, the one having the smallest q is kept. The optimal solutions of $P_{N_p}(x(k),\theta(k))$ are denoted as $\mathbf{u}^*(x(k),\theta(k))$ and $q^*(x(k),\theta(k))$, or $\mathbf{u}^*(k)$ and $q^*(k)$ for brevity, and the corresponding values $\underline{\Gamma}$, J, V_{N_p} and j_{opt} are:

$$\underline{\Gamma}^*(x(k),\theta(k)) := \underline{\Gamma}(x(k),\mathbf{u}^*(k),q^*(k)) \tag{4.21}$$

$$J^*(x(k),\theta(k)) := J(x(k),\mathbf{u}^*(k),q^*(k)) \tag{4.22}$$

$$V^*_{N_p}(x(k),\theta(k)) := V_{N_p}(x(k),\theta(k),\mathbf{u}^*(k),q^*(k)) \tag{4.23}$$

$$j^*(x(k),\theta(k)) := j_{\text{opt}}(x(k),\mathbf{u}^*(k),q^*(k)). \tag{4.24}$$

We also define $\hat{\mathbf{x}}^*(k) := \{\hat{x}^*(0|k),\dots,\hat{x}^*(q^*(k)|k)\}$ as the nominal state sequence provided by the optimal control sequence $\mathbf{u}^*(x(k),\theta(k))$, that is $\hat{x}^*(j|k) = \phi(j;x(k),\mathbf{u}^*(k),\mathbf{0})$. The control law related to the GP-MPC is the following: $\kappa_{N_p}(x(k),\theta(k)) := u^*(0|k)$, where $u^*(0|k)$ is the first element of the optimal control sequence $\mathbf{u}^*(k)$ obtained by solving $P_{N_p}(x(k),\theta(k))$.
The initial condition of the controller state θ is set to $\theta(0) = \max\{\epsilon,\Gamma(x(0))\}$, where $\epsilon > 0$ and $\nu\in(0,1)$ are design parameters, and is updated at every time instant k by using the following policy:

$$\theta(k) = f_\theta(x(k),\theta(k-1)) := \begin{cases}\theta(k-1), & \Gamma(x(k)) > \theta(k-1)\\ \max\{\epsilon,\Gamma(x(k))\}, & \text{otherwise}\end{cases} \tag{4.25}$$

Before continuing, we define $\omega > 0$ as a constant such that $\{x : \Gamma(x) \leq \omega\} \subseteq \Omega \ominus \mathcal{R}(1)$.

We are now ready to introduce the theorem that proves that the origin is input-to-state practically stable ISpS for the closed-loop system under the here-proposed GP-MPC.

Theorem 4.1: Stability of the robust contractive GP-MPC for the regulation of Lipschitz continuous nonlinear systems

Under Assumptions 4.1-4.6, if the following conditions hold simultaneously:

1. $\xi \geq 2N_p\bar{\ell}/(1-\gamma)$,
2. $\theta(k) = f_\theta\left(x(k), \theta(k-1)\right)$,
3. $\gamma \leq \frac{\omega}{\Gamma_{\max}}$,

then, given a feasible initial condition $x \in \mathcal{X}$, $P_{N_p}\left(x,\theta\right)$ is recursively feasible and the equilibrium $x = 0$ is ISpS for the closed-loop system.

Proof. First, note that Assumptions 4.4, 4.5 and 4.6 are respectively equivalent to Assumptions 3.3, 3.6 and 3.7 from Chapter 3. Furthermore, under Assumptions 4.1 and 4.3, all the conditions of the Lemmas 4.2, 4.3 and 4.4 are verified. Therefore, we get that the set $\mathcal{W}$ is bounded and the sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$ exist. The last assumption that Chapter 3 makes about the system under control is that its state equations are component-wise uniformly continuous. Given that component-wise Lipschitz continuity is a stronger condition, and provided that the conditions on ξ, $\theta(k)$ and γ are identical to those employed in Chapter 3, all the conditions of Theorem 3.1 from Chapter 3 are verified. As a consequence, the origin is ISpS for the closed-loop system under our robust contraction-based GP-MPC. ■

4.6 A contractive GP-MPC formulation with two optimization problems

The MPC formulation introduced in (4.20) involves the integer variable q, which could make the resolution of the problem computationally demanding. In this section, an alternative two-stage algorithm without integer decision variables is introduced.

The two steps of the newly proposed algorithm are the following:

1. First, a fixed-horizon problem is solved:

$$\begin{aligned}\min_{\mathbf{u}(k)}\ & \underline{\Gamma}(x(k),\mathbf{u}(k),N_p)\\ \text{s.t. } & \hat{x}(0|k)=x(k),\\ & \hat{x}(j|k)=\hat{f}\left(\hat{x}(j-1|k),u(j-1|k)\right),\forall j\in[1,N_p],\\ & \hat{x}(j|k)\in\mathcal{X}\ominus\mathcal{R}(j),\ \forall j\in[0,N_p],\\ & \mathbf{u}(k)\in\mathcal{U}^{N_p}\end{aligned}\tag{4.26}$$

and the instant $j_{N_p}\in[1,N_p]$ of maximum contraction is memorized.

2. A second fixed-horizon optimization problem with prediction horizon equal to j_{N_p} is solved to get the desired optimal control sequence:

$$\begin{aligned}\min_{\mathbf{u}(k)}\ & V_{N_p}\left(x(k),\theta(k),\mathbf{u}(k),j_{N_p}\right)\\ \text{s.t. } & \hat{x}(0|k)=x(k),\\ & \hat{x}(j|k)=\hat{f}\left(\hat{x}(j-1|k),u(j-1|k)\right),\forall j\in[1,j_{N_p}],\\ & \hat{x}(j|k)\in\mathcal{X}\ominus\mathcal{R}(j),\ \forall j\in[0,j_{N_p}],\\ & \mathbf{u}(k)\in\mathcal{U}^{j_{N_p}}.\end{aligned}\tag{4.27}$$

Remark 4.8

The computational cost of this two-stage formulation can be limited by using the first j_{N_p} elements of the solution to (4.26) *as initial candidate solution to problem* (4.27)*, as proposed also in Chapter 3.*

This formulation inherits all the stability and robustness properties of the one introduced before, as proved by the next theorem.

Theorem 4.2: Stability of the two-stage formulation of the robust contractive GP-MPC for the regulation of Lipschitz continuous nonlinear systems

If the following conditions hold:

1. *Assumptions 4.1-4.6 are satisfied;*
2. *the penalty ξ involved in the second optimization problem satisfies $\xi\geq 2N_p\bar{\ell}/(1-\gamma)$;*
3. *the rule $\theta(k)=f_\theta(x(k),\theta(k-1))$ is used to update the value of θ,*
4. $\gamma\leq\frac{\omega}{\Gamma_{\max}}$,

then, given a feasible initial condition $x \in \mathcal{X}$, the 2-stage version of $P_{N_p}(x, \theta)$ is recursively feasible and the equilibrium $x = 0$ is ISpS for the closed-loop system.

Proof. Theorem 3.2 in Chapter 3 proves that the two-stage formulation of our robust contraction-based MPC keeps the same stability and recursive feasibility properties of the all-in-one formulation, provided that the same conditions on the system under control and the design parameters are verified. Given that also the conditions of Theorem 4.2 are identical to those listed in Theorem 4.1, we can state that the origin is ISpS for the closed-loop system also under the two-stage formulation of our robust contraction-based GP-MPC. ∎

4.7 Case study

In order to validate the proposed method, the designed contractive GP-MPC and a standard nonlinear MPC are implemented to control a continuous stirred-tank reactor (CSTR) whose state is subject to additive Gaussian noise, the details of which are given in the next subsection, and their performances are compared. In particular, their operation is simulated 50 times over a simulation horizon of 30 minutes, which is equal to 60 time steps, given that the sampling time of the system is set to $T_s = 30$ [s].

4.7.1 System description and control problem definition

In a CSTR, a reactant A is converted into a product B. The base model of the reaction is given by the following differential equations:

$$\frac{dC_A(t)}{dt} = \frac{q_0}{V}\big(C_{Af} - C_A(t)\big) - k_0 \exp\left(-\frac{E}{RT(t)}\right)C_A(t), \tag{4.28a}$$

$$\frac{dT(t)}{dt} = \frac{q_0}{V}\big(T_f - T(t)\big) - \frac{\Delta H_r k_0}{\rho C_p}\exp\left(-\frac{E}{RT(t)}\right)C_A(t) + \frac{UA}{V\rho C_p}\big(T_c(t) - T(t)\big), \tag{4.28b}$$

$$\frac{dT_c(t)}{dt} = \frac{T_r(t) - T_c(t)}{\tau_c}, \tag{4.28c}$$

where $C_A\,[\text{mol l}^{-1}]$ is the concentration of the reactant, $T_r\,[\text{K}]$ is the reference of the coolant temperature, $T\,[\text{K}]$ is the temperature of the tank and $T_c\,[\text{K}]$ is the temperature of the coolant. The input of the system is the reference of the coolant temperature ($u = T_r$). The model parameters are shown in Table 16.

For simulation purposes, the system (4.28) is discretized using a fifth-order Runge-Kutta method with sampling time equal to T_s, and i.i.d. Gaussian noise is added to emulate the effect of unknown

Parameter	Definition	Value
q_0	Input flow of the reactive	10 l/min
V	Liquid volume in the tank	150 l
k_0	Frequency constant	$6 \times 10^{10}\text{min}^{-1}$
E/R	Arrhenius constant	9750 K
$-\Delta H_r$	Enthalpy of the reaction	10000 J mol^{-1}
UA	Heat transfer coefficient	70000 J $\text{min}^{-1}\text{K}^{-1}$
ρ	Density	1100 gl^{-1}
C_p	Specific heat	0.3 $\text{Jg}^{-1}\text{K}^{-1}$
τ_c	Time constant	1.5 min
C_{Af}	C_A in the input flow	1 mol l^{-1}
T_f	Temperature (input flow)	370 K

Table 16: CSTR parameters.

dynamics on the evolution of C_A and T. In detail,

$$\begin{bmatrix} C_A \\ T \\ T_c \end{bmatrix}(k+1) = \text{RK}_5\left(\begin{bmatrix} C_A \\ T \\ T_c \end{bmatrix}(k \cdot T_s), T_s\right) + \begin{bmatrix} \eta_1(k) \\ \eta_2(k) \\ 0 \end{bmatrix} \tag{4.29}$$

where $\text{RK}_5([C_A, T, T_c]^\top(k \cdot T_s), T_s)$ denotes the solution provided by the fifth-order Runge-Kutta method starting from the initial condition $[C_A, T, T_c]^\top(k \cdot T_s)$ and integrating over the time-span $[k \cdot T_s, (k+1) \cdot T_s]$, $\eta_1(k) \sim \mathcal{N}(0, 0.003^2)$ with $|\eta_1(k)| \leq 0.012$, and $\eta_2(k) \sim \mathcal{N}(0, 0.05^2)$ with $|\eta_2(k)| \leq 0.2$.

The system input and state variables are subject to hard constraints; in particular, the reference of the coolant temperature T_r, which is the input of the system, must verify $T_r(k) \in [335, 372]$K, $\forall k \geq 0$, while the states C_A and T must verify $C_A(k) \in [0.2, 0.8053]$mol/l and $T(k) \in [335, 372]$K, $\forall k \geq 0$.

The objective is to steer the system state from the initial condition

$$\big(C_A(0), T(0), T_c(0)\big) = (0.778, 338.850, 337)$$

to the equilibrium point $(0.439, 357.43, 356)$, verifying the state and input constraints.

4.7.2 Contractive GP-MPC design

The contractive GP-MPC is implemented in its two-stage formulation and is designed assuming to have access only to the state variables C_A and T; therefore, we define the state and input of the system as $x = (x_1, x_2) := (C_A, T)$ and $u = T_r$. For this reason, using the same notation as in Section 4.3, we define the regressor $z = (x, u)$.

Remark 4.9

We assume to know and have access to only two state variables of the system to simulate the realistic case in which the here-proposed controller is supposed to stabilize a system having incomplete knowledge about its state variables.

In order to implement the contractive GP-MPC, it is first necessary to collect the dataset that is then used to fit the GPs for x_1 and x_2, and to find the perturbation set $\mathcal{W} \subset \mathbb{R}^2$. This dataset is generated by applying a sequence of chirp signals with different frequencies, amplitudes and centers. The input and state trajectories of the data collection phase are shown in Figure 15. Then, all the variables are normalized to stay in the range $[0, 1]$. For this reason, the state and input constraint sets are defined as

$$\mathcal{X} \coloneqq [0,1] \times [0,1], \quad \mathcal{U} \coloneqq [0,1].$$

The final dataset is then shuffled and split in a training set, composed of around the 70% of the data entries, that is then used to train the two GPs, and a validation set with all the remaining data entries, which is used to compute $\mathcal{W}$.

Defining the state equations of the real system as $x_1(k+1) = f_1(z(k)) + \eta_1(k)$ and $x_2(k+1) = f_2(z(k)) + \eta_2(k)$, $f_1(\cdot)$ and $f_2(\cdot)$ are assumed to be respectively sampled from $\mathcal{GP}(m_1(z), k_1(z, \check{z}))$ and $\mathcal{GP}(m_2(z), k_2(z, \check{z}))$, where the prior mean functions are set as constant-mean functions $m_1(z) = m_1$ and $m_2(z) = m_2$, with $m_1, m_2 \in \mathbb{R}$, while the two covariance functions are chosen as SE-ARD, i.e.

$$k_i(z, \check{z}) = \sigma_{i,f}^2 \cdot \exp\left(-\frac{1}{2}\sum_{k=1}^{3}\frac{z_k - \check{z}_k}{l_{i,k}^2}\right), \ \forall i \in \{1,2\}.$$

The hyperparameter values $\{m_1, \sigma_{1,f}, l_{1,1}, l_{1,2}, l_{1,3}, \sigma_{\eta_1}\}$ and $\{m_2, \sigma_{2,f}, l_{2,1}, l_{2,2}, l_{2,3}, \sigma_{\eta_2}\}$ are found via maximization of the log marginal likelihood on a subset of the available dataset, which is composed of 83 data entries selected by maximizing their relative distance. The final hyperparameter values are reported in Table 17.

m_1	$\sigma_{1,f}$	$l_{1,1}$	$l_{1,2}$	$l_{1,3}$	σ_{η_1}
−0.1959	1.2587	2.6573	3.8371	367.7207	0.0049
m_2	$\sigma_{2,f}$	$l_{2,1}$	$l_{2,2}$	$l_{2,3}$	σ_{η_2}
0.3860	1.2140	27.3806	2.4554	8.0650	0.0164

Table 17: Hyperparameters of the GPs used to model $f_1(x, u)$ and $f_2(x, u)$.

The found hyperparameters and the selected subset of the training data, which we denote by $\mathcal{D}$, are then used to compute $\mathcal{W}$. In particular, as $\mathcal{W} \coloneqq \{w \in \mathbb{R}^2 : |w_i| \leq \bar{w}_i, \forall i \in \{1,2\}\}$, the computation of $\mathcal{W}$ translates into finding $\bar{w}_1$ and $\bar{w}_2$, which are computed as the maximum absolute prediction error

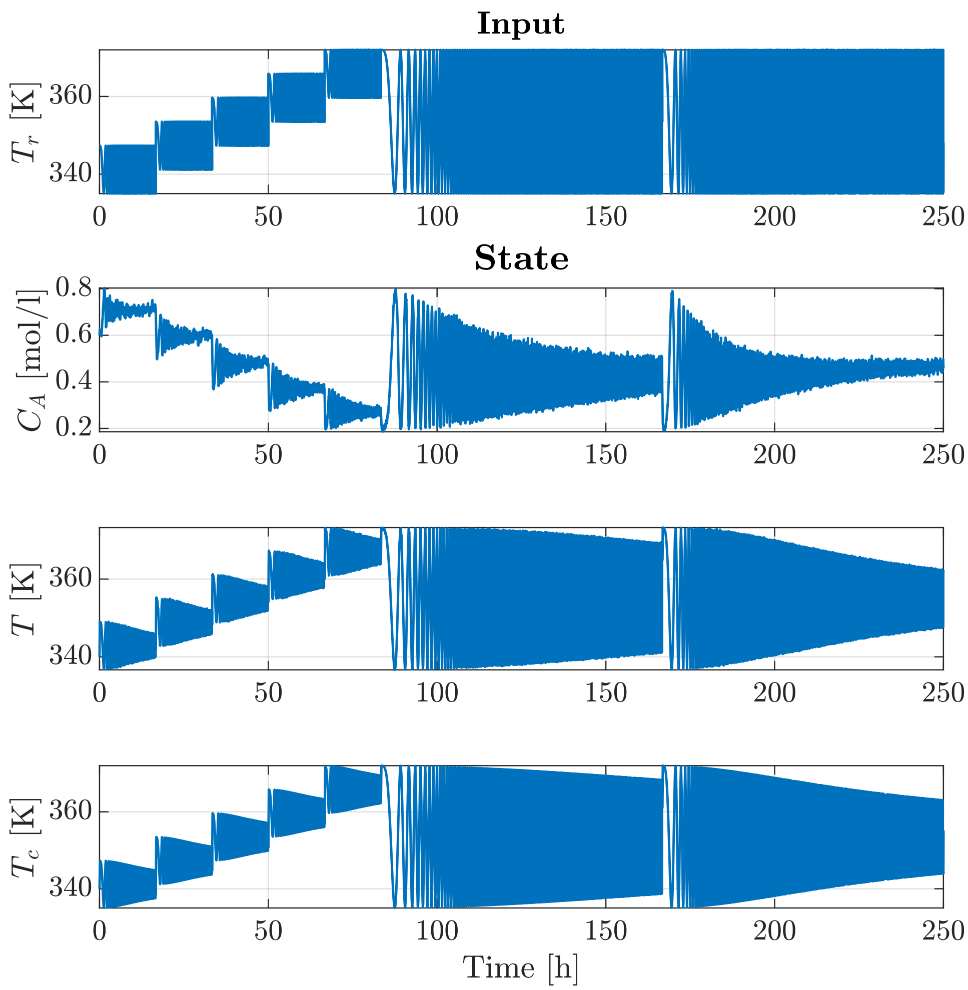


Figure 15: Input and state trajectories during the data collection phase. The choice of employing chirp input signals instead of the widely used pseudo-random signals is motivated by the fact that the latter tend to provide more information about the dynamics of the system around the equilibrium points, while chirp signals tend to provide information also about non-equilibrium points [101].

of the two estimated GPs on the validation set. The obtained values are $\bar{w}_1 = 0.0214$ and $\bar{w}_2 = 0.0515$, which have to be considered as normalized values, given that $x \in [0, 1] \times [0, 1]$.

Because of the compactness of the sets $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{W}$, and given the choice of covariance and mean functions that verify Assumption 4.3, the estimated models $\hat{f}_1(\cdot)$ and $\hat{f}_2(\cdot)$ are such that (4.13) holds true. The found component-wise Lipschitz constants are shown in Table 18, and are used to compute the set sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$, and consequently find the maximum prediction horizon $N_p \in \mathbb{N}$ such that $\mathcal{X} \ominus \mathcal{R}(N_p)$ is non-empty and contains the reference equilibrium in its interior. We get that $N_p = 7$ steps, which corresponds to 3.5 minutes.

As the computed sets $\mathcal{F}(j)$ and $\mathcal{R}(j)$ are box-shaped, i.e.

$$\mathcal{F}(j) = \{x \in \mathbb{R}^2 : \ |x_i| \leq \bar{x}_{i,\mathcal{F}(j)}, \forall i \in \{1, 2\}\}$$

and

$$\mathcal{R}(j) = \{x \in \mathbb{R}^2 : \ |x_i| \leq \bar{x}_{i,\mathcal{R}(j)}, \forall i \in \{1,2\}\},$$

we provide the computed $\bar{x}_{i,\mathcal{F}(j)}$ and $\bar{x}_{i,\mathcal{R}(j)}$ in Table 19.

$L_{x,11}$	$L_{x,12}$	$L_{u,11}$	$L_{x,21}$	$L_{x,22}$	$L_{u,21}$
0.970	0.210	0.008	0.130	0.915	0.230

Table 18: Component-wise Lipschitz constants of the estimated GPs.

The sets $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{W}$ are also used to compute an estimate of the needed robust controlled invariant set Ω, whose knowledge is necessary to find the maximum value of the contraction factor γ that verifies all the conditions of Theorem 4.2. In detail, the graph-based method proposed in [34] is used to compute an outer approximation of Ω, which turns out to be identical to $\mathcal{X}$, i.e. $\Omega \equiv \mathcal{X}$.

The chosen candidate contractive function is $\Gamma(x) \coloneqq \|x - x_s\|^2$, where x_s is composed of the normalized values of the concentration and temperature references. The maximum value of $\Gamma(x)$ over $\mathcal{X}$ is $\Gamma_{\max} = 0.7336$, while the found value for ω is $\omega = 0.1418$. Consequently, in order for all the conditions of Theorem 4.2 to hold true, we check that the found contraction factor is such that $\gamma \leq \omega/\Gamma_{\max} = 0.1919$. We follow the method proposed in Section 3.7.1 to find the real contraction factor, which is found to be $\gamma = 0.1403 < 0.1919$.

Given that the chosen stage cost function is $\ell(x,u) \coloneqq (x - x_s)^\top Q(x - x_s) + (u - u_s)^\top R(u - u_s)$, with $Q = 10 \cdot I_2$ and $R = 1$, the maximum stage cost value is $\bar{\ell} = 10.3548$. To be consistent with the conditions of Theorem 4.2, the parameter ξ is set to $\xi = 1686$. The internal controller state θ is set to the initial condition $\theta(0) = \nu\Gamma(x(0)) = 0.56$, while the parameters ν and ϵ are respectively set to $\nu = 0.99$ and $\epsilon = 10^{-8}$.

The two optimization problems are solved using the interior-point algorithm of the `fmincon` function provided by MATLAB, and all the simulations are run on an Intel Core™ i7-10610U @ 1.80GHz 16GB RAM machine.

j	$\mathcal{F}(j)$		$\mathcal{R}(j)$	
	$\bar{x}_{1,\mathcal{F}(j)}$	$\bar{x}_{2,\mathcal{F}(j)}$	$\bar{x}_{1,\mathcal{R}(j)}$	$\bar{x}_{2,\mathcal{R}(j)}$
0	0.0214	0.0214	0	0
1	0.0257	0.0220	0.0214	0.0214
2	0.0300	0.0232	0.0470	0.0434
3	0.0345	0.0247	0.0770	0.0665
4	0.0392	0.0267	0.1115	0.0913
5	0.0443	0.0292	0.1507	0.1180
6	0.0499	0.0320	0.1951	0.1472
7	0.0559	0.0353	0.2449	0.1792

Table 19: Description of the computed sequences $\{\mathcal{F}(j)\}_{j\geq 0}$ and $\{\mathcal{R}(j)\}_{j\geq 0}$.

4.7.3 Results

The state and input trajectories obtained with the implemented GP-MPC are compared to those obtained by means of a standard discrete-time MPC based on a noise-less version of the model (4.29), which is assumed to have access to the entire state of the system, with quadratic stage cost function, terminal equality constraint and prediction horizon $N_p = 15$. The obtained state and input trajectories are respectively shown in Figures 16 and 17, which highlight how the mean trajectories obtained by means of the contractive GP-MPC and a standard MPC are very close, the main difference residing in a greater aggressiveness of the GP-MPC when the system state is far from the reference.

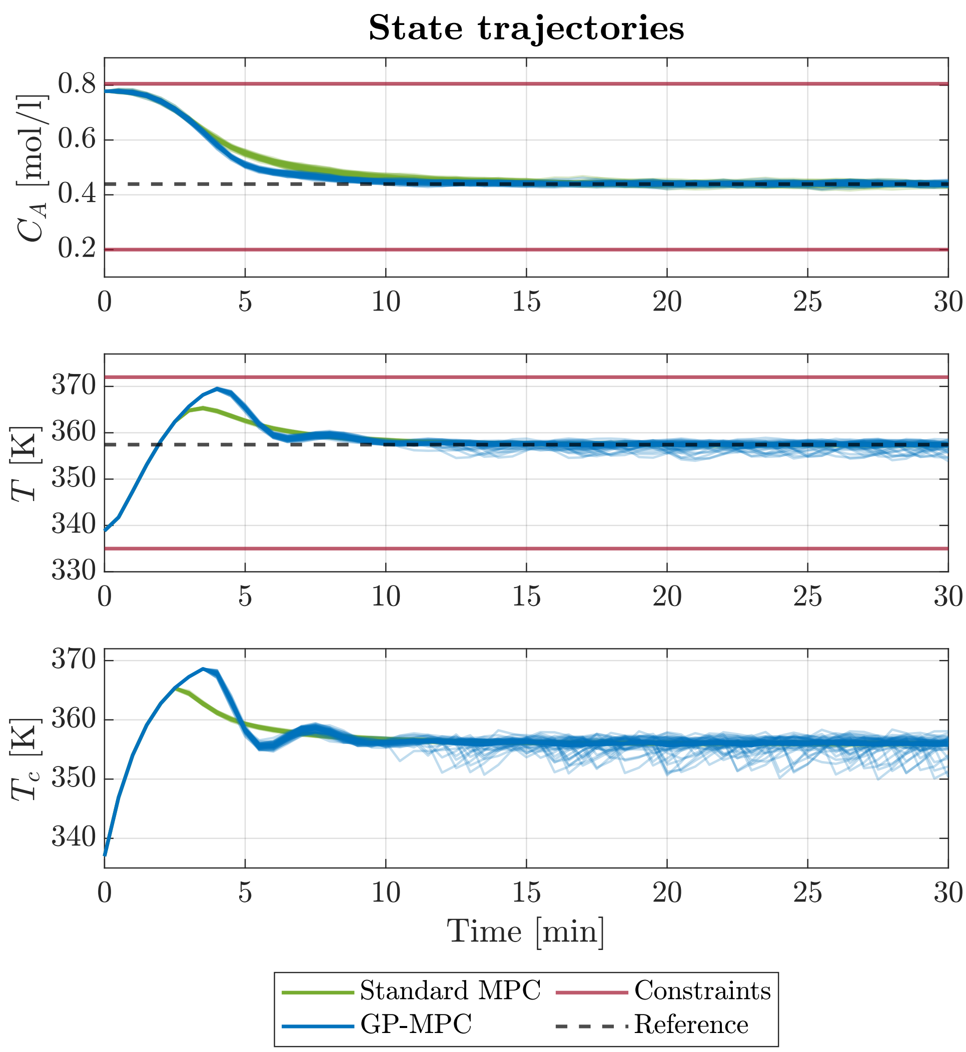


Figure 16: State trajectories obtained by means of the contractive GP-MPC and a standard MPC. The bold lines represent the mean of the trajectories obtained with each controller. The GP-MPC results to be a little more aggressive than a standard MPC. Nonetheless, it effectively stabilizes the closed-loop state around the reference.

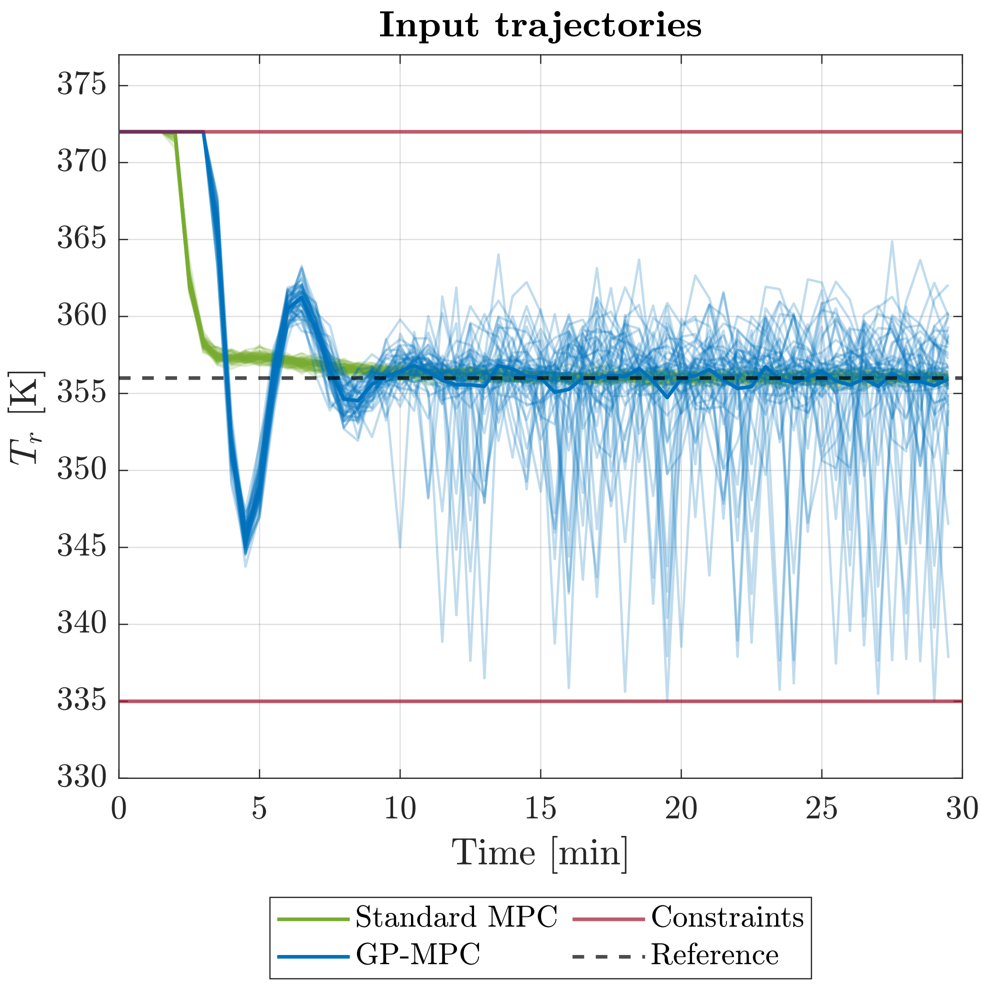


Figure 17: Input trajectories obtained by means of the contractive GP-MPC and a standard MPC. The bold lines represent the mean of the trajectories obtained with each controller. While the input trajectory provided by the standard MPC is more regular, the input trajectory provided by the GP-MPC gets quite noisy when the closed-loop system state is close to the reference. However, the constraints are always verified.

Remark 4.10

The standard discrete-time MPC that is implemented to provide a benchmark for the GP-MPC includes also a terminal equality constraint. The prediction horizon is chosen as $N = 15$ to avoid possible infeasibility issues.

The closed-loop performances provided by the two controllers are also compared by computing the infinite-horizon costs $\sum_{k=0}^{\infty} \ell(x(k) - x_s, u(k) - u_s)$ obtained for each simulation, under both the GP-MPC and the standard MPC. The stage cost function that is used to compute such infinite-horizon cost is the same employed in the GP-MPC, therefore considering only the state variables C_A and T, and normalizing all the state and input variables to stay in the range $[0, 1]$. The minimum, maximum, and mean costs obtained by employing the GP-MPC and a standard MPC are shown in Table 20, and prove how the performances given by the GP-MPC are comparable to those provided by a standard MPC, though showing a larger variability.

	Min. cost	Max. cost	Mean cost
GP-MPC	30.6723	33.4139	31.9987
MPC	28.8202	31.5737	30.3968

Table 20: Performances of the closed-loop system under the GP-MPC and a standard MPC.

Given that the contractive GP-MPC also optimizes the prediction horizon, it is interesting to analyze how this evolves over time and how its choice affects the time spent for the solution of the two optimization problems. Both the execution times and the optimal prediction horizons are shown in Figure 18. As expected, when the system state is far from the reference, the contractive GP-MPC often chooses longer prediction horizons, resulting also in longer execution times; when the system state is close to the reference, the chosen prediction horizon varies consistently, though showing a smaller mean value and therefore leading to execution times that are smaller on average. During all the simulations, the total execution time, i.e. the sum of the time spent for the solution of both the involved optimization problems, is smaller than the chosen sampling time.

4.8 Conclusion

In this chapter, we presented a tube-based contractive MPC for the regulation of nonlinear systems that are modeled by GPs. In detail, the aim was to show under what conditions a nonlinear system modeled by GPs can be controlled by the robust contraction-based MPC presented in Chapter 3, therefore ensuring the same recursive feasibility and stabilizing properties. The resulting GP-MPC was tested on a CSTR modeled by GPs and was found to be effective in practice, providing control performances similar to those of a standard nonlinear MPC. Although our controller proved effective and the assumptions on which it is based are reasonable from an engineering point of view, it would be interesting to investigate how to switch to a stochastic framework, allowing the information given by the posterior distributions to be exploited.

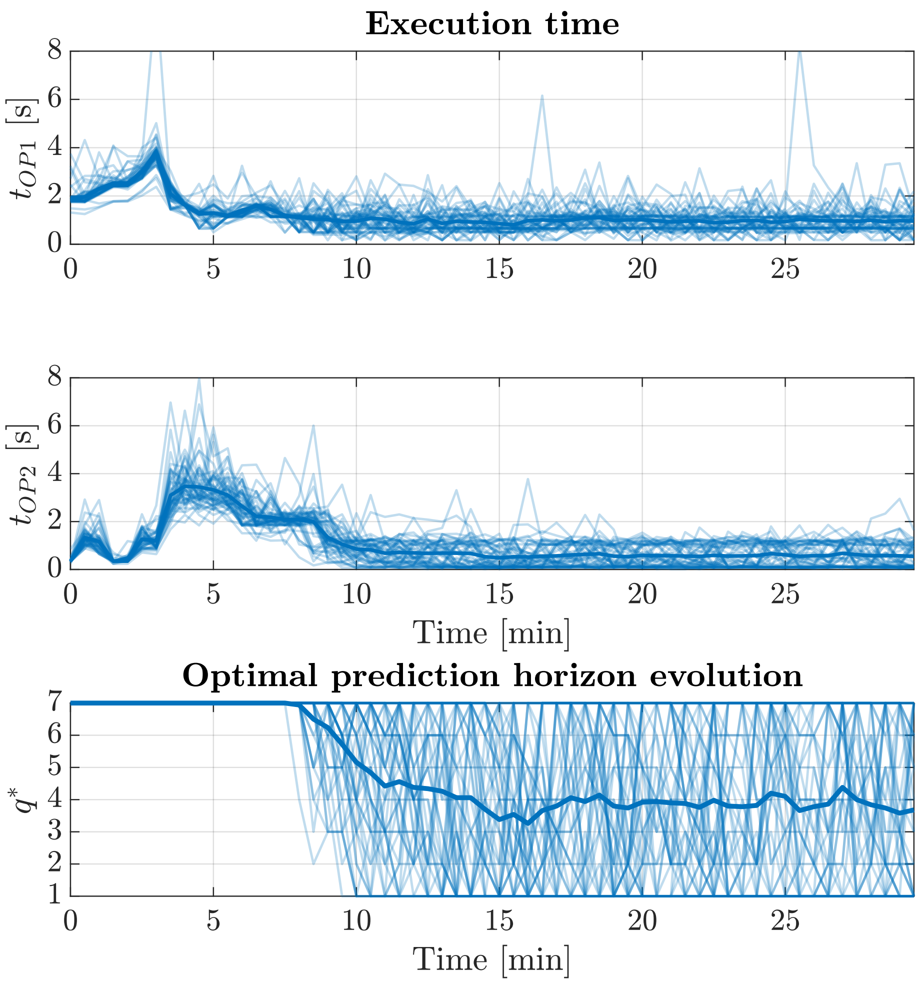


Figure 18: Execution times of the first optimization problem t_{OP1} and the second optimization problem t_{OP2}, and optimal prediction horizons q^* found by the contractive GP-MPC. The bold lines represent the mean execution times of the two optimization problems and the mean of the optimal prediction horizon. While the chosen prediction horizon is large when the closed-loop system state is far from the reference, it gets more unpredictable as the state approaches the reference.

Part II

Applicative contributions

Chapter 5. Artificial Pancreas under a zone MPC based on GPs

5.1 Introduction

Chapters 2, 3 and 4 presented the theoretical contributions. In particular, we dealt with three issues that are common in several contexts in which the design of an MPC is sensible:

- ensuring the robustness of an MPC for tracking designed for the control of nonlinear systems;
- simplifying the design of the terminal ingredients of a robust MPC for the regulation of nonlinear systems;
- using machine learning methods, like GPs, to model the system under control, still ensuring robustness and input-to-state stability, again employing simplified stability-related terminal ingredients.

The objectives we set ourselves during the design of the methods presented in Chapters 2, 3 and 4 were to expand the set of systems that can be controlled with an MPC and to provide formulations that could be easier to implement, in some situations, compared to other MPC schemes that were already available. Therefore, in line with what is often done in purely theoretical research, we strove to make our methods as generalizable as possible.

In this chapter, we present an applicative contribution whose results were published in [113]. Unlike the works described in the previous chapters, the result presented in the next sections does not bring any particular theoretical contributions. The work published in [113] belongs to a family of works that the Control Systems & Automation Lab of the University of Bergamo has carried out since 2022, including those in [133, 134], that aim at finding new MPC-based solutions for the automatic regulation of the blood glucose (BG) level in Type 1 Diabetes Mellitus (T1DM) subjects. In particular, [113] presents a zone MPC based on GPs. Therefore, in the remainder of this chapter we first introduce the problem of regulating the BG level in T1DM subjects by means of automatic insulin delivery (AID) systems, then we return to GPs, as the way they are employed in this chapter is different than in Chapter 4, and discuss the MPC that we designed. At the end, the method is validated in simulation.

5.2 On the usage of GPs and MPC for automatic insulin delivery in T1DM subjects

The BG concentration in healthy individuals is between 70 and 100[mg/dl] on an empty stomach. When this concentration changes, the pancreas reacts to bring it back to its normal value. In particular, when this level increases, the beta cells of the pancreas secrete insulin, while when this level decreases, the alpha cells of the pancreas produce glucagon. This does not occur in diabetic subjects. Indeed, T1DM is a chronic disease characterized by the irreversible destruction of the beta cells contained in the islets of Langerhans of the pancreas [64]. Given that insulin promotes the transportation of glucose to all cells, the absence of beta cells in T1DM subjects results in less efficient transportation of glucose to the cells and the accumulation of glucose in the blood stream, a condition that takes the name of hyperglycemia that can also harm some organs and tissues in the long term.

Another concern related to T1DM is given by its increasing incidence and costs on healthcare. Indeed, in 2021 there were about 8.4 million individuals with T1DM worldwide, but these are predicted to increase to 13.5-17.4 million by 2040, as reported in [53], with the largest increase in low-income and lower-middle-income countries. In Italy the incidence increases by 3.6% each year [49]. Given that T1DM patients have remarkable mean total medical costs ($18,817 per patient per year in the US [61]), this predicted increase needs to be carefully faced by governments, as it can also threaten the sustainability of healthcare systems, that nowadays are already facing important costs related to diabetes (considering also Type 2 Diabetes Mellitus, which is a different disease). For example, the direct costs of diabetes in Italy amounted to €7.94 billion in 2010 [63]. For these reasons, it is important to study new treatments that can both effectively regulate the BG level in diabetic patients and reduce the costs of diabetes on healthcare systems.

The treatment of T1DM typically consists of daily injections of slow-acting insulin, which aims at avoiding hyperglycemia events. However, the effectiveness of such a therapy highly depends on the precision of the subjects in planning mealtimes, carbohydrate intakes, and sessions of physical exercise; these, indeed, greatly influence BG levels. The last statement makes it easy to understand what the second objective of any diabetes therapy is, namely to make the life of a diabetic as similar as possible to that of a healthy person. To do so, it is necessary to reduce the effect of the variables listed before, which is made possible by exploiting closed-loop control techniques.

Every closed-loop technique needs a sensor to measure the current value of the variable under control, which might be influenced by external disturbances, a control algorithm to compute the right control action that has to be provided to the system based on the value of the variable under control and a reference, and an actuator that concretely provides the computed control action to the system. The Artificial Pancreas (AP) is a modern AID system that is used to treat T1DM and that is made of components that cover all the features of a closed-loop system listed before, namely: (i) a continuous glucose monitor (CGM), which is a tiny and non-invasive sensor that checks glucose levels through a needle inserted under the skin every few minutes, (ii) a control algorithm that computes the right amount of insulin to be injected based on the last CGM readings, and (iii) an insulin infusion pump [107]. APs usually handle two different types of insulin, which are: (i) basal insulin, which is injected throughout the entire day to keep the BG level between 70 and 180mg/dl, namely the euglycemic range, between the meals, and (ii) insulin boluses, which are provided only at mealtimes to avoid or limit the hyperglycemic peaks due to carbohydrate intakes.

The birth of APs has also encouraged researchers to develop algorithms for BG control in diabetic subjects. Several controllers have been proposed and tested on real and in-silico subjects, starting from classic controllers like PID, as in [59, 127], and moving on to more complex solutions like MPC, e.g. [2, 19, 38, 50, 51, 52, 56, 58, 72, 130, 135, 140]. Nevertheless, MPC requires a model of the system under control to make predictions, which makes the usage of MPC in APs challenging. On one hand, MPC can handle constraints and both predictable and unpredictable perturbations, making it a potentially good choice for the control of BG in T1DM subjects; on the other hand, the behavior of the endocrine-metabolic system considerably changes between different subjects and also within the same person over time, making it noticeably difficult to model.

For this reason, even though MPC algorithms based on first-principle models of the endocrine-metabolic system have been proposed, control system researchers have been increasingly opting for black-box models. Particular interest has been posed in developing MPC schemes based on NNs, as in [12, 42]. Other works focused on the usage of other black-box models, such as models that are based on the Lipschitz continuity of the process under control, e.g. [133, 134], or kernel-based regression, like [106]. Among the kernel-based solutions, GPs also represent an interesting alternative to first-principle models, for the same reasons explained in Chapter 4. GPs have been mainly exploited inside sliding mode controllers, as in [112], reinforcement learning algorithms, see [33], and to detect changes in the insulin sensitivity (IS) of a person, e.g. [109].

5.3 GPR for uncertain inputs

In Section 4.2 we discussed the main concepts about GPs and how they can be used to model dynamic systems. However, despite the probabilistic nature of GPs, the reader may have noticed that such probabilistic nature was not explicitly taken into account in the design of the MPC presented in

Chapter 4. In fact, instead of relying on the variance of the posterior Gaussian distributions obtained via GPR to tighten the state constraints of the defined optimal control problem, we relied on the Lipschitz continuity of the latent function and the chosen mean and covariance functions to bound the prediction error and the effect of uncertainty propagation.

On the contrary, the MPC presented in this chapter makes use of the posterior variances to tighten the state constraints, thus taking into account the probabilistic nature of GPs. However, using a GP to make open-loop predictions within a certain horizon is challenging. Let us provide an example to explain the issues related to making multi-step-ahead predictions. Suppose we employ a GP to model a state function $f(x, u)$, where $x \in \mathbb{R}$ for simplicity, and suppose we use a GP-MPC to control the system. In particular, let us analyze what happens at time k when the GP-MPC has to predict a state trajectory $\hat{\mathbf{x}}(k) = \{\hat{x}(0|k), \ldots, \hat{x}(N_p|k)\}$, based on the known initial condition $x(k)$ and a candidate control sequence $\mathbf{u}(k) = \{u(0|k), \ldots, u(N_p - 1|k)\}$, taking into account the PDF of the made predictions:

1. The first state of the trajectory is imposed to be equal to the initial condition, i.e. $\hat{x}(0|k) = x(k)$.

2. The one-step-ahead prediction, $\hat{x}(1|k)$, is inferred by GPR based on the regressor $z(k) = (x(k), u(k))$, which is deterministic. Indeed, the initial condition $x(k)$ is known and the candidate control action $u(0|k)$ has no probability distribution. In detail, we get that $\hat{x}(1|k) \sim \mathcal{N}\left(\mu(z(k)|\mathcal{D}), \sigma^2(z(k)|\mathcal{D})\right)$, where $\mu(\cdot|\mathcal{D})$ and $\sigma^2(\cdot|\mathcal{D})$ are defined in (4.4a) and (4.4b).

3. Following the same rationale of the previous point, the predictions $\hat{x}(j|k)$, with $j \in [2, N_p]$, should be computed using GPR on the regressor $z(j-1|k) = (\hat{x}(j-1|k), u(j-1|k))$. However, the regressor $z(j-1|k)$ is stochastic. Therefore, it is necessary to understand how to take into account the probabilistic nature of the regressor in the inference step.

The issue highlighted at point 2 is very common when GPs are used to simulate dynamic systems or time series, and has different solutions, as specified in [119]. Formalizing this issue mathematically, considering the case in which the test regressor z_* is Gaussian distributed, i.e. $z_* \sim \mathcal{N}(\mu_{z_*}, \Sigma_{z_*})$.

Remark 5.1

In the context of GP-MPC schemes, where GPs are employed to make state predictions within a finite horizon with $N_p > 1$, it is actually rare to have Gaussian distributed regressors. In fact, apart from the one-step-ahead state prediction, which is surely Gaussian due to the fact that it results from a standard GPR step carried out on a deterministic regressor, all the following *state predictions are non-Gaussian. Indeed, the true posterior PDF is*

$$p\left(f(z_*)|\mu_{z_*},\Sigma_{z_*},Z,\mathsf{y},\zeta,\sigma_\eta\right)=\int p\left(f(z_*)|z_*,Z,\mathsf{y},\zeta,\sigma_\eta\right)p\left(z_*|\mu_{z_*},\Sigma_{z_*}\right)dz_*, \tag{5.1}$$

where $p\left(z_*|\mu_{z_*},\Sigma_{z_*}\right)$ *is known to be Gaussian, but* $p\left(f(z_*)|z_*,Z,\mathsf{y},\zeta,\sigma_\eta\right)$*, defined in* (4.3)*, is clearly a nonlinear function. Therefore, knowing that mapping a Gaussian distribution through a nonlinear function results in a non-Gaussian distribution,* (5.1) *is non-Gaussian and is not even guaranteed to be unimodal.*

Nonetheless, it will be clarified later in the text why considering a regressor $z_*\sim\mathcal{N}(\mu_{z_*},\Sigma_{z_*})$ *is convenient.*

Given that the integral (5.1) is analytically intractable, approximate ways need to be found for computing $p\left(f(z_*)|\mu_{z_*},\Sigma_{z_*},Z,\mathsf{y},\zeta,\sigma_\eta\right)$. In [119], several solutions are considered. Hereafter, the three most meaningful solutions are described.

The first solution, that is not suggested, is to simply ignore the stochasticity of the regressor z_*, therefore carrying out GPR on its mean vector μ_{z_*} and dropping the information provided by Σ_{z_*}. This, of course, results in Gaussian posterior distributions with underestimated variance. The second solution consists in numerically approximating the integral (5.1) by a Monte-Carlo approach, namely taking n_s independent samples $z_{*(j)}$, with $j\in[1,n_s]$, from $p\left(z_*|\mu_{z_*},\Sigma_{z_*}\right)$ and computing

$$\begin{aligned}p\left(f(z_*)|\mu_{z_*},\Sigma_{z_*},Z,\mathsf{y},\zeta,\sigma_\eta\right)&=\int p\left(f(z_*)|z_*,Z,\mathsf{y},\zeta,\sigma_\eta\right)p\left(z_*|\mu_{z_*},\Sigma_{z_*}\right)dz_*\\&\approx\frac{1}{n_s}\sum_{j=1}^{n_s}p\left(f(z_*)|z_{*(j)},Z,\mathsf{y},\zeta,\sigma_\eta\right).\end{aligned}$$

The third solution, which is computationally more efficient than a Monte-Carlo approximation, is limited to GPs that employ a covariance function that belongs to the family of Gaussian covariance functions, e.g. the previously presented and widely used SE-ARD function. Indeed, with such covariance functions it is possible to exactly compute the first and the second moments of the distribution (5.1). Therefore, ignoring the real shape of such posterior distribution, one can approximate it to a Gaussian distribution whose first and second moments match the real ones. In this way, information is lost about the shape of the posterior distribution, but at least the computed variances tend to be larger, therefore modeling the propagation of uncertainty. Next, we provide the new posterior mean and variance functions $\mu(\cdot|\mu_{z_*},\Sigma_{z_*},\mathcal{D})$ and $\sigma^2(\cdot|\mu_{z_*},\Sigma_{z_*},\mathcal{D})$ to be used with Gaussian regressors $z_*\sim\mathcal{N}(\mu_{z_*},\Sigma_{z_*})$ and considering a SE-ARD covariance function:

$$\mu(\cdot|\mu_{z_*},\Sigma_{z_*},\mathcal{D})=\upsilon^\top\psi, \tag{5.2a}$$

$$\sigma^2(\cdot|\mu_{z*}, \Sigma_{z*}, \mathcal{D}) = \sigma_f^2 - \mathrm{tr}\left((K + \sigma_\eta^2 I_{n_\mathcal{D}})^{-1}\tilde{Q}\right) + \upsilon^\top \tilde{Q}\upsilon - \upsilon^\top \psi, \tag{5.2b}$$

where:

- $\upsilon \coloneqq (K + \sigma_\eta^2 I_{n_\mathcal{D}})^{-1}\mathsf{y}$;
- $\psi \in \mathbb{R}^{n_\mathcal{D}}$ is composed of elements ψ_j, for $j \in [1, n_\mathcal{D}]$ defined as

$$\begin{aligned}\psi_j \coloneqq\ & \sigma_f^2 \det\left(\Sigma_{z*}\Lambda^{-1} + I_{n_z}\right)^{-1/2} \\ & \cdot \exp\left[-\frac{1}{2}(z_{(j)} - \mu_{z*})^\top(\Sigma_{z*} + \Lambda)^{-1}(z_{(j)} - \mu_{z*})\right];\end{aligned}$$

- the elements of $\tilde{Q} \in \mathbb{R}^{n_\mathcal{D}\times n_\mathcal{D}}$ are defined as

$$\begin{aligned}\tilde{Q}_{i,j} \coloneqq\ & \frac{k(z_{(i)}, \mu_{z*})k(z_{(j)}, \mu_{z*})}{\det(2\Sigma_{z*}\Lambda^{-1} + I_{n_z})^{1/2}} \\ & \cdot \exp\left[(\tilde{z}_{(ij)} - \mu_{z*})^\top\left(\Sigma_{z*} + \frac{\Lambda}{2}\right)^{-1}\Sigma_{z*}\Lambda^{-1}(\tilde{z}_{(ij)} - \mu_{z*})\right],\end{aligned}$$

  where $\tilde{z}_{(ij)} \coloneqq \frac{1}{2}(z_{(i)} + z_{(j)})$.

For further details on how the last equations are obtained, we refer the reader to [35].

5.4 Problem formulation

The endocrine-metabolic system is a highly complex nonlinear time-varying system. Therefore, in order to simplify its modeling, we assume the endocrine-metabolic system of T1DM subjects, treated with an AP providing basal insulin and insulin boluses, to be modeled by a nonlinear discrete-time multiple-input-single-output system. The output of this system is the subcutaneous glucose level, which (indirectly) represents the BG level and is measured by a CGM, and is considered to behave as a Nonlinear AutoRegressive eXogenous model (NARX) of the form

$$\begin{aligned}y(k+1) = f(&y(k), \dots, y(k - n_a + 1), \\ &u_{\text{meal}}(k), \dots, u_{\text{meal}}(k - n_b + 1), \\ &u_{\text{bolus}}(k), \dots, u_{\text{bolus}}(k - n_c + 1), \\ &u_{\text{basal}}(k), \dots, u_{\text{basal}}(k - n_d + 1)) + \eta(k),\end{aligned} \tag{5.3}$$

where k denotes the discrete-time index, $y(k) \in \mathbb{R}$ is corrupted by i.i.d. white Gaussian noise $\eta(k) \sim \mathcal{N}(0, \sigma_\eta^2)$, and n_a, n_b, n_c, n_d are the memory horizons of glucose, meals, insulin boluses and basal insulin that are considered to build the NARX state.

Remark 5.2

CGM sensors measure the BG level only indirectly. Indeed, they measure the concentration of glucose in the subcutaneous tissues, which can be considered as a noisy and delayed measure of the real BG level, as outlined in [129].

The function $f(\cdot)$ is considered to be unknown and is to be modeled by means of a GP. Therefore, supposing a dataset $\mathcal{D}$ is available composed of $n_{\mathcal{D}}$ couples $(y(k), z(k))$, where the regressor $z(k)$ is defined as

$$\begin{aligned} z(k) \coloneqq \Big(& y(k-1), \dots, y(k-n_a), \\ & u_{\text{meal}}(k-1), \dots, u_{\text{meal}}(k-n_b), \\ & u_{\text{bolus}}(k-1), \dots, u_{\text{bolus}}(k-n_c), \\ & u_{\text{basal}}(k-1), \dots, u_{\text{basal}}(k-n_d) \Big) \end{aligned} \tag{5.4}$$

and allows us to redefine (5.3) as

$$y(k+1) = f(z(k)) + \eta(k). \tag{5.5}$$

While u_{bolus} and u_{basal} are controlled inputs, u_{meal} is an uncontrolled input. Moreover, meals are considered to be announced in a timely manner and without any time advance. The controlled inputs are subject to constraints that depend on the infusion pump capabilities and the subject's IS:

$$(u_{\text{bolus}}(k), u_{\text{basal}}(k)) \in \mathcal{U}, \quad \forall k \geq 0. \tag{5.6}$$

The real control objective is keeping the BG level inside the euglycemic range. However, since the real BG level is considered to be inaccessible, the final control goal becomes that of keeping the measured subcutaneous glucose level y to the mentioned euglycemic range, namely:

$$70\text{mg/dl} \leq y(k) \leq 180\text{mg/dl}, \quad \forall k \geq 0. \tag{5.7}$$

5.5 Model design

As stated in the previous section, historical data about the considered T1DM subject are assumed to be available and to be exploited for the training of a GP, i.e. to determine its hyperparameters. However, before determining such hyperparameters, it is necessary to choose the mean function $m(\cdot)$ and the covariance function $k(\cdot,\cdot)$ that characterize the GP and the memory horizons n_a, n_b, n_c and n_d.
Supposing the latent function $f(\cdot)$ to be smooth, $m(z) = 0$ is set and a modified version of the SE-ARD covariance function is used, namely:

$$k(z, \check{z}) = \exp\left[-\frac{1}{2}(z-\check{z})^\top \Lambda^{-1} (z-\check{z})\right], \tag{5.8}$$

where $\Lambda = \mathrm{diag}(l_1^2, \ldots, l_{n_a+n_b+n_c+n_d}^2)$ and $\zeta = (l_1, \ldots, l_{n_a+n_b+n_c+n_d})$ are the covariance hyperparameters. From a practical point of view, (5.8) is equivalent to (4.1), but the hyperparameter σ_f is imposed to be $\sigma_f = 1$, therefore reducing the number of hyperparameters to be estimated.

Moreover, assuming a reduction in the complexity of both the training and inference phases by employing a SoD approximation of GPs. Therefore, another parameter that has to be chosen is the number n_r of data points to be kept. The memory horizons n_a, n_b, n_c and n_d and the number n_r of observations to be kept have to be chosen for each subject.

While a straightforward way of finding the optimal value of these parameters might be to train different models on a training set and choose as the final model the one providing the maximum marginal likelihood over a validation set, this method might bring unsatisfactory results. This is due to the fact that a GP can fit the data well even if it misunderstands the effect that each regressor component has on the output function. In particular, since insulin boluses are often provided at mealtimes, many models providing good marginal likelihood values actually interpret boluses as inputs that cause an increase in blood glucose and meals as inputs causing a decrease.

For this reason, provided that the parameters to be found belong to the set of natural numbers and can take only a finite number of values, a better sequence of actions that can be accomplished for model selection is the following:

1. For each combination of n_a, n_b, n_c, n_d and n_r:

    a) A temporary dataset $\mathcal{D}_{\text{temp}}$ is built considering the regressors

    $$\begin{aligned} z(k) := (&y(k-1), \ldots, y(k-n_a), \\ &u_{\text{meal}}(k-1), \ldots, u_{\text{meal}}(k-n_b), \\ &u_{\text{bolus}}(k-1), \ldots, u_{\text{bolus}}(k-n_c), \\ &u_{\text{basal}}(k-1), \ldots, u_{\text{basal}}(k-n_d)), \end{aligned} \tag{5.9}$$

    where n_a, n_b, n_c, n_d are given by the currently addressed combination.

    b) To keep the same number of $(y(k), z(k))$ pairs for each combination, and to reduce the computational complexity of both the training and inference phases, a subset of $\mathcal{D}_{\text{temp}}$ is built by selecting the n_r most informative observations. This selection is done by running a k-means clustering algorithm with k set to n_r and then keeping the observations that are closest to the cluster centroids.

    c) The hyperparameters of the GP are found by maximization of the log marginal likelihood.

2. The 10 models providing the best marginal likelihood values over a validation set are manually evaluated in order to choose the most consistent with the known effects of the system inputs. In particular, a good model is expected to predict a decrease in the glucose level when the value of the provided basal insulin is large and to predict an increase after meals, which should be smaller when also an insulin bolus is provided at mealtime.

Remark 5.3
The aforementioned model selection process is cumbersome and theoretically less elegant than desired. A fully automated, reliable model selection procedure would be preferable.

5.5.1 The optimization problem

The goal of the here-presented controller is to keep the measured subcutaneous glucose level of T1DM subjects within the euglycemic range (5.7), while trying to minimize the amount of injected insulin. At each sampling instant k, the controller has to compute the best $u_{\text{basal}}(k)$ and $u_{\text{bolus}}(k)$ based on $y(k)$, a collection of past input and output values $z(k)$, where $z(k)$ has the same structure introduced in (5.9), and $u_{\text{meal}}(k)$.

The predictions of the blood glucose levels along the prediction horizon are carried out via GPR. At each prediction step j, a regressor is built such that

$$\begin{aligned} z(j|k) := (&\hat{y}(j|k), \ldots, \hat{y}(j-n_a+1|k),\\ &u_{\text{meal}}(j|k), \ldots, u_{\text{meal}}(j-n_b+1|k),\\ &u_{\text{bolus}}(j|k), \ldots, u_{\text{bolus}}(j-n_c+1|k),\\ &u_{\text{basal}}(j|k), \ldots, u_{\text{basal}}(j-n_d+1|k)), \end{aligned}$$

where:

- $\hat{y}(j-a|k) = y(k+j-a)$ for $j \leq a$;
- $\hat{y}(j-a|k) \sim \mathcal{N}(\mu_{y(j-a|k)}, \sigma^2_{y(j-a|k)})$, where

$$\mu_{y(j-a|k)} = \mu(z(j-a-1|k)|\mu_{z(j-a-1|k)}, \Sigma_{z(j-a-1|k)}, \mathcal{D})$$

  and

$$\sigma^2_{y(j-a|k)}) = \sigma^2(\mu(z(j-a-1|k)|\mu_{z(j-a-1|k)}, \Sigma_{z(j-a-1|k)}, \mathcal{D}).$$

The regressor $z(j|k)$ is then used to get the prediction $\hat{y}(j+1|k)$, whose mean and variance are computed as explained in Section 5.3.

Given that hyperglycemia and hypoglycemia events, although undesirable, are possible and that the latter are more critical than the former, the cost function and the constraints involved in the MPC are inspired by [2]. In particular, soft constraints are provided to keep the predicted subcutaneous glucose levels within the euglycemic range if possible, and the related slack variables are weighted inside the cost function in such a way that makes hypoglycemia events less likely than hyperglycemia ones, because they are more dangerous.

The soft constraints on the predicted subcutaneous glucose levels are made even safer by considering also the standard deviation of each prediction. This comes from the fact that both predictions with expected values that lie outside the euglycemic range and predictions with large variances can be considered unsafe.

Moreover, the MPC belongs to the family of zone MPCs, as it brings the system output to a specified range, or zone, instead of bringing it to a reference point. This is done by introducing a new optimization variable y_a, which is imposed to stay within a desired range, and by penalizing the distance between the predicted glucose levels and this variable in the cost function.

Along the chosen prediction horizon N_p, basal insulin is free to change at every prediction step. The same does not apply to insulin boluses; a bolus can be provided only at the first prediction step and only when $u_{\text{meal}} > 0$.

The last concepts and assumptions formally translate into the following optimization problem $P_{N_p}(y(k), z(k))$, which has to be solved at each sampling instant k:

$$\begin{aligned}\min_{(\mathbf{u}(k),\delta_{\text{hypo}}(k),\delta_{\text{hyper}}(k),y_a(k))} \sum_{j=0}^{N_p} &\Big[Q(\hat{y}(j|k) - y_a(k))^2 + \tau_{\text{hypo}}\delta_{\text{hypo}}(j|k)^2 + \tau_{\text{hyper}}\delta_{\text{hyper}}(j|k)^2\Big] \\ &+ \sum_{j=0}^{N_p-1} \Big[R_{\text{basal}}u_{\text{basal}}(j|k)^2\Big] + R_{\text{bolus}}u_{\text{bolus}}(0|k)^2\end{aligned} \tag{5.10a}$$

$$\text{s.t. } \hat{y}(0|k) = y(k) \tag{5.10b}$$

$$\hat{y}(j|k) \sim \mathcal{N}(\mu_{y(j|k)}, \sigma^2_{y(j|k)}) \quad \forall j \in [1, N_p] \tag{5.10c}$$

$$0 \le u_{\text{basal}}(j|k) \le \overline{u}_{\text{basal}} \quad \forall j \in [0, N_p - 1] \tag{5.10d}$$

$$0 \le u_{\text{bolus}}(0|k) \le \overline{u}_{\text{bolus}}(u_{\text{meal}}(k)) \tag{5.10e}$$

$$\underline{y} \le y_a(k) \le (\underline{y} + \overline{y})/2 \tag{5.10f}$$

$$\delta_{\text{hypo}}(j|k) \ge 0,\ \delta_{\text{hyper}}(j|k) \ge 0\ \forall j \in [0, N_p] \tag{5.10g}$$

$$\hat{y}(j|k) - n_\sigma\sigma_{y(j|k)} \ge \underline{y} - \delta_{\text{hypo}}(j|k)\ \forall j \in [0, N_p] \tag{5.10h}$$

$$\hat{y}(j|k) + n_\sigma\sigma_{y(j|k)} \le \overline{y} + \delta_{\text{hyper}}(j|k)\ \forall j \in [0, N_p] \tag{5.10i}$$

where:

- $\hat{y}(j|k)$ are Gaussian distributions obtained by means of multi-step-ahead prediction.
- $\overline{u}_{\text{basal}}$ is a design parameter and depends on the insulin sensitivity of the subject.
- The insulin bolus upper bound is computed as

$$\overline{u}_{\text{bolus}}(u_{\text{meal}}(k)) = \begin{cases} 0, & u_{\text{meal}}(k) = 0 \\ \overline{\overline{u}}_{\text{bolus}}, & u_{\text{meal}}(k) > 0 \end{cases}$$

  where $\overline{\overline{u}}_{\text{bolus}}$ is a design parameter.

- $\delta_{\text{hypo}}(j|k)$ and $\delta_{\text{hyper}}(j|k)$ are the slack variables related to hypoglycemia and hyperglycemia events.
- $\underline{y} = 70\text{mg/dl}$ and $\overline{y} = 180\text{mg/dl}$ are the lower and upper bounds of the euglycemic range. Note that y_a is imposed to stay within the lower half of the euglycemic range $[70, 125]\text{mg/dl}$ instead of the entire euglycemic range $[70, 180]\text{mg/dl}$ to recover more quickly from possible hyperglycemia events.
- n_σ is a design parameter. The higher n_σ, the more the MPC is expected to avoid uncertain predictions.

5.6 Case study

The designed zone MPC was tested on 11 in-silico adult subjects by means of the UVA/Padova T1DM simulator, whose details can be found in [100]. The simulations were run on a machine equipped with an Intel® Xeon® E3-1225 CPU and 8GB of RAM. The controller was implemented in MATLAB® and the optimization problem that is inherent in each MPC step was solved by means of the `fmincon` function.

Both the data collection phase and the final simulations were carried out considering a sampling time $T_s = 10\text{min}$. This choice is motivated by the need to guarantee that the execution time of a single MPC step is shorter than the sampling time. In order to create GPs that really understand the dynamics of the endocrine-metabolic system, it is necessary to create regressors that take into account output and input values of the past 1-2 hours; for this reason choosing a shorter sampling time would lead to the need of building regressors with larger memory horizons n_a, n_b, n_c and n_d, thus making the optimization problem involved in the here proposed MPC computationally demanding.

In both the data collection phase and the final simulations, the daily nutrition plan was the same for all the subjects and is described in Table 21.

Time	7a.m.	10a.m.	12a.m.	3p.m.	6p.m.	10p.m.
Intake [g]	30	10	45	10	60	10

Table 21: Daily carbohydrate consumption for the in-silico subjects.

5.6.1 Data collection and model identification

In order to identify high-quality GPs for each subject, it was necessary to collect large amounts of data about all the conditions a diabetic subject can run into. For this reason, data about each subject were collected over a 35 days simulation with the goal of leading the subject to hyperglycemia, euglycemia and hypoglycemia by randomly changing the basal insulin provided over the simulation time and giving random boluses at mealtime.

Giving random values of basal insulin and boluses requires knowing an upper bound for the two, namely the parameters $\overline{u}_{\text{basal}}$ and $\overline{\overline{u}}_{\text{bolus}}$ introduced in (5.10). In order to find these values it was necessary to run multiple simulations on each subject and to fix the upper bounds in order to avoid too dangerous hypoglycemia and hyperglycemia events. The bounds that were chosen are shown in Table 22. Once the dataset $\mathcal{D}$ was built, the process explained in Section 3.1 was followed to determine n_a, n_b, n_c, n_d and n_r and find the best GP for each subject. In particular, in order to limit the number of parameter combinations to be evaluated, the following assumptions were made:

$$n_a \in \{4, 6\}, \tag{5.11a}$$

$$n_b, n_c, n_d \in [6, 12], \tag{5.11b}$$

$$n_r \in \{150, 200, 250, 300, 350, 400\}. \tag{5.11c}$$

The parameters of the chosen models are shown in Table 23.

5.6.2 Controller settings

The personalization of the here-proposed controller to meet the needs of the single T1DM subject is not only guaranteed by the GP model that is fitted to the single subject's data, but also by the possibility

	Subject										
	Avg	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**
$\overline{u}_{\text{basal}}$	180	255	260	320	190	160	350	160	240	140	230
$\overline{\overline{u}}_{\text{bolus}}$	10000	15000	12000	15000	12000	9000	27000	3000	18000	8000	20000

Table 22: Performance metrics of the closed-loop system on the 11 in-silico subjects.

	Subject										
	Avg	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**
n_a	4	4	4	6	6	4	4	4	4	4	4
n_b	8	6	10	10	10	6	10	10	10	10	10
n_c	10	9	12	6	10	6	8	12	10	10	10
n_d	6	9	10	12	8	9	8	6	12	12	12
n_r	150	250	250	350	250	250	350	250	250	250	250

Table 23: Selected model for each subject.

Subject	N_p	Q	R_{basal}	R_{bolus}	τ_{hypo}	τ_{hyper}	n_σ
#1, #3-10	12	100	0.5	0.1	10^5	10^4	2
#2, Avg	12	100	0.1	0.1	10^5	10^4	2

Table 24: MPC settings.

to choose different settings for the zone MPC. In particular, the parameters that can be changed to optimize the performance of the controller on the single subject are N_p, Q, R_{basal}, R_{bolus}, τ_{hypo}, τ_{hyper} and n_σ.

The settings that were chosen for the in-silico subjects after several trials were similar. In particular, only R_{basal} took different values depending on the subject. The parameters for all the subjects are presented in Table 24.

5.6.3 Simulation results

The results of the simulations are promising; in particular, no subjects ran into severe hypoglycemia events and, apart from Adult 7, all the other subjects experienced a limited number of hyperglycemia events.

Before evaluating the performance of the proposed method, let us briefly discuss the metrics that can be used to carry out such evaluation. In detail, the ideal controller should keep the glucose level of the treated subject almost constant and close to 110mg/dl. Therefore, two metrics that can be used to evaluate the closed-loop performance are the glucose level mean and standard deviation, as they give insight about how close the glucose level was to the ideal level and the magnitude of its changes over time. Other important metrics are the time spent in hypoglycemia, i.e. below 70mg/dl, and the time spent in hyperglycemia, in particular above 180mg/dl. Two metrics are often used to measure the time spent in euglycemia, namely the time spent in the tighter range [70,140]mg/dl, and the time spent in the wider range [70,180]mg/dl. What changes between these two ranges is the upper bound, as hyperglycemia is less dangerous than hypoglycemia in the short term, thus making hyperglycemia events in the range [140,180]mg/dl not particularly relevant. The values of the described metrics for the in-silico subjects that were treated with the MPC are reported in Tables 25 and 26. Note how almost all the subjects spent between the 85% and the 100% of the simulation time in euglycemic range, the only exception being Adult 7. Furthermore, none of the subjects ran into hypoglycemia events.

Subject	Avg	#1	#2	#3	#4	#5
Mean glucose [mg/dl]	141.55	140.17	134.43	130.38	131.24	119.66
Glucose SD [mg/dl]	28.13	17.78	24.67	33.61	20.48	30.17
Time in [70,140]mg/dl [%]	57.76	55.45	71.49	71.42	66.30	76.28
Time in [70,180]mg/dl [%]	89.93	96.41	94.38	93.57	98.10	93.91
Time above 180mg/dl [%]	10.07	3.59	5.62	6.43	1.90	6.09
Time below 70mg/dl [%]	0.00	0.00	0.00	0.00	0.00	0.00

Table 25: Performance metrics of the closed-loop system on the 11 in-silico subjects.

Subject	#6	#7	#8	#9	#10
Mean glucose [mg/dl]	122.93	181.05	145.10	140.49	133.68
Glucose SD [mg/dl]	24.07	57.81	26.79	35.74	17.26
Time in [70,140]mg/dl [%]	80.84	33.28	56.79	59.52	63.13
Time in [70,180]mg/dl [%]	95.76	51.15	88.43	85.81	99.79
Time above 180mg/dl [%]	4.24	48.85	11.57	14.19	0.21
Time below 70mg/dl [%]	0.00	0.00	0.00	0.00	0.00

Table 26: Performance metrics of the closed-loop system on the 11 in-silico subjects.

Another important tool for the evaluation of the performance of our method is the so-called Control-Variability Grid Analysis (CVGA) introduced in [97]. This is a graphical tool that summarizes two important pieces of information, namely the minimum and the maximum glucose levels that were reached in closed loop. Therefore, we would like each subject to have similar minimum and maximum glucose levels, with the minimum being above the hypoglycemic threshold (70 or ideally 90mg/dl) and the maximum laying below the hyperglycemic threshold, 180mg/dl, or below the severe hyperglycemic threshold, 300mg/dl. The CVGA related to the in-silico subjects treated with the method is shown in Figure 19 and proves the general quality of the controller, given that:

- 2 subjects never reached hypoglycemia and hyperglycemia situations (A zone);
- 2 subjects experienced some benign hyperglycemia events and approached the euglycemic range lower bound (B zone);
- 6 subjects experienced some benign hyperglycemia events, without approaching the euglycemic range lower bound (Upper B zone);
- One subject, in particular Adult 7, ran into some severe hyperglycemia events, but never approached hypoglycemic levels (Upper C zone).

Figure 20 displays the evolution of the measured subcutaneous glucose level over the three-day simulation for three representative subjects. In detail, it is possible to see how the controller was able to keep most subjects within the euglycemic range for most of the time, responding promptly to carbohydrate intakes with well-computed boluses, while being too conservative in the case of Adult 7, who experienced several long-lasting hyperglycemia events due to an insufficient infusion of basal insulin and small boluses.

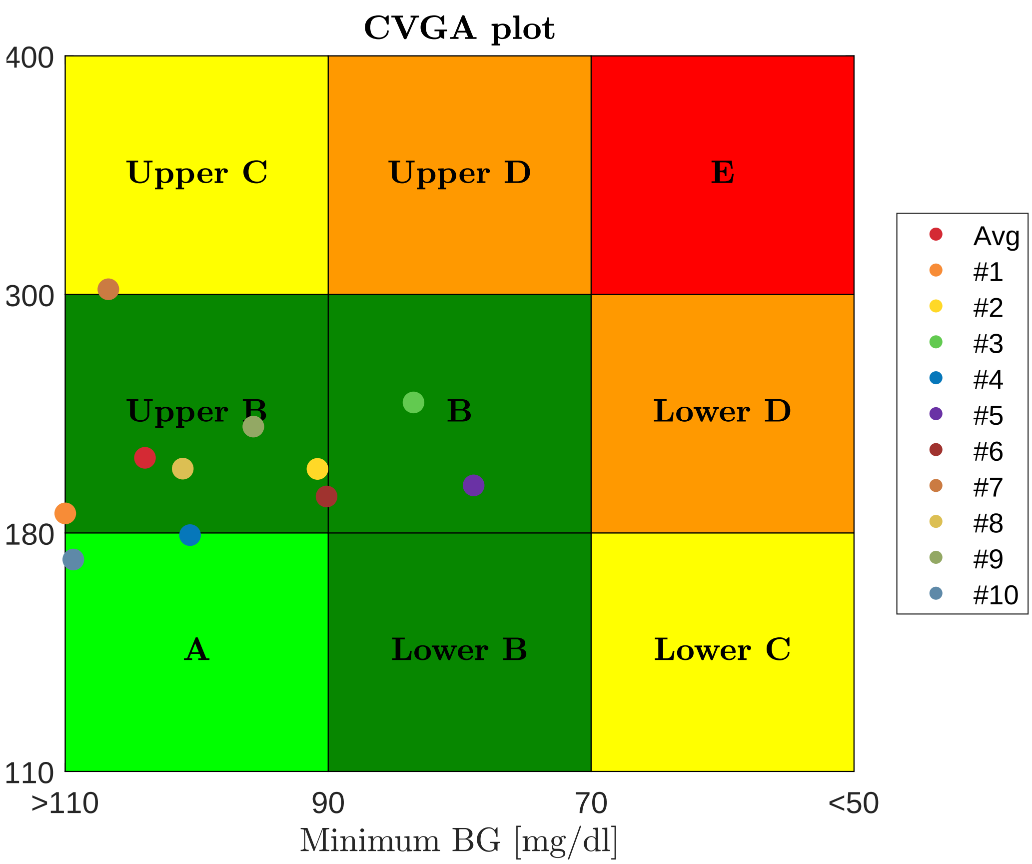


Figure 19: CVGA of the simulations run for the 11 adult subjects. Adult 7 was the only one reaching noticeable hyperglycemia values, while the other subjects were treated successfully with the MPC.

In addition to achieving encouraging results, the proposed MPC always had execution times that were consistent with the chosen sampling time. Indeed, as shown in Figure 21, only the execution time of the controller built for adult 6 approached the sampling time, but never exceeded it.

5.7 Conclusion

In this chapter, we presented a zone MPC based on GPs for the regulation of the subcutaneous glucose level in T1DM subjects. In particular, given the complexity of the endocrine system, GPs are

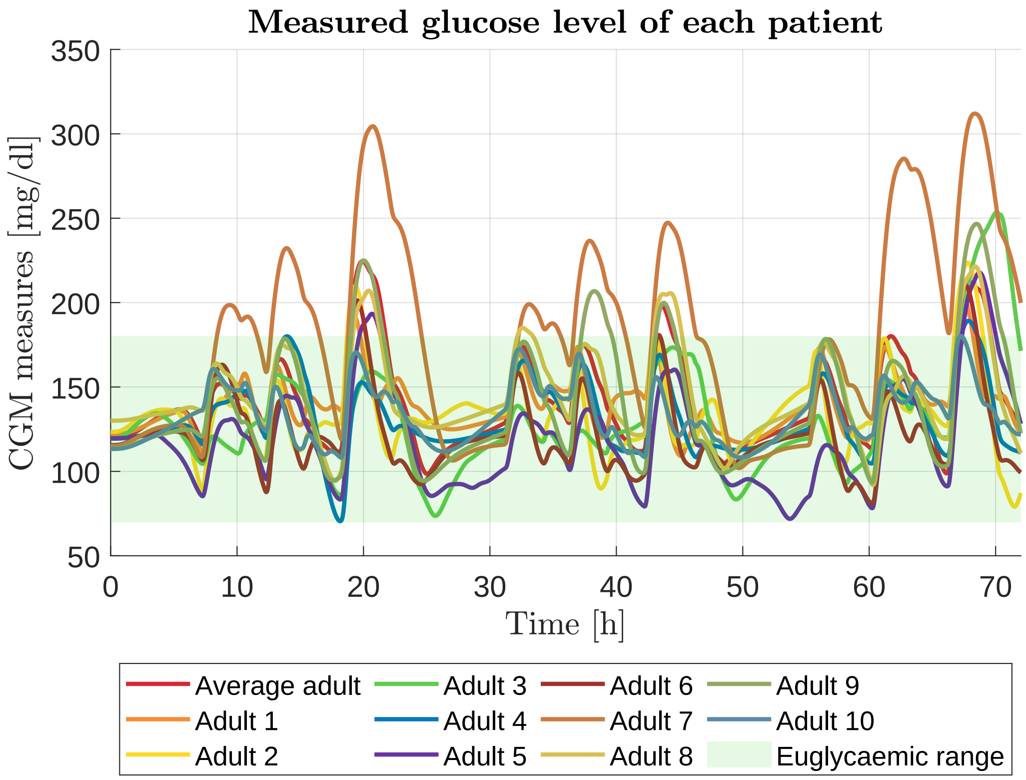


Figure 20: Evolution of the blood glucose level in the treated in-silico patients. The plots confirm the information provided by the previously addressed performance metrics. Indeed, none of the treated subjects experienced hypoglycemia, while non-severe hyperglycemic events were more common.

considered to be employed for its modeling, in order to avoid the need of a first-principle model. GPR is used to make predictions and the standard deviation of the resulting posterior distribution is used to restrict the soft constraints imposed in the optimal control problem. The method was tested on 11 in-silico diabetic subjects, for which a sufficient amount of data were previously collected to enable the training of the needed GPs. The obtained results confirm the good potential of GP-MPC schemes for the regulation of the glucose level in T1DM patients. In particular, none of the treated subjects experienced hypoglycemia, while sporadic hyperglycemic events were reported for several subjects. Only one of the treated subjects was ineffectively treated and reached severe hyperglycemic levels. Nonetheless, it would be interesting to study GP-MPC formulations that can update the employed model online, in order to take into account the time-varying nature of the endocrine system, and to elaborate a design method that does not require a stressful data collection phase and a manual model selection.

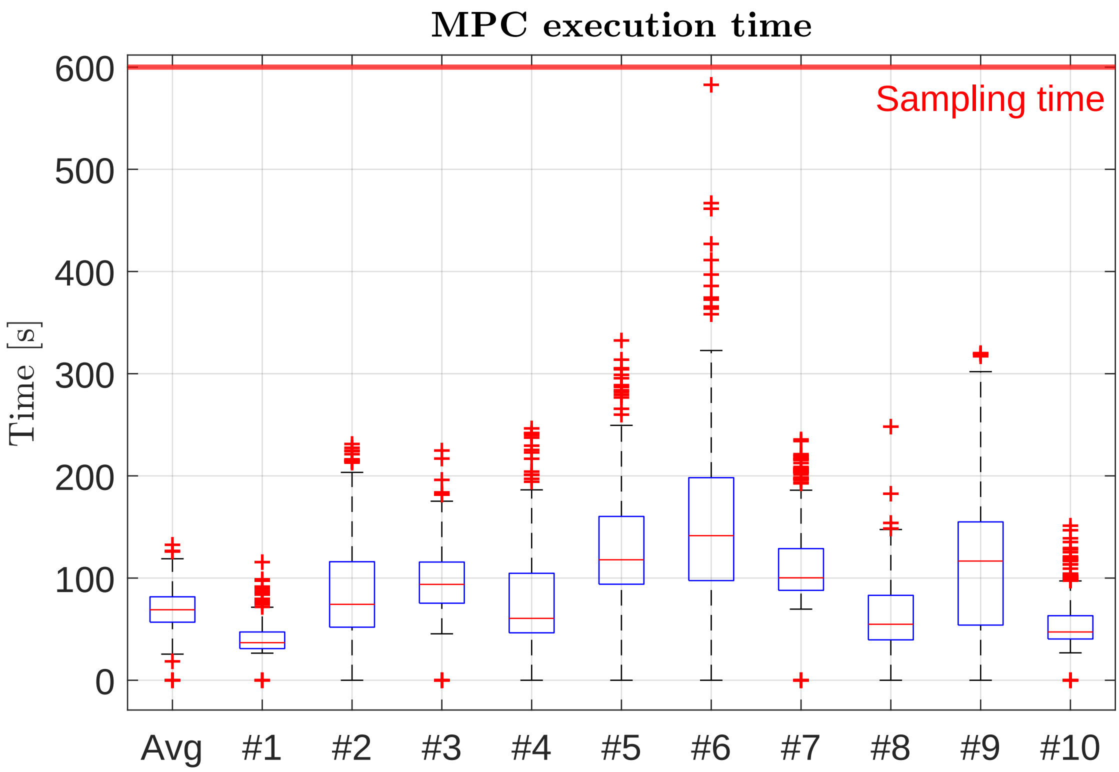


Figure 21: Box plots of the single MPC step execution times related to each subject. The execution time of a single MPC step was never greater than the sampling time.

Chapter 6. Conclusions and future developments

This chapter summarizes the content and the contributions of this book and proposes some research lines that could arise in the future starting from the results that are presented in this manuscript.

6.1 Summary of the contributions

This book aims at providing novel MPC formulations for the control of nonlinear systems that take into account the perturbations acting on the system to ensure robustness and that can be designed and implemented in contexts in which the design of standard MPC schemes would be too challenging.

Chapter 1 reviewed different MPC schemes for the control of nonlinear systems. Formulations for the regulation of nonlinear systems to a fixed setpoint were first addressed, focusing on different ways of formulating the stability-related terminal ingredients and commenting the advantages and drawbacks of each formulation. Then, MPC schemes designed to ensure closed-loop stability and recursive feasibility also in case of changing (and potentially infeasible) setpoints were analyzed. Finally, the problem of dealing with perturbations acting on the system was addressed, explaining how robust MPC schemes are designed to ensure input-to-state stability and recursive feasibility when such perturbations are bounded.

Chapter 2 presented a novel tube-based MPC for tracking changing setpoints that can be applied to a wide family of nonlinear systems. The conditions that the system under control and the employed cost function have to verify to enable the implementation of such MPC were listed, and a proof of recursive feasibility and input-to-state stability of the closed loop was provided. In particular, it was shown that the closed-loop system output is ensured to converge to a bounded region containing the best admissible steady output, also thanks to the exploitation of a robust invariant set for tracking as constraint set on both the terminal state and the artificial reference.

Chapter 3 returned to regulation problems, focusing on the problem of designing the terminal ingredients for perturbed nonlinear systems whose linearization at the desired equilibrium point is not fully controllable. Relying on the contractivity of a positive definite function, a property that is ensured by the stabilizability of the system under control, it was shown how contractive functions, which are easy to design, can replace the CLF that is usually employed in standard MPC schemes to ensure the stability of the closed-loop system. Moreover, though the knowledge of an RPI is still required to guarantee the recursive feasibility and the stabilizing capabilities of the robust contraction-based MPC, it does not have to be explicitly used in a terminal constraint.

Chapter 4 leveraged the theoretical concepts explained in Chapter 3 to show under what conditions an unknown nonlinear system modeled by means of a GP can be effectively controlled by a robust contraction-based MPC. In particular, a simple Lipschitz continuity assumption on both the latent function and the employed GP mean and covariance functions and a boundedness assumption on the Gaussian noise acting on the system were relied upon to prove that nonlinear systems modeled by GPs can be effectively controlled by a robust contraction-based MPC. Therefore, the resulting contractive GP-MPC inherits both the recursive feasibility and the stabilizing capability of the robust contraction-based MPC presented in Chapter 3.
Finally, Chapter 5 examined the effectiveness of GP-MPC schemes in biomedical applications, addressing the specific problem of controlling the glucose level in T1DM subjects, a control problem that is challenging due to the presence of uncontrolled inputs, i.e. meals, and to the complexity and time-varying nature of the human endocrine system. Considering the employment of a common AP, a zone MPC was designed that makes predictions about the glucose level of a certain subject by means of a GP that is trained on previously collected data. The results were satisfying and showed that GPs and other black-box learning methods can play an important role in future AID systems.

6.2 Limitations and future works

While the results achieved are satisfactory, the theoretical and applicative contributions of this volume have limitations that merit further investigation.
The issues encountered during the development of the tube-based MPC for tracking changing setpoints are first addressed. While the theoretical results obtained are sound and based on meticulous mathematical proofs, the computation of the ingredients ensuring its effectiveness is complex. Indeed:

- The theory works for any component-wise uniformly continuous system, but a practical way of computing all the needed ingredients was provided only for the smaller family of component-wise Lipschitz continuous systems.
- How to compute pre-stabilizing control laws $\pi(y_s, x, v)$ that can reduce the conservativeness of the computed tubes $\forall y_s$ is still a subject to be investigated.
- The employed terminal ingredients require to know the contractive sets $\Omega(y_s)$, but finding a family of pre-stabilizing control laws and a robust invariant set for tracking that meet all the needed conditions is challenging. In practice, it is difficult to implement such MPC scheme when the perturbations acting on the system are large.

- Only a scheme employing complete terminal ingredients was provided.

These are the main reasons why further work on practical methods for computing the terminal ingredients would be beneficial, as well as development of a version of the tube-based MPC for tracking without a constraint on the terminal state.
Regarding the proposed robust contraction-based MPC for regulation:

- Its usage requires either to solve a mixed-integer problem or two standard nonlinear optimization problems, thus making the implementation of such MPC scheme challenging in case of limited technological resources.

- Even though contractivity enables the closed-loop input-to-state (practical) stability proof, it would be desirable to have further information about how to choose a contractive function that meets all the conditions.

For these reasons, it would be interesting to investigate how to simplify the optimization problem(s) to be solved, and how to choose a contractive function that could enable the usage of short prediction horizons and extend the region of attraction of the MPC. Moreover, another appealing line of research could be that of applying contractivity to MPC schemes for tracking changing setpoints.
The robust contractive GP-MPC that was presented in Chapter 4 can surely limit the time spent in system modeling and its stability properties are inherited by the robust contraction-based MPC. Nonetheless:

- The choice of bounding the prediction error, which is theoretically Gaussian distributed, is reasonable from a pure engineering point of view, but limits the mathematical soundness of the proposed method.

- GPR provides posterior probability distributions, but the method just exploits the expected value of such distributions and relies on the Lipschitz continuity of the model to build the tightened constraints that guarantee robustness, hence wasting information.

Thus, it would be interesting to edit the controller so as to take into account the stochastic nature of GPs and deal with the problem of recursive feasibility in a probabilistic way, therefore allowing for prediction error distributions with infinite support. Finally, the employment of multiple-output GPs could improve the quality of the trained models.
Finally, the GP-MPC designed for the regulation of the glucose level in T1DM subjects needs substantial adjustments in all its facets:

- First of all, the time-varying nature of the endocrine system should not be ignored. Moreover, further perturbations should be taken into account, e.g. physical exercise. All these issues require to rethink the design of the GP and to give the MPC the possibility to update the employed model based on new data.
- The goal of AID systems is not just that of regulating the glucose level of a diabetic subject. In fact, it is also important to limit the quantity of insulin that has been injected and not yet absorbed, the so-called *insulin on board*. The MPC does not bound this quantity and just relies on its low aggressiveness to prevent the insulin on board from exceeding dangerous levels.
- The training of GPs in the AID context requires data that can represent all the situations a diabetic subject can run into, which translates in the need of conducting a data collection phase that is physically and mentally stressful for the subject.
- The designed MPC was not mathematically synthesized to ensure particular closed-loop stability properties.

Consequently, there is much that can be done to improve the results, the soundness and the reliability of the method.

Appendices

Appendix A

Stability theory

This appendix introduces the main concepts about the stability of discrete-time systems. In detail, some preliminary concepts that are needed to fully understand the stability-related definitions are first introduced, then the nominal stability of discrete-time nonlinear systems is analyzed, i.e. with no disturbances and uncertainties, and finally the idea of input-to-state stability is introduced.

A.1 Comparison functions

Definition A.1: $\mathcal{K}$, $\mathcal{K}_\infty$, $\mathcal{KL}$, and $\mathcal{PD}$ functions

- *A function* $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ *is a* $\mathcal{K}$ *function if:*
  - *it is a continuous function;*
  - *it is strictly increasing, i.e.* $\alpha(a) > \alpha(b)$ *if* $a > b$*;*
  - $\alpha(0) = 0$.
- *A function* $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ *is a* $\mathcal{K}_\infty$ *function if it is a* $\mathcal{K}$ *function and* $\alpha(a) \to \infty$ *when* $a \to \infty$.
- *A function* $\beta : \mathbb{R}_{\geq 0} \times \mathbb{N} \to \mathbb{R}_{\geq 0}$ *is a* $\mathcal{KL}$ *function if:*
  - *the function* $\beta(a, k)$ *is a* $\mathcal{K}$ *function with respect to* a *for all fixed* $k \geq 0$*;*
  - *the function* $\beta(a, k)$ *is a decreasing function with respect to* k *for all fixed* $a \geq 0$, *and such that* $\beta(a, k) \to 0$ *when* $k \to \infty$.
- *A function* $\gamma : \mathbb{R} \to \mathbb{R}_{\geq 0}$ *is a* $\mathcal{PD}$ *(positive definite) function if* $\gamma(0) = 0$ *and* $\gamma(s) > 0$ *for all* $s \neq 0$.

Property A.1: Main properties of $\mathcal{K}$, $\mathcal{K}_\infty$, and $\mathcal{KL}$ functions

Consider two $\mathcal{K}$ *functions* $\alpha_1(\cdot)$ *and* $\alpha_2(\cdot)$ *defined in* $[0, a)$, *two* $\mathcal{K}_\infty$ *functions* $\alpha_3(\cdot)$ *and* $\alpha_4(\cdot)$, *and a* $\mathcal{KL}$ *function* $\beta(\cdot, \cdot)$. *Therefore:*

- *the function* $\alpha_1^{-1}(\cdot)$ *is a* $\mathcal{K}$ *function defined in* $[0, \alpha_1(a))$*;*
- *the function* $\alpha_3^{-1}(\cdot)$ *is a* $\mathcal{K}_\infty$ *function;*

- *the composition of* $\alpha_1(\cdot)$ *and* $\alpha_2(\cdot)$, $\alpha_1 \circ \alpha_2$, *is a* $\mathcal{K}$ *function;*
- *the composition of* $\alpha_3(\cdot)$ *and* $\alpha_4(\cdot)$, $\alpha_3 \circ \alpha_4$, *is a* $\mathcal{K}_\infty$ *function;*
- *the composition of a* $\mathcal{K}$ *function and a* $\mathcal{KL}$ *function,* $\alpha_1 \circ \beta$, *is a* $\mathcal{KL}$ *function;*
- *the function* $\alpha_5(s) \coloneqq \max\{\alpha_1(s), \alpha_2(s)\}$ *is a* $\mathcal{K}$ *function;*
- *the function* $\alpha_6(s) \coloneqq \max\{\alpha_3(s), \alpha_4(s)\}$ *is a* $\mathcal{K}_\infty$ *function;*
- *the function* $\alpha_7(s) \coloneqq \alpha_1(s) + \alpha_2(s)$ *is a* $\mathcal{K}$ *function;*
- *the function* $\alpha_8(s) \coloneqq \alpha_3(s) + \alpha_4(s)$ *is a* $\mathcal{K}_\infty$ *function;*
- *given a* $\mathcal{K}_\infty$ *function,* $\alpha_3(\cdot)$, *there exists a* $\mathcal{K}_\infty$ *function* $\alpha_9(\cdot)$ *such that:*
  - $\alpha_9(s) \leq \alpha_3(s)$ *for all* $s > 0$*;*
  - *the function* $\alpha_{10}(s) \coloneqq s - \alpha_9(s)$ *is a* $\mathcal{K}$ *function, that is* $\alpha_9(s) < s$ *for all* $s > 0$*;*
- *if* $\alpha_3(s) \leq \alpha_4(s)$ *for all* $s \geq 0$, *then* $\alpha_3^{-1}(s) \geq \alpha_4^{-1}(s)$*;*
- *given* $a, b \in \mathbb{R}_{\geq 0}$, $\alpha_1(a+b) \leq \alpha_1(2a) + \alpha_1(2b)$*;*
- *given a* $\mathcal{K}_\infty$ *function,* $\alpha_3(\cdot)$, *there exist two* $\mathcal{K}_\infty$ *functions* $\alpha_{11}(\cdot), \alpha_{12}(\cdot)$ *such that* $\alpha_3(rs) \leq \alpha_{11}(r)\alpha_{12}(s)$, *for all* $r, s \in \mathbb{R}_{\geq 0}$.

A.2 Set invariance

A concept that is important to grasp when designing an MPC is that of set invariance. Indeed, sets with invariance properties are often employed in the design of the MPC stability-related terminal ingredients.

Definition A.2: Positively invariant set

A set $\Omega \subseteq \mathbb{R}^n$ *is a positively invariant set for the autonomous system* $x(k+1) = f(x(k))$ *if*

$$\forall x : x \in \Omega \implies f(x) \in \Omega.$$

Definition A.3: Robust positively invariant set (RPI)

A set $\Omega \subseteq \mathbb{R}^n$ is a robust positively invariant set for the system $x(k+1) = f(x(k), w(k))$, where $w \in \mathcal{W}$, if

$$\forall x : x \in \Omega \implies f(x, w) \in \Omega, \forall w \in \mathcal{W}.$$

A.3 Nominal stability of discrete-time nonlinear systems

We now define the main concepts about the nominal stability of a generic (continuous) discrete-time nonlinear system at the origin.

Definition A.4: Stability

An autonomous system $x(k+1) = f(x(k))$ is stable at the origin if, for all $\delta > 0$, a constant $\epsilon = \epsilon(\delta)$ exists such that

$$\forall x(0) : \|x(0)\| \leq \epsilon \implies \|x(k)\| \leq \delta \; \forall k \geq 0.$$

Definition A.5: Attractivity

The system $x(k+1) = f(x(k))$ is attractive on a set $\mathcal{X}$ if $\lim_{k\to\infty} x(k) = 0$ for all $x(0) \in \mathcal{X}$.

Mind that attractivity does not imply stability. In fact, the state of an attractive system can converge to the origin even though constants $\delta > 0$ and $\epsilon(\delta)$ verifying the conditions that are needed for stability do not exist.

Definition A.6: Asymptotic stability

An autonomous system $x(k+1) = f(x(k))$ is asymptotically stable at the origin if it is stable and there exists a constant $\epsilon > 0$ such that

$$\forall x(0) : \|x(0)\| \leq \epsilon \implies x(k) \to 0 \text{ for } k \to \infty.$$

Equivalently, the system $x(k+1) = f(x(k))$ is asymptotically stable at the origin if a $\mathcal{KL}$ function $\beta(\cdot,\cdot)$ exists such that

$$\|x(k)\| \leq \beta(\|x(0)\|, k)$$

for all $k \geq 0$ and all $x(0)$ that verify $\|x(0)\| \leq \epsilon$.

Alternatively, the $x(k+1) = f(x(k))$ is asymptotically stable at the origin if it is stable and attractive.

Definition A.7: Exponential stability

An autonomous system $x(k+1) = f(x(k))$ is exponentially stable at the origin if it is asymptotically stable and there exist constants $\epsilon > 0$, $a > 0$, and $\lambda \in [0, 1)$ that verify

$$\|x(k)\| \leq a \cdot \|x(0)\| \cdot \lambda^k$$

for all $k \geq 0$ and all $x(0)$ such that $\|x(0)\| \leq \epsilon$.

All the previous stability definitions are defined on a neighborhood of the origin, i.e. $\{x \in \mathbb{R}^n : \|x\| \leq \epsilon\}$. In this case the system is such to be locally stable. When such properties apply for the entire state space $\mathbb{R}^n$, the system is such to be globally stable.

A.4 Lyapunov stability theory

The Lyapunov stability theory poses sufficient conditions to guarantee the stability of an autonomous system, and is based on the existence of a positive definite function, called Lyapunov function, depending on whose properties it is possible to prove stability, asymptotic stability, and exponential stability.

Definition A.8: Lyapunov function

A function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ associated to a system $x(k+1) = f(x(k))$ is a Lyapunov function in $\mathcal{X}$ if there exist functions $\alpha_1(\cdot), \alpha_2(\cdot) \in \mathcal{K}_\infty$ such that the following conditions hold for any $x \in \mathcal{X}$:

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|)$$
$$V(f(x)) - V(x) \leq 0.$$

Theorem A.1

Given an autonomous system $x(k+1) = f(x(k))$ with an equilibrium at the origin, if there exists an associated Lyapunov function, then the origin is a stable equilibrium of the system. If these conditions hold in $\mathbb{R}^n$, then the origin is globally stable.

Theorem A.2

Given an autonomous system $x(k+1) = f(x(k))$ with an equilibrium at the origin, if there exists an associated Lyapunov function such that $V(f(x)) - V(x) \leq -\alpha_3(\|x\|)$, where $\alpha_3(\cdot)$ is *a $\mathcal{PD}$ function, then the origin is an asymptotically stable equilibrium of the system. If these conditions hold in $\mathbb{R}^n$, then the origin is globally asymptotically stable.*

Theorem A.3

Given an autonomous system $x(k+1) = f(x(k))$ with an equilibrium at the origin, if there exists an associated Lyapunov function $V(x)$ and constants $a > 0$, $b > 0$, $c > 0$ and $\sigma \geq 1$ such that

$$a\|x\|^\sigma \leq V(x) \leq b\|x\|^\sigma$$

and

$$V(f(x)) - V(x) \leq -c\|x\|^\sigma,$$

then the origin is an exponentially stable equilibrium of the system. If these conditions hold in $\mathbb{R}^n$, then the origin is globally exponentially stable.

It is also interesting to study how to apply the Lyapunov stability theory to controlled systems. We will not delve into details of this topic, but we provide the definition of control Lyapunov function (CLF), as it is employed in the stability proofs provided throughout the present book.

Definition A.9: Control Lyapunov function

A function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ associated to a system $x(k+1) = f(x(k), u(k))$ is a CLF in $\mathcal{X}$ if there exist functions $\alpha_1(\cdot), \alpha_2(\cdot), \alpha_3(\cdot) \in \mathcal{K}_\infty$ and a control law $\kappa(x)$ defined and feasible in $\mathcal{X}$ such that the following conditions hold for any $x \in \mathcal{X}$:

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|)$$
$$V(f(x, \kappa(x))) - V(x) \leq -\alpha_3(\|x\|).$$

A.5 Input-to-state stability

While in the perturbation-free case it is possible to use the Lyapunov stability theory to prove the stability of a system, the system evolution changes from what is expected when perturbations arise. Therefore, it is desirable to prove that the effect of the aforementioned perturbations is bounded and depends on the size of such perturbations. This is the core concept behind the idea of input-to-state stability, whose formal definition is provided in the following.

Definition A.10: Input-to-state stability

A system $x(k+1) = f(x(k), w(k))$ is input-to-state stable (ISS) with respect to w if there exist a $\mathcal{KL}$ function $\beta(\cdot,\cdot)$ and a $\mathcal{K}$ function $\alpha(\cdot)$ such that

$$\|x(k)\| \leq \beta(\|x(0)\|, k) + \sup_{j\in[0,k]} \alpha(\|w(j)\|).$$

An extension to input-to-state stability is the so-called input-to-state practical stability, in which the norm of the state is not considered to be bounded only by the $\mathcal{KL}$ function $\beta(\cdot,\cdot)$ and the $\mathcal{K}$ function $\alpha(\cdot)$, but also by a constant c, as stated in the following definition.

Definition A.11: Input-to-state practical stability

A system $x(k+1) = f(x(k), w(k))$ is input-to-state practically stable (ISpS) with respect to w if there exist a $\mathcal{KL}$ function $\beta(\cdot,\cdot)$, a $\mathcal{K}$ function $\alpha(\cdot)$ and a constant $c > 0$ such that

$$\|x(k)\| \leq \beta(\|x(0)\|, k) + \sup_{j\in[0,k]} \alpha(\|w(j)\|) + c.$$

Definition A.12: ISpS-Lyapunov function

A function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is an ISpS-Lyapunov function for the system $x(k+1) = f(x(k), w(k))$, with respect to w, in a RPI Γ containing the origin if there exists a compact set $\Omega \subseteq \Gamma$, with $\{0\} \in \Omega$, such that:

$$V(x) \geq \alpha_1(\|x\|),\ \forall x \in \Gamma,$$

$$V(x) \leq \alpha_2(\|x\|) + c_1,\ \forall x \in \Omega,$$

$$V(f(x,w)) - V(x) \leq -\alpha_3(\|x\|) + \gamma(\|w\|) + c_2, \forall x \in \Gamma, w \in \mathcal{W},$$

where $\alpha_1(\cdot)$, $\alpha_2(\cdot)$ and $\alpha_3(\cdot)$ are $\mathcal{K}_\infty$ functions, $\gamma(\cdot)$ is a $\mathcal{K}$ function, $c_1 \geq 0$ and $c_2 \geq 0$. If $c_1 = c_2 = 0$, then $V(x)$ is called an ISS-Lyapunov function.

Theorem A.4

If the system $x(k+1) = f(x(k), w(k))$ admits an IS(p)S-Lyapunov function in Γ with respect to w, then it is IS(p)S in Γ with respect to w.

Appendix B

Gaussian distribution properties

This appendix presents the main properties of Gaussian distributions, which are particularly useful for the comprehension of Section 4.2.

Considering a random variable A that takes values $a \in \mathbb{R}^{n_{\mathcal{D}}}$ and is Gaussian distributed, i.e. $A \sim \mathcal{N}(\mu, \Sigma)$, where μ is the mean vector and Σ is the covariance matrix, the probability density function (PDF) of A evaluated in a is

$$p(a) \coloneqq \frac{1}{\sqrt{(2\pi)^{n_{\mathcal{D}}} \det(\Sigma)}} \exp\left[(a-\mu)^\top \Sigma^{-1} (a-\mu)\right].$$

The *joint* distribution of two Gaussian random variables A and B evaluated in a and b is

$$p(a, b) = \mathcal{N}\left(\begin{bmatrix} \mu_a \\ \mu_b \end{bmatrix}, \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{bmatrix}\right). \tag{B.1}$$

The *marginal* distributions evaluated in a and b can be obtained from (B.1) as

$$p(a) = \mathcal{N}(\mu_a, \Sigma_{aa}),$$

$$p(b) = \mathcal{N}(\mu_b, \Sigma_{bb}),$$

while the *conditional* distribution of A given $B = b$ evaluated in a is

$$p(a|b) = \mathcal{N}(\mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(b - \mu_b), \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}).$$

Appendix C

Uncertain quadruple-tank system

The quadruple-tank system is a multivariable plant with nonlinear dynamics and both input and state constraints that is often employed as a benchmark for the evaluation of the performances of controllers for nonlinear systems. In this book, the quadruple-tank system is exploited to evaluate the performances of the proposed robust MPC formulations presented in Chapters 2 and 3.

The standard continuous-time equations of the quadruple-tank system are:

$$\frac{dh_1(t)}{dt} = -\frac{a_1}{S}\sqrt{2gh_1(t)} + \frac{a_3}{S}\sqrt{2gh_3(t)} + \frac{\gamma_a}{3600S}q_a(t), \tag{C.1a}$$

$$\frac{dh_2(t)}{dt} = -\frac{a_2}{S}\sqrt{2gh_2(t)} + \frac{a_4}{S}\sqrt{2gh_4(t)} + \frac{\gamma_b}{3600S}q_b(t), \tag{C.1b}$$

$$\frac{dh_3(t)}{dt} = -\frac{a_3}{S}\sqrt{2gh_3(t)} + \frac{(1-\gamma_b)}{3600S}q_b(t), \tag{C.1c}$$

$$\frac{dh_4(t)}{dt} = -\frac{a_4}{S}\sqrt{2gh_4(t)} + \frac{(1-\gamma_a)}{3600S}q_a(t), \tag{C.1d}$$

In particular, in the present book the employed parameters, the state bounds and the input bounds are the same as in [90] and are reported in Table 27 for the sake of completeness.

Parameter	Value	Description
$h_{1,\max}$	1.36 m	Maximum level of the tank 1
$h_{2,\max}$	1.36 m	Maximum level of the tank 2
$h_{3,\max}$	1.30 m	Maximum level of the tank 3
$h_{4,\max}$	1.30 m	Maximum level of the tank 4
$h_{\min}$	0.2 m	Minimum level (all tanks)
$q_{a,\max}$	3.6 m^3/h	Maximum flow of pump A
$q_{b,\max}$	4.0 m^3/h	Maximum flow of pump B
$q_{\min}$	0 m^3/h	Minimum flow of pumps A and B
a_1	1.2938e-4 m^2	Section of the hole in tank 1
a_2	1.5041e-4 m^2	Section of the hole in tank 2
a_3	1.0208e-4 m^2	Section of the hole in tank 3
a_4	9.3258e-5 m^2	Section of the hole in tank 4
S	0.06 m^2	Cross-section of all tanks
$\bar{\gamma}_a$	0.3	Nominal value of γ_a
$\bar{\gamma}_b$	0.4	Nominal value of γ_b

Table 27: Parameters of the quadruple-tank system.

It is important to note the importance of the parameters of the two three-way valves, γ_a and γ_b. Indeed, different combinations of values for these two parameters can lead to the quadruple-tank plant having minimum or nonminimum phase multivariable zeros. In order to test the controllers presented in Chapters 2 and 3 in difficult conditions, we suppose the values of such parameters to be uncertain, i.e.

$$\begin{aligned} \gamma_a &= \bar{\gamma}_a + w_1, \\ \gamma_b &= \bar{\gamma}_b + w_2, \end{aligned} \tag{C.2}$$

where $\bar{\gamma}_a = 0.3$ and $\bar{\gamma}_b = 0.4$ are the nominal values of such parameters, which are chosen in order to obtain a system with nonminimum phase multivariable zeros, while w_1 and w_2 represent bounded uncertainties on the parameter values whose bounds are defined differently in each example.

List of Figures

List of Tables

References

[1] F. A. Cuzzola, J. C. Geromel, and M. Morari. An improved approach for constrained robust model predictive control. *Automatica*, 38(7):1183–1189, 2002. ISSN 0005-1098.

[2] P. Abuin, P. S. Rivadeneira, A. Ferramosca, and A. H. González. Artificial pancreas under stable pulsatile MPC: Improving the closed-loop performance. *Journal of Process Control*, 92: 246–260, 2020.

[3] A. Afram, F. Janabi-Sharifi, A. S. Fung, and K. Raahemifar. Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: A state of the art review and case study of a residential HVAC system. *Energy and Buildings*, 141:96–113, 2017. ISSN 0378-7788.

[4] M. Alamir. *Stabilization of nonlinear systems using receding-horizon control schemes: a parametrized approach for fast systems*, volume 339. Springer, 2006.

[5] M. Alamir. Contraction-based nonlinear model predictive control formulation without stability-related terminal constraints. *Automatica*, 75(1):288–292, 2017.

[6] M. Alamir and G. Bornard. Stability of a truncated infinite constrained receding horizon scheme: the general discrete nonlinear case. *Automatica*, 31(9):1353–1356, 1995. ISSN 0005-1098.

[7] M. Althoff. An introduction to CORA 2015. In *Proc. of the 1st and 2nd Workshop on Applied Verification for Continuous and Hybrid Systems*, pages 120–151. EasyChair, December 2015.

[8] D. Angeli, A. Casavola, and E. Mosca. Command governors for constrained nonlinear systems: direct nonlinear vs. linearization-based strategies. *International Journal of Robust and Nonlinear Control*, 9(10):677–699, 1999.

[9] W. Ariza Ramirez, Z. Q. Leong, H. Nguyen, and S. G. Jayasinghe. Non-parametric dynamic system identification of ships using multi-output gaussian processes. *Ocean Engineering*, 166: 26–36, 2018. ISSN 0029-8018.

[10] K. J. Åström. *Introduction to stochastic control theory*. Courier Corporation, 2012.

[11] D. Atherton. Early developments in nonlinear control. *IEEE Control Systems Magazine*, 16(3): 34–43, 1996.

[12] S. Bahremand, H. S. Ko, R. Balouchzadeh, H. Felix Lee, S. Park, and G. Kwon. Neural network-based model predictive control for type 1 diabetic rats on artificial pancreas system. *Medical & biological engineering & computing*, 57(1):177–191, 2019.

[13] A. Bemporad, A. Casavola, and E. Mosca. Nonlinear control of constrained linear systems via predictive reference management. *IEEE Transactions on Automatic Control*, 42(3):340–349, 1997.

[14] A. Bemporad, F. Borrelli, and M. Morari. Min-max control of constrained uncertain discrete-time linear systems. *IEEE Transactions on Automatic Control*, 48(9):1600–1606, 2003.

[15] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer. Data-driven model predictive control with stability and robustness guarantees. *IEEE Transactions on Automatic Control*, 66(4): 1702–1717, 2020.

[16] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer. Linear tracking mpc for nonlinear systems—part i: The model-based case. *IEEE Transactions on Automatic Control*, 67(9): 4390–4405, 2022.

[17] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer. Linear tracking mpc for nonlinear systems—part ii: The data-driven case. *IEEE Transactions on Automatic Control*, 67(9): 4406–4421, 2022.

[18] R. Bitmead, M. Gevers, and V. Wertz. Adaptive optimal control the thinking man's gpc. *SERBIULA (sistema Librum 2.0)*, 01 1990.

[19] D. Boiroux, V. Bátora, M. Hagdrup, S. L. Wendt, N. K. Poulsen, H. Madsen, and J. B. Jørgensen. Adaptive model predictive control for a dual-hormone artificial pancreas. *Journal of Process Control*, 68:105–117, 2018.

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

[21] E. Bradford, L. Imsland, D. Zhang, and E. A. del Rio Chanona. Stochastic data-driven model predictive control using gaussian processes. *Computers & Chemical Engineering*, 139:106844, 2020.

[22] E. Camacho and C. Alba. *Model Predictive Control*. Advanced Textbooks in Control and Signal Processing. Springer London, 2013. ISBN 9780857293985.

[23] G. Cao, E. M.-K. Lai, and F. Alam. Gaussian process model predictive control of an unmanned quadrotor. *Journal of Intelligent & Robotic Systems*, 88:147–162, 2017.

[24] C. Chen and L. Shaw. On receding horizon feedback control. *Automatica*, 18(3):349–352, 1982. ISSN 0005-1098.

[25] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.

[26] L. Chisci and G. Zappa. Dual mode predictive tracking of piecewise constant references for constrained linear systems. *International Journal of Control*, 76(1):61–72, 2003.

[27] L. Chisci, J. A. Rossiter, and G. Zappa. Systems with persistent disturbances: predictive control with restricted constraints. *Automatica*, 37(7):1019–1028, 2001. ISSN 0005-1098.

[28] A. Cotorruelo, D. R. Ramirez, D. Limon, and E. Garone. Nonlinear mpc for tracking for a class of nonconvex admissible output sets. *IEEE Transactions on Automatic Control*, 66(8): 3726–3732, 2021.

[29] J. Coulson, J. Lygeros, and F. Dörfler. Data-enabled predictive control: In the shallows of the DeePC. In *2019 18th European Control Conference (ECC)*, pages 307–312. IEEE, 2019.

[30] J. Coulson, J. Lygeros, and F. Dörfler. Distributionally robust chance constrained data-enabled predictive control. *IEEE Transactions on Automatic Control*, 67(7):3289–3304, 2022.

[31] V. M. Cunha and T. L. M. Santos. Robust nonlinear model predictive control based on nominal predictions with piecewise constant references and bounded disturbances. *International Journal of Robust and Nonlinear Control*, 32(6):3944–3968, 2022.

[32] G. De Nicolao, L. Magni, and R. Scattolini. Stabilizing receding-horizon control of nonlinear time-varying systems. *IEEE Transactions on Automatic Control*, 43(7):1030–1036, 1998.

[33] M. De Paula, G. G. Acosta, and E. C. Martínez. On-line policy learning and adaptation for real-time personalization of an artificial pancreas. *Expert Systems with Applications*, 42(4): 2234–2255, 2015.

[34] B. Decardi-Nelson and J. Liu. Computing robust control invariant sets of constrained nonlinear systems: A graph algorithm approach. *Computers & Chemical Engineering*, 145:107177, 2021. ISSN 0098-1354.

[35] M. P. Deisenroth. *Efficient Reinforcement Learning using Gaussian Processes*. PhD thesis, 2010.

[36] M. P. Deisenroth and C. E. Rasmussen. Pilco: a model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 465–472, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

[37] M. P. Deisenroth, D. Fox, and C. E. Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37 (2):408–423, 2015.

[38] S. Del Favero, C. Toffanin, L. Magni, and C. Cobelli. Deployment of modular MPC for type 1 diabetes control: the Italian experience 2008–2016. In *The Artificial Pancreas*, pages 153–182. Elsevier, 2019.

[39] M. Diehl. Formulation of closed-loop min–max MPC as a quadratically constrained quadratic program. *IEEE Transactions on Automatic Control*, 52(2):339–343, 2007.

[40] M. Diehl and J. Bjornberg. Robust dynamic programming for min-max model predictive control of constrained uncertain systems. *IEEE Transactions on Automatic Control*, 49(12):2253–2257, 2004.

[41] A. D'Jorge, B. F. Santoro, A. Anderson, A. H. González, and A. Ferramosca. Stochastic model predictive control for tracking linear systems. *Optimal Control Applications and Methods*, 41 (1):65–83, 2020.

[42] S. Dutta, T. Kushner, and S. Sankaranarayanan. Robust data-driven control of artificial pancreas systems using neural networks. In *International Conference on Computational Methods in Systems Biology*, pages 183–202. Springer, 2018.

[43] E. Fernández Cara and E. Zuazua Iriondo. Control theory: History, mathematical achievements and perspectives. *Boletín de la Sociedad Española de Matemática Aplicada, 26, 79-140.*, 2003.

[44] A. Ferramosca, D. Limon, I. Alvarado, T. Alamo, and E. Camacho. MPC for tracking with optimal closed-loop performance. *Automatica*, 45(8):1975–1978, 2009. ISSN 0005-1098.

[45] A. Ferramosca, D. Limon, A. González, I. Alvarado, and E. Camacho. Robust MPC for tracking zone regions based on nominal predictions. *Journal of Process Control*, 22(10):1966–1974, 2012. ISSN 0959-1524.

[46] G. Galuppini, L. Magni, and A. Ferramosca. Nonlinear mpc for tracking piecewise-constant reference signals: the positive semidefinite stage cost case. *IFAC-PapersOnLine*, 56(1):210–215, 2023. ISSN 2405-8963. 12th IFAC Symposium on Nonlinear Control Systems NOLCOS 2022.

[47] C. E. García, D. M. Prett, and M. Morari. Model predictive control: Theory and practice - a survey. *Automatica*, 25(3):335–348, 1989. ISSN 0005-1098.

[48] 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.

[49] O. P. B. Gesù. Diabete nei bambini. `https://www.ospedalebambinogesu.it/diabete-in-eta-pediatrica-80052/`, 2024.

[50] R. Gondhalekar, E. Dassau, and F. J. Doyle III. Periodic zone-MPC with asymmetric costs for outpatient-ready safety of an artificial pancreas to treat type 1 diabetes. *Automatica*, 71: 237–246, 2016.

[51] A. H. González, P. S. Rivadeneira, A. Ferramosca, N. Magdelaine, and C. H. Moog. Impulsive zone mpc for type i diabetic patients based on a long-term model. *IFAC-PapersOnLine*, 50(1): 14729–14734, 2017.

[52] A. H. González, P. S. Rivadeneira, A. Ferramosca, N. Magdelaine, and C. H. Moog. Stable impulsive zone model predictive control for type 1 diabetic patients based on a long-term model. *Optimal Control Applications and Methods*, 41(6):2115–2136, 2020.

[53] G. A. Gregory, T. I. G. Robinson, S. E. Linklater, F. Wang, S. Colagiuri, C. de Beaufort, K. C. Donaghue, J. L. Harding, P. L. Wander, X. Zhang, X. Li, S. Karuranga, H. Chen, H. Sun, Y. Xie, R. Oram, D. J. Magliano, Z. Zhou, A. J. Jenkins, R. C. Ma, J. Maniam, T. J. Orchard, P. Rai, and G. D. Ogle. Global incidence, prevalence, and mortality of type 1 diabetes in 2021 with projection to 2040: a modelling study. *The Lancet Diabetes & Endocrinology*, 10(10): 741–760, Oct 2022. ISSN 2213-8587.

[54] G. Grimm, M. Messina, S. Tuna, and A. Teel. Model predictive control: for want of a local control lyapunov function, all is not lost. *IEEE Transactions on Automatic Control*, 50(5): 546–558, 2005.

[55] L. Grüne, J. Pannek, L. Grüne, and J. Pannek. *Nonlinear model predictive control*. Springer, 2017.

[56] I. Hajizadeh, M. Rashid, and A. Cinar. Plasma-insulin-cognizant adaptive model predictive control for artificial pancreas systems. *Journal of Process Control*, 77:97–113, 2019.

[57] L. Hewing, J. Kabzan, and M. N. Zeilinger. Cautious model predictive control using gaussian process regression. *IEEE Transactions on Control Systems Technology*, 28(6):2736–2743, 2019.

[58] R. Hovorka, V. Canonico, L. J. Chassin, U. Haueter, M. Massi-Benedetti, M. O. Federici, T. R. Pieber, H. C. Schaller, L. Schaupp, T. Vering, et al. Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes. *Physiological measurement*, 25(4):905, 2004.

[59] L. M. Huyett, E. Dassau, H. C. Zisser, and F. J. Doyle III. Design and evaluation of a robust PID controller for a fully implantable artificial pancreas. *Industrial & engineering chemistry research*, 54(42):10311–10321, 2015.

[60] J. Iqbal, M. Ullah, S. G. Khan, B. Khelifa, and S. Ćukovic. Nonlinear control systems - a brief overview of historical and recent advances. *Nonlinear Engineering*, 6(4):301–312, 2017.

[61] V. N. Joish, F. L. Zhou, R. Preblick, D. Lin, M. Deshpande, S. Verma, M. J. Davies, S. Paranjape, and J. Pettus. Estimation of annual health care costs for adults with type 1 diabetes in the united states. *J Manag Care Spec Pharm*, 26(3):311–318, Mar. 2020.

[62] R. E. Kalman and J. E. Bertram. Control System Analysis and Design Via the "Second Method" of Lyapunov: I—Continuous-Time Systems. *Journal of Basic Engineering*, 82(2):371–393, 06 1960. ISSN 0021-9223.

[63] P. Kanavos, S. Aardweg, and W. Schurer. Diabetes expenditure, burden of disease and management in 5 eu countries. *London School of Economics*, 01 2012.

[64] A. Katsarou, S. Gudbjörnsdottir, A. Rawshani, D. Dabelea, E. Bonifacio, B. J. Anderson, L. M. Jacobsen, D. A. Schatz, and Å. Lernmark. Type 1 diabetes mellitus. *Nature reviews Disease primers*, 3(1):1–17, 2017.

[65] E. C. Kerrigan and J. M. Maciejowski. Feedback min-max model predictive control using a single linear program: robust stability and the explicit solution. *International Journal of Robust and Nonlinear Control*, 14(4):395–413, 2004.

[66] K. Kiš and M. Klaučo. Neural network based explicit MPC for chemical reactor control. *arXiv preprint arXiv:1912.04684*, 2019.

[67] J. Kocijan. *Modelling and control of dynamic systems using Gaussian process models*. Springer, 2016.

[68] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and A. Girard. Gaussian process model based predictive control. In *Proceedings of the 2004 American control conference*, volume 3, pages 2214–2219. IEEE, 2004.

[69] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause. Learning-Based Model Predictive Control for Safe Exploration. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6059–6066, 2018.

[70] M. V. Kothare, V. Balakrishnan, and M. Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, 1996.

[71] B. Kouvaritakis and M. Cannon. *Model predictive control: Classical, Robust and Stochastic*. Springer, 2016.

[72] B. Kovatchev. Automated closed-loop control of diabetes: the artificial pancreas. *Bioelectronic Medicine*, 4(1):1–12, 2018.

[73] J. Köhler. *Analysis and design of MPC frameworks for dynamic operation of nonlinear constrained systems*. PhD thesis, 2021.

[74] J. Köhler, M. Müller, and F. Allgöwer. A nonlinear tracking model predictive control scheme for dynamic target signals. *Automatica*, 118:109030, 2020. ISSN 0005-1098.

[75] J. Köhler, M. A. Müller, and F. Allgöwer. A Nonlinear Model Predictive Control Framework Using Reference Generic Terminal Ingredients. *IEEE Transactions on Automatic Control*, 65 (8):3576–3583, 2020.

[76] J. Köhler, R. Soloperto, M. Müller, and F. Allgöwer. A Computationally Efficient Robust Model Predictive Control Framework for Uncertain Nonlinear Systems. *IEEE Transactions on Automatic Control*, 66(2):794–801, 2021.

[77] J. Köhler, M. A. Müller, and F. Allgöwer. Analysis and design of model predictive control frameworks for dynamic operation—an overview. *Annual Reviews in Control*, 57:100929, 2024. ISSN 1367-5788.

[78] N. Lanzetti, Y. Z. Lian, A. Cortinovis, L. Dominguez, M. Mercangöz, and C. Jones. Recurrent neural network based MPC for process industries. In *2019 18th European Control Conference (ECC)*, pages 1005–1010. IEEE, 2019.

[79] M. Lazar and M. Tetteroo. Computation of terminal costs and sets for discrete–time nonlinear MPC. *IFAC-PapersOnLine*, 51(20):141–146, 2018. ISSN 2405-8963. 6th IFAC Conference on Nonlinear Model Predictive Control NMPC 2018.

[80] M. Lazar, D. M. De La Peña, W. Heemels, and T. Alamo. Min-max nonlinear model predictive control with guaranteed input-tostate stability. In *17th Symposium on Mathematical Theory for Networks and Systems. Kyoto, Japan*. Citeseer, 2006.

[81] A. Lederer, J. Umlauft, and S. Hirche. Uniform error bounds for gaussian process regression with application to safe control. *Advances in Neural Information Processing Systems*, 32, 2019.

[82] E. Lee and L. Markus. *Foundations of Optimal Control Theory*. Wiley, New York, 1967.

[83] F. Li, H. Li, and Y. He. Adaptive stochastic model predictive control of linear systems using Gaussian process regression. *IET Control Theory & Applications*, 15(5):683–693, 2021.

[84] D. Limon and T. Alamo. Tracking Model Predictive Control. In *Encyclopedia of Systems and Control*, pages 1475–1484. Springer London, London, 2015. ISBN 978-1-4471-5058-9.

[85] D. Limon, T. Alamo, F. Salas, and E. F. Camacho. On the stability of constrained MPC without terminal constraint. *IEEE Transactions on Automatic Control*, 51(5):832–836, 2006.

[86] D. Limon, I. Alvarado, T. Alamo, and E. Camacho. MPC for tracking piecewise constant references for constrained linear systems. *Automatica*, 44(9):2382–2387, 2008. ISSN 0005-1098.

[87] D. Limon, T. Alamo, D. M. Raimondo, D. M. de la Peña, J. M. Bravo, A. Ferramosca, and E. F. Camacho. *Input-to-State Stability: A Unifying Framework for Robust Model Predictive Control*, pages 1–26. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-01094-1.

[88] D. Limon, I. Alvarado, T. Alamo, and E. Camacho. Robust tube-based MPC for tracking of constrained linear systems with additive disturbances. *Journal of Process Control*, 20(3): 248–260, 2010. ISSN 0959-1524.

[89] D. Limon, I. Alvarado, A. Ferramosca, T. Alamo, and E. F. Camacho. Enhanced robust NMPC based on nominal predictions *. *IFAC Proceedings Volumes*, 43(14):220–225, 2010. ISSN 1474-6670. 8th IFAC Symposium on Nonlinear Control Systems.

[90] D. Limon, A. Ferramosca, I. Alvarado, and T. Alamo. Nonlinear MPC for Tracking Piece-Wise Constant Reference Signals. *IEEE Transactions on Automatic Control*, 63(11):3735–3750, 2018.

[91] C. Liu, H. Li, J. Gao, and D. Xu. Robust self-triggered min–max model predictive control for discrete-time nonlinear systems. *Automatica*, 89:333–339, 2018. ISSN 0005-1098.

[92] B. T. Lopez, J.-J. E. Slotine, and J. P. How. Dynamic Tube MPC for Nonlinear Systems. In *2019 American Control Conference (ACC)*, pages 1655–1662, 2019.

[93] J. Maciejowski. *Predictive Control With Constraints*. Prentice Hall, 2002. ISBN 9780201398236.

[94] L. Magni and R. Scattolini. On the solution of the tracking problem for non-linear systems with MPC. *International Journal of Systems Science*, 36(8):477–484, 2005.

[95] L. Magni, G. Nicolao, L. Magnani, and R. Scattolini. A stabilizing model-based predictive control algorithm for nonlinear systems. *Automatica*, 37(9):1351–1362, 2001. ISSN 0005-1098.

[96] L. Magni, G. De Nicolao, R. Scattolini, and F. Allgöwer. Robust model predictive control for nonlinear discrete-time systems. *International Journal of Robust and Nonlinear Control*, 13 (3-4):229–246, 2003.

[97] L. Magni, D. M. Raimondo, C. D. Man, M. Breton, S. Patek, G. De Nicolao, C. Cobelli, and B. P. Kovatchev. Evaluating the efficacy of closed-loop glucose regulation via control-variability grid analysis. *Journal of Diabetes Science and Technology*, 2(4):630–635, 2008.

[98] M. Maiworm. *Gaussian process in control : model predictive control with guarantees and control of scanning quantum dot microscopy*. PhD thesis, 2021.

[99] M. Maiworm, D. Limon, and R. Findeisen. Online learning-based model predictive control with Gaussian process models and stability guarantees. *International Journal of Robust and Nonlinear Control*, 31(18):8785–8812, 2021.

[100] C. D. Man, F. Micheletto, D. Lv, M. Breton, B. Kovatchev, and C. Cobelli. The UVA/PADOVA type 1 diabetes simulator: new features. *Journal of diabetes science and technology*, 8(1): 26–34, 2014.

[101] J. M. Manzano. *Learning-based model predictive control for constrained nonlinear systems*. PhD thesis, 06 2020.

[102] J. M. Manzano, D. M. de la Peña, J.-P. Calliess, and D. Limon. Componentwise Hölder Inference for Robust Learning-Based MPC. *IEEE Transactions on Automatic Control*, 66(11): 5577–5583, 2021.

[103] D. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 35(7):814–824, 1990.

[104] D. Q. Mayne, M. M. Seron, and S. V. Rakovic. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005. ISSN 0005-1098.

[105] D. Q. Mayne, E. C. Kerrigan, E. van Wyk, and P. Falugi. Tube-based robust nonlinear model predictive control. *International Journal of Robust and Nonlinear Control*, 21(11):1341–1353, 2011.

[106] M. Messori, C. Toffanin, S. Del Favero, G. De Nicolao, C. Cobelli, and L. Magni. Model individualization for artificial pancreas. *Computer Methods and Programs in Biomedicine*, 171:133–140, 2019.

[107] S. J. Moon, I. Jung, and C.-Y. Park. Current advances of artificial pancreas systems: a comprehensive review of the clinical evidence. *Diabetes & Metabolism Journal*, 45(6):813–839, 2021.

[108] M. M. Morato, G. Q. Bao Tran, G. N. dos Reis, J. E. Normey-Rico, and O. Sename. NMPC Through qLPV Embedding: A Tutorial Review of Different Approaches. *IFAC-PapersOnLine*,

54(6):302–307, 2021. ISSN 2405-8963. 7th IFAC Conference on Nonlinear Model Predictive Control NMPC 2021.

[109] L. Ortmann, D. Shi, E. Dassau, F. J. Doyle, S. Leonhardt, and B. J. Misgeld. Gaussian process-based model predictive control of blood glucose for patients with type 1 diabetes mellitus. In *2017 11th Asian Control Conference (ASCC)*, pages 1092–1097. IEEE, 2017.

[110] G. Pannocchia and E. Kerrigan. Offset-free receding horizon control of constrained linear systems. *AIChE Journal*, 51(12):3134–3146, 2005.

[111] G. Pannocchia, J. B. Rawlings, and S. J. Wright. Conditions under which suboptimal nonlinear mpc is inherently robust. *Systems & Control Letters*, 60(9):747–755, 2011.

[112] A. K. Patra and P. K. Rout. Adaptive sliding mode Gaussian controller for artificial pancreas in TIDM patient. *Journal of Process Control*, 59:13–27, 2017.

[113] M. Polver, B. Sonzogni, M. Mazzoleni, F. Previdi, and A. Ferramosca. Artificial Pancreas under a Zone Model Predictive Control based on Gaussian Process models: toward the personalization of the closed loop. *IFAC-PapersOnLine*, 56(2):9642–9647, 2023.

[114] M. Polver, D. Limon, M. Mazzoleni, F. Previdi, and A. Ferramosca. Robust contraction-based Model Predictive Control for nonlinear systems. *arXiv*, 2024.

[115] M. Polver, D. Limon, F. Previdi, and A. Ferramosca. Robust contractive model predictive control based on Gaussian Processes. *arXiv*, 2024.

[116] M. Polver, D. Limon, F. Previdi, and A. Ferramosca. Robust tracking mpc for perturbed nonlinear systems. *IEEE Transactions on Automatic Control*, 2025.

[117] A. Propoĭ. Application of linear programming methodes for the synthesis of automatic sampled-data systems. *Avtom. Telemeh.*, pages 912–920, 1963.

[118] J. Quinonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.

[119] J. Quinonero-Candela, A. Girard, and C. E. Rasmussen. Prediction at an uncertain input for Gaussian processes and relevance vector machines-application to multiple-step ahead time-series forecasting. 2003.

[120] D. M. Raimondo, D. Limon, M. Lazar, L. Magni, and E. F. Camacho. Min-max model predictive control of nonlinear systems: A unifying overview on stability. *European Journal of Control*, 15(1):5–21, 2009.

[121] S. V. Rakovic and W. S. Levine. *Handbook of model predictive control*. Springer, 2018.

[122] S. V. Rakovic, B. Kouvaritakis, M. Cannon, C. Panos, and R. Findeisen. Parameterized Tube Model Predictive Control. *IEEE Transactions on Automatic Control*, 57(11):2746–2761, 2012.

[123] S. V. Rakovic, B. Kouvaritakis, R. Findeisen, and M. Cannon. Homothetic tube model predictive control. *Automatica*, 48(8):1631–1638, 2012. ISSN 0005-1098.

[124] S. V. Rakovic, W. S. Levine, and B. Açikmese. Elastic tube model predictive control. In *2016 American Control Conference (ACC)*, pages 3594–3599, 2016.

[125] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[126] J. B. Rawlings, D. Q. Mayne, M. Diehl, et al. *Model predictive control: theory, computation, and design*, volume 2. Nob Hill Publishing Madison, WI, 2017.

[127] N. Rosales, H. De Battista, and F. Garelli. Hypoglycemia prevention: PID-type controller adaptation for glucose rate limiting in Artificial Pancreas System. *Biomedical Signal Processing and Control*, 71:103106, 2022.

[128] J. A. Rossiter, B. Kouvaritakis, and J. Gossner. Guaranteeing feasibility in constrained stable generalised predictive control. *IEE Proceedings-Control Theory and Applications*, 143(5): 463–469, 1996.

[129] G. Schmelzeisen-Redeker, M. Schoemaker, H. Kirchsteiger, G. Freckmann, L. Heinemann, and L. Del Re. Time delay of cgm sensors: relevance, causes, and countermeasures. *Journal of Diabetes Science and Technology*, 9(5):1006–1015, 2015.

[130] D. Shi, E. Dassau, and F. J. Doyle. Adaptive zone model predictive control of artificial pancreas based on glucose-and velocity-dependent control penalties. *IEEE Transactions on Biomedical Engineering*, 66(4):1045–1054, 2018.

[131] E. Snelson and Z. Ghahramani. Sparse gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005.

[132] R. Soloperto, J. Köhler, and F. Allgöwer. A nonlinear mpc scheme for output tracking without terminal ingredients. *IEEE Transactions on Automatic Control*, 68(4):2368–2375, 2023.

[133] B. Sonzogni, J. M. Manzano, M. Polver, F. Previdi, and A. Ferramosca. Choki-based mpc for blood glucose regulation in artificial pancreas. *IFAC-PapersOnLine*, 56(2):9672–9677, 2023.

[134] B. Sonzogni, J. M. Manzano, M. Polver, F. Previdi, and A. Ferramosca. Choki-based mpc for blood glucose regulation in artificial pancreas with probabilistic constraints. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 1619–1624. IEEE, 2023.

[135] P. Soru, G. De Nicolao, C. Toffanin, C. Dalla Man, C. Cobelli, L. Magni, A. H. Consortium, et al. MPC based artificial pancreas: strategies for individualization and meal compensation. *Annual Reviews in Control*, 36(1):118–128, 2012.

[136] Z. Sun, S. J. Qin, A. Singhal, and L. Megan. Performance monitoring of model-predictive controllers via model residual assessment. *Journal of Process Control*, 23(4):473–482, 2013.

[137] S. Tang, K. Fujimoto, and I. Maruta. Learning dynamical systems using a novel multiple-output gaussian process model. In *Proceedings of the 2022 7th International Conference on Machine Learning Technologies*, ICMLT '22, page 260–265, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450395748.

[138] P. Tatjewski. Advanced control and on-line process optimization in multilayer structures. *IFAC Proceedings Volumes*, 40(9):13–26, 2007. ISSN 1474-6670. 11th IFAC Symposium on Large Scale Complex Systems Theory and Applications.

[139] M. Titsias. Variational learning of inducing variables in sparse gaussian processes. In D. van Dyk and M. Welling, editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 567–574, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR.

[140] C. Toffanin, M. Messori, F. Di Palma, G. De Nicolao, C. Cobelli, and L. Magni. Artificial pancreas: model predictive control design from clinical experience. *J Diabetes Sci Technol*, 7 (6):1470–1483, Nov. 2013.

[141] M. E. Villanueva, R. Quirynen, M. Diehl, B. Chachuat, and B. Houska. Robust MPC via min–max differential inequalities. *Automatica*, 77:311–321, 2017. ISSN 0005-1098.

[142] Z. Wan and M. V. Kothare. An efficient off-line formulation of robust model predictive control using linear matrix inequalities. *Automatica*, 39(5):837–846, 2003. ISSN 0005-1098.

[143] Z. Wan and M. V. Kothare. Efficient scheduled stabilizing model predictive control for constrained nonlinear systems. *International Journal of Robust and Nonlinear Control*, 13(3-4): 331–346, 2003.

[144] J. Willems, P. Rapisarda, Markovsky, and B. De Moor. A note on persistency of excitation. *Systems & Control Letters*, 54(4):325–329, Apr. 2005. ISSN 0167-6911.

[145] S. Yu, C. Maier, H. Chen, and F. Allgöwer. Tube MPC scheme based on robust control invariant set with application to Lipschitz nonlinear systems. *Systems & Control Letters*, 62(2):194–200, 2013. ISSN 0167-6911.

[146] A. Zamani and H. Bolandi. Robust MPC for tracking changing setpoints in discrete-time nonlinear systems with non-additive unknown disturbance. *IET Control Theory & Applications*, 16(9):846–860, 2022.

[147] M. A. Álvarez, L. Rosasco, and N. D. Lawrence. *Kernels for Vector-Valued Functions: A Review*. 2012.