DOCTORAL THESIS

Learning-based model predictive control for constrained nonlinear systems

José María Manzano Crespo

Seville, 2020

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

Doctoral Thesis

Learning-based model predictive control for constrained nonlinear systems

José María Manzano Crespo

Thesis submitted for the degree of Doctor of Philosophy in the University of Seville

---

Supervised by:

Daniel Limón Marruedo

David Muñoz de la Peña Sequedo

A mi familia,
mis amigos
y mis profesores

Acknowledgements
Agradecimientos

A Daniel Limón, por todo lo que he aprendido contigo estos años, de lo cual lo menos imporante que me llevo es todo lo que me has enseñado de control predictivo. Espero que el fin de esta tesis no suponga el fin de nuestros almuerzos, charlas y por supuesto del trabajo juntos. Muchas gracias, Dani.

A David Muñoz, por transmitirme su punto de vista y eficacia, que atesoro como útiles cualidades de lo que requiere ser buen investigador y docente. Muchas gracias por tantas horas de dedicación a todos los resultados de esta tesis.

Huge thanks to Jan Calliess, not only for receiving me in Oxford during my research stay, but also for all the work done together, and all the teachings and enlightening discussions that arose these years.

A Teo Álamo, por todas las ideas y debates de pizarra que han conseguido que mejore mi visión crítica de la investigación.

A los demás profesores del Grupo de Estimación, Predicción, Optimización y Control: Daniel Rodríguez, Ignacio Alvarado y Mario Pereira, por todos los consejos y ayuda prestada. A José Ramón Salvador y Pablo Krupa, por todos los años compartiendo despacho y etapa de doctorando, así como a los demás miembros del grupo y de despacho, Juan, Joaquín, Daniel, Víctor y Jose, y demás profesores de la Escuela Técnica Superior de Ingeniería. Muchas gracias a Bernardo Pérez, por hacer que me sintiera en Oxford como en casa.

A todos mis profesores durante todas las etapas de mi educación, como Ana Montero, que me enseñó el primer problema de matemáticas o Antonio Almansa, que hizo que las amase. Especialmente gracias a Manolo Fernández Arias: sin tus enseñanzas no estaría escribiendo hoy esta tesis.

Muchas gracias a toda mi familia y amigos, no sólo por vuestro apoyo durante estos años, sino también por vuestra contribución en mi educación. Todos habéis aportado para convertirme en quien soy hoy, y por tanto este trabajo es en parte vuestro; espero que os haga sentir orgullosos. Finalmente, me gustaría dedicar esta tesis a Marina, por su paciencia y ayuda los años que ha durado la elaboración.

Sevilla, 2020.

Abstract

This thesis stands upon the assumption that nothing is known about the dynamics of the system to be controlled. Instead, only inputs and outputs are measurable, and hence an historical set of these measurements may be available. The main objective is to control this system under efficient and safe conditions using only these measurements. To this aim, the class of learning rules known as kinky inference is used to model nonlinear output-feedback systems, in a model predictive control framework.

Hence, the contributions of this thesis are two-fold. First, the kinky inference methods are extended, proposing methods both to reduce the computational burden and to decrease the prediction errors. Second, robust learning-based predictive controllers are derived. These controllers are stable by design, and are able to ensure robust constraint satisfaction and to improve their performance by benefiting from new information gathered online.

Resumen

Esta tesis está dedicada al control de sistemas, bajo la hipótesis de que no se conoce nada sobre la dinámica del sistema a controlar. En vez de eso, solamente las entradas y salidas son accesibles, y por tanto se puede tener acceso a un histórico de datos. El objetivo principal es el control de la planta en condiciones eficientes y seguras usando únicamente dichas medidas. Con este fin, se usará un conjunto de métodos de aprendizaje automático conocido como *kinky inference*, para modelar sistemas no lineales desconocidos, usando controladores predictivos basados en modelo.

Por ello, esta tesis presenta contribuciones en dos campos distintos. En primer lugar, se extenderán las técnicas de kinky inference, proponiendo métodos con dos objetivos: reducir tanto el tiempo de cálculo de los algoritmos como el error de predicción cometido por ellos. En segundo lugar, se desarrollarán controladores predictivos robustos y con la habilidad de aprender basándose en datos. Estos controladores serán estables por diseño, capaces de satisfacer restricciones robustamente y de mejorar su actuación beneficiándose de nuevas medidas recogidas en línea.

Contents

1 Introduction **1**

1.1 Motivation and objectives 1

1.2 Context 4

1.2.1 Identification and machine learning 5

1.2.2 Efficient and safe data-driven control 7

1.3 Data-based models 8

1.3.1 Nonlinear autoregressive models 8

1.3.2 Machine learning 10

1.4 Kinky inference 13

1.5 Model predictive control 17

1.5.1 Learning-based MPC 20

1.6 Main results 22

1.7 Thesis overview 23

1.8 Publications 25

2 Output feedback MPC based on smoothed projected kinky inference **27**

2.1 Introduction 27

2.2 Smoothed projected kinky inference 28

2.2.1 Smoothed kinky inference 28

2.2.2 Projected kinky inference 29

2.3 Data-based predictive control 33

2.3.1 Stability analysis of the controller 35
2.4 Case study 42

3 Robust learning-based MPC for nonlinear constrained systems 54
3.1 Introduction 54
3.2 Problem setting 55
3.3 Stabilizing data-based NMPC 56
3.4 Case study 64

4 Componentwise Hölder inference for robust learning-based MPC 68
4.1 Introduction 68
4.2 Extension of Hölder properties 69
4.2.1 Componentwise Hölder continuity 69
4.2.2 Componentwise Hölder kinky inference 72
4.3 Reachability sets 80
4.4 Stabilizing conditions for CHoKI-MPC 83
4.5 Case studies 84
4.5.1 Quadruple tank process 85
4.5.2 Isothermal chemical reactor 91

5 Online learning constrained MPC based on double prediction 93
5.1 Introduction 93
5.2 Double prediction framework 94
5.3 Online learning-based MPC 95
5.4 Case study 108

6 Online learning robust MPC: an exploration-exploitation approach 113
6.1 Introduction 113

6.2 Exploration-exploitation approach . . . 114
6.2.1 Exploration . . . 114
6.2.2 Exploitation . . . 115
6.3 Online learning controller . . . 117
6.3.1 Stability analysis . . . 119
6.4 Case study . . . 123

7 Conclusions and future work 129
7.1 Summary of contributions . . . 129
7.2 Future work . . . 131

Appendix A Working with data 133
A.1 Input design . . . 133
A.2 Dimensionality reduction . . . 135
A.2.1 Principal component analysis . . . 136
A.3 Data processing . . . 138
A.3.1 Data pruning . . . 139

Appendix B Further introduction to kinky inference 140
B.1 Computing the Hölder parameters . . . 140
B.1.1 Lazily adapted constant kinky inference . . . 140
B.1.2 Parameter optimized kinky inference . . . 141
B.2 Properties . . . 142
B.2.1 Complexity . . . 142
B.2.2 Consistency . . . 143
B.2.3 Bounded uncertainty . . . 143
B.2.4 Learning . . . 144

Appendix C Stability of MPC 147
C.1 Stability of nonlinear systems . . . 147
C.2 Lyapunov stability . . . 150

C.3 Stability of optimal controllers 151
C.4 Stability of MPC 152
C.5 Robust stability 155

List of figures **158**

List of tables **161**

Bibliography **162**

Notation

Vectors

$v \in \mathbb{R}^n$	—	n-dimensional column vector v.
$\lvert v \rvert$	—	Elementwise absolute value.
$\lVert v \rVert$	—	Euclidean norm (unless otherwise indicated).
$\lVert v \rVert_\infty$	—	Infinity norm.
(v, w)	—	Concatenation of vectors. $(v, w) := [v^T, w^T]^T$.
$v \leq w$	—	Componentwise inequality.
$\mathbb{1}_n$	—	n-dimensional column vector of ones.
$\mathbb{0}_n$	—	n-dimensional column vector of zeros.
e_i	—	Vector of zeros with a 1 in the i-th element.
$\tilde{y}$	—	Noisy measurement of y.
$\hat{y}$	—	Predicted value of y.
$\hat{y}(j\lvert k)$	—	Predicted value of y at time step j, given the measurement at time step k.

Matrices

$A \in \mathbb{R}^{m \times n}$	—	m-n-dimensional matrix.
A_i	—	i-th row of the matrix A.
I_n	—	n-dimensional identity matrix.
$\mathbb{1}_{m \times n}$	—	m-n-dimensional matrix of ones.
$\mathbb{0}_{m \times n}$	—	m-n-dimensional matrix of zeros.

Sets

$\mathbb{I}_a^b$	—	Set of integers from a to b.
$\mathbb{N}_0$	—	Set of natural numbers and zero.
$\mathbb{R}^{m \times n}$	—	m-n-dimensional set of real numbers.
$\mathcal{A} \subseteq \mathbb{R}^n$	—	n-dimensional set $\mathcal{A}$.
$\lVert \mathcal{A} \rVert_\infty$	—	Infinity norm of the set $\mathcal{A}$. $\lVert \mathcal{A} \rVert_\infty := \max_{a \in \mathcal{A}} \lVert a \rVert_\infty$.

$\mathcal{A} \oplus \mathcal{B}$	—	Minkowski sum. $\mathcal{A} \oplus \mathcal{B} := \{a + b : a \in \mathcal{A}, b \in \mathcal{B}\}$.
$\mathcal{A} \ominus \mathcal{B}$	—	Pontryagin difference. $\mathcal{A} \ominus \mathcal{B} := \{c : c + b \in \mathcal{A}, \forall b \in \mathcal{B}\}$.
$\mathcal{A} \times \mathcal{B}$	—	Cartesian product. $\mathcal{A} \times \mathcal{B} := \{(a, b) : a \in \mathcal{A}, b \in \mathcal{B}\}$.
$\mathcal{B}(r)$	—	Ball of radius $r \in \mathbb{R}^n$. $\mathcal{B}(r) := \{x : \lvert x\rvert \leq r\} \subseteq \mathbb{R}^n$.
$\mathcal{B}(\mathcal{A})$	—	Cartesian closed topological hull of $\mathcal{A}$.
$\mathbb{B}(r)$	—	Positive box of radius $r \in \mathbb{R}^n$. $\mathbb{B}(r) := \{x : 0 \leq x \leq r\} \subset \mathbb{R}^n$.
$\mathfrak{d}(a, \mathcal{A})$	—	Distance metric from point a to set $\mathcal{A}$.
$\mathrm{Proj}_{\mathcal{A}}(v)$	—	Projection of v onto the set $\mathcal{A}$. $\mathrm{Proj}_{\mathcal{A}}(v) = \arg\min_{a \in \mathcal{A}} \|a - v\|$.

Symbols and functions

$\exp(\cdot)$	—	Exponential. $\exp(x) = e^x$.
∇	—	Gradient.
Δ	—	Increment.
$\wedge$	—	Logical conjunction, *and*.
$\langle \cdot, \cdot \rangle$	—	Scalar product.
k	—	Time step.
$\mathbb{E}(\cdot)$	—	Expected value.
$\mathbb{P}(\cdot)$	—	Probability.
$\mathbb{P}(\cdot, \cdot)$	—	Joint probability.
$\mathbb{P}(\cdot \vert \cdot)$	—	Conditional probability.
x^*	—	Optimal value of x.
x^+	—	Subsequent value of $x(k)$, i.e., $x(k + 1)$.
$\lceil x \rceil$	—	Closest integer greater or equal to x.
$\lfloor x \rfloor$	—	Closest integer lower or equal to x.
$\square$	—	*Quod erat demonstrandum*, end of a proof.

Abbreviations

a.e.	—	Almost everywhere.
cf.	—	*Conferatur*, meaning 'compare'.
e.g.	—	Example given.
GP	—	Gaussian process.
i.e.	—	*Id est*, meaning 'that is'.
iff	—	If and only if.
i.i.d.	—	Independent and identically distributed.
IS(p)S	—	Input-to-state (practical) stability.

KI	—	Kinky inference.
CHoKI	—	Componentwise Hölder KI.
LACKI	—	Lazily adapted constant KI.
PKI	—	Projected KI.
POKI	—	Parameter optimized KI.
SKI	—	Smoothed KI.
MIMO	—	Multiple-input multiple-output (system).
ML	—	Machine learning.
MPC	—	Model predictive control.
EMPC	—	Economic MPC.
LBMPC	—	Learning-based MPC.
NMPC	—	Nonlinear MPC.
NARX	—	Nonlinear autoregressive exogenous (model).
NSM	—	Nonlinear set membership.
PCA	—	Principal component analysis.
o.w.	—	Otherwise.
SISO	—	Single-input single-output (system).
s.t.	—	Such that *or* subject to.
u.o.i.	—	Unless otherwise indicated.
w.o.l.g.	—	Without loss of generality.
w.r.t.	—	With respect to.

Chapter 1

Introduction

1.1 Motivation and objectives

This thesis focuses on the design of advanced control strategies for complex dynamical systems; that is, real-time decision making algorithms devoted to drive the behaviour of the plant according to a given criteria. To this end, knowledge of how a system behaves is of paramount importance: one cannot drive home without knowing that rotating the steering wheel right, makes the car turn right.

The main assumption upon which the thesis stands is that nothing is known from the system to be controlled, only input-output measurements are accessible. This assumption forbids the classic approach of obtaining models of the system based on first principles, that is, describing the physical equations that govern the plant. In opposite, input-output models will have to be derived, leading to a data-based framework.

Many of us who drive daily may not have a full understanding of the functioning of internal combustion engines, however, we were taught that pressing the throttle increased the car speed. Through feedback, that is, observation, we understood the relationship between inputs and outputs of the system, and we were intelligent enough to *learn* the behaviour of the car.

That is precisely the spirit of *data-based* control: developing knowledge of the system based on input and output observations. Humans are indeed excellent data-based learners. In this thesis, knowing a system will mean being able to predict its behaviour based on input-output observations. Therefore, learning nonlinear output-feedback models will our first objective.

The process of learning, when referred to machines instead of humans, reminds of the *artificial intelligence* (AI) field. By "artificial intelligence" we describe all machines able to make decision or to carry out tasks in a "smart" way, like beating chess players or recognizing speech or patterns in images. Among all the available technologies within the AI field, the *machine learning* (ML) sub-field refers to those algorithms that are able to learn (and to improve this knowledge) as the exposure to data increases, without being explicitly programmed to do so. The development of computer capacity has favoured the continuous appearance of more powerful AI and ML applications. In this thesis, ML techniques will be used to learn the behaviour of dynamical systems and to compute predictions in different scenarios.

The second objective of this thesis characterises the control itself: what kind of decision-making process do we want to apply? Once the response of the system to a given input is modelled, a typical closed-loop controller penalises the deviation of the output to the desired reference, so control inputs that steer the system to the given reference are applied. Considering more advanced control techniques, we aim to design controllers that orbit around two crucial aspects: *efficiency* and *safety*. By "efficient", we address the objective of implementing controllers that provide the *best* control actions, according to some performance index that measures what "best" is. With respect to safety, we account for the guarantee that the closed-loop system is not going to violate the security limits, by means of admissible control actions, while asymptotically converging to its reference.

Safety is a key issue in most control applications. For instance, consider the automatic control (autopilot) of an aircraft. This is considered a critical system, in the sense that if something fails, the potential damage is devastating. The same applies to the control systems of drones or large chemical plants. Achieving safety and stability by design of the controlled systems is one of the main challenges of this thesis, due to the black-box nature of the plants to be controlled. In addition, note that because of the dynamical nature of the systems in consideration and their constraints, control algorithms require the decision making process to be solved in real time. For this reason, it is of critical importance to develop computationally efficient implementations of them.

Summing up, the main objective of this thesis is the design of efficient and safe control strategies for constrained nonlinear systems, that only use both historical data and real-time measurements in the decision making process. A schematic representation of the framework is represented in Figure 1.1.

To achieve these objectives, the model predictive control (MPC) frame-

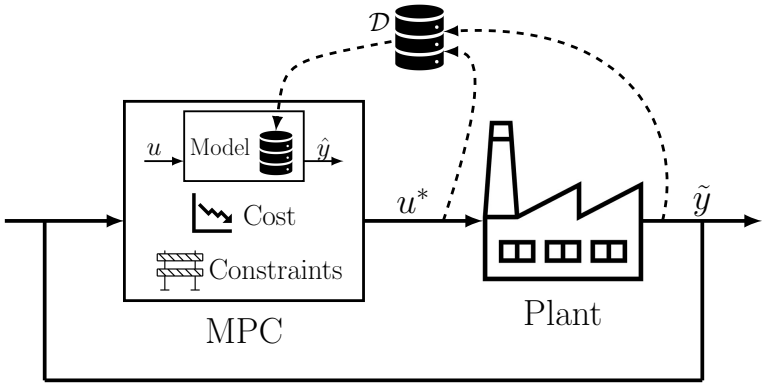


Figure 1.1: Scheme of the overall framework.

work will be used, since it is an advanced control technique that, based on a prediction model, minimizes a performance index in the operation, while being able to guarantee safety by taking the constraints into account. In this thesis, a learning-based approach is taken, in which machine learning techniques that offer accurate predictions will be tailored to identify the unknown dynamics in a computationally tractable form.

It has been proven that in their standard form, predictive controllers can ensure robust stability of the closed-loop if (among other requirements) the model's transition function is Lipschitz continuous and if the uncertainty is bounded with a known worst-case value. Because of this relevant property, the machine learning method considered in this thesis is a class of Lipschitz interpolation techniques known as *kinky inference*, whose resulting predictor is Lipschitz continuous, and it provides guaranteed uncertainty bounds and an accurate estimation of the Lipschitz constant, which will be used for the controller design.

Consequently, this thesis contributes to both the fields of machine learning and model predictive control. Regarding the first, several improvements to the class of kinky inference methods are proposed, aimed to reduce computational times and to improve the quality of the predictions, decreasing their estimation error. Second, new formulations of robust learning-based model predictive controllers are developed, able to ensure stability in the presence of both hard and soft constraints in inputs and outputs, as well as to include new measurements of the plant in an online fashion, thus learning and improving during the operation.

This chapter presents an introductory analysis of the work proposed in

this thesis, its context in the existing literature, its motivation and objectives. Section 1.2 presents the context and the state-of-the-art. The problem setting is addressed in Section 1.3. Section 1.4 introduces the kinky inference technique, and Section 1.5 the model predictive control framework. The main contributions of the thesis are sketched in Section 1.6. Finally, Section 1.7 offers an overview of the contents of each chapter and appendices, and Section 1.8 lists the publications resulting from their elaboration.

1.2 Context

In the light of growing successes of machine learning algorithms, data-based control techniques have become increasingly popular in the control community in recent years. Although the conditions under which a model of the system can be obtained from input-output data have been studied since the late 1980's [41, 76], the recent development of advanced machine learning techniques has favoured the appearance of data-based controllers, using different approaches, like dual [8], adaptive [17], or reinforcement learning [150] controllers.

Another control methodology in which ML finds application consists in deriving data-based models of the system, and to use model predictive controllers based on those, yielding the learning-based MPC field [13]. In contrast to the standard identification approach [86], these new techniques offer a wider class of functions they are able to learn, and have proven to be useful when learning dynamical systems, e.g. [73].

In these cases, a machine learning technique can be used to learn and to predict the evolution of the plant. Among the different techniques applied to learning-based MPC, one of the most popular consists in using a machine learning algorithm to obtain a predictor from input-output data, such as direct weight optimization [137, 136], Gaussian processes [52, 94] or neural networks [159], among many others. Such techniques may handle data in a deterministic or stochastic way, e.g., Lipschitz interpolation [40] for the former and Gaussian processes [2, 18] for the latter. These approaches have been applied in different model predictive controllers [63].

One of the most interesting properties of using ML techniques for control is its natural capability to learn from fresh data, that is, to enhance its predictions by adding new information, like the one collected online from feedback. Control techniques that incorporate this ability are referred to as online learning controllers. While being a popular topic in reinforcement

learning (RL) [78, 115] and adaptive control [1, 153], the matter of online learning still seems to be somewhat under-researched in data-based model predictive control [43, 121]. The different results found in the literature can be classified according to the use they make of new data, like improving an initial feasible solution in an optimization problem [138], designing safety filters [156], or improving the model of the system [62, 87].

In conclusion, there exists many data-driven controllers based on machine learning. However, when the systems in consideration are complex and have constraints, these techniques must be able able to guarantee adequate properties over the closed-loop, like safety of operation and stability. This is currently an open and challenging field, in which this thesis is centered. Further motivation and state-of-the-art of such fields is presented next.

1.2.1 Identification and machine learning

One of the main objectives in this thesis is the study and design of novel data-based methods tailored to real-time decision making processes. Over the last two decades, different techniques such as statistical learning [53], pattern recognition [30], and kernel-based methods [140] have been applied to different fields, like process monitoring [128], fault detection [65] and nonlinear functions approximation [151].

In statistical learning, in order to manage the trade-off between model complexity and performance, regularization (or penalization) parameters to enhance robustness of the results are included in the learning process [58]. Another source of uncertainty in machine learning methods is the lack of sufficient historical data [6]. Techniques to minimize the amount of data required have been studied, using probabilistic methods based on synthetic generation of scenarios [33].

Besides, the machine learning field is not only based on statistics, but also on optimization. This is the case, for example, of random and kernel methods [5]. Specific optimization techniques must be used to handle the resulting high dimensional problems, often with non differentiable objective functions. There exists a manifold of optimization techniques used in machine learning, like accelerated methods, proximal methods, dual decomposition or coordinate descent [147].

With the appropriate data analysis methods, different inference problems in nonlinear systems have been addressed, like state estimation, prediction and reachability analysis. State estimation is a key aspect in many fields

related to dynamical systems. In this context, there exists many strategies in the machine learning field, like parametric [126] or non-parametric methods [29], or multivariate statistical distributions. Recently, seeking to handle uncertainty, nonlinearities and lack of data, new works have satisfactorily applied the tools of Bayesian inference [50], which benefit from the ever increasing computational power available.

This thesis will focus on combining classic identification scopes with data-based inference techniques, aiming to improve the results available in the literature. Existing inference methods will be extended to state estimation, based on the information provide by historical data.

In order to design processes for optimal decision making in real-time, it is necessary to obtain guaranteed predictions of the future behaviour of the plant. The reachability analysis of a dynamical system usually leads to highly complex problems that require different approximation methods, like those based on ellipsoids, zonotopes [3], or interval algebra [28]. One of the main challenges on reachability analysis is the scaling to higher dimensional problems [48], as well as to combine available methods with techniques based on data to bound the reachable sets of the systems in a more precise and efficient way. This problem will be studied in this thesis.

Beyond data-based inference, for an efficient real-time operation in the presence of uncertainty, it is necessary to asses the quality of the models used online, being able to adapt such models and to quantify the uncertainty [2]. Data validation and online model adaptation (both structural and parametric) is a key aspect to guarantee safe operation of a system [138]. Therefore, the derivation of data-based methods able to learn and to adapt themselves online, recursively, as more data become available during the operation will also be studied in this thesis.

Finally, it is worth mentioning methods such as data completion [152], that have been applied to various fields and which use stored data to find connecting patterns and to recover lost information. Similarly, data-based virtual sensors [148] usually offer a method to obtain estimated values of physical variables that cannot be measured by a physical sensor. All these technologies motivate the study of black-box systems carried out in this thesis.

1.2.2 Efficient and safe data-driven control

Efficiency is a term that measures the behaviour of a given system, taking into account, for example, energy consumption, use of actuators, evolution according to a prescribed setting, etc. This term can encompass a collection of measurements, and in a control engineering terminology, this is comprised in a general function named *performance index* or *economic cost*.

The controller must choose the action that each time step provides the least economic cost, associated to the dynamical evolution of the system [44]. The decision-making process must also ensure that the system, during its evolution, guarantees the imposed operation limits, derived for example from environmental conditions or physical constraints [60]. This problem becomes much more complex if the inherent uncertainty of the unknown plant, as well as the modelling errors due to the learning methods applied are taken into account [82].

Designing control laws to optimize the economic cost of the closed-loop operation of the plant is challenging, due to the general metric used to measure efficiency, which does not necessarily penalize the tracking error w.r.t. a set-point, as it is customary in classic control approaches. Recently, model predictive control techniques have been proposed to deal with this problem, leading to the so-called economic MPC. Following this line, stable designs of economic predictive controllers for large processes have been proposed in the last decade [60]. One of the most important properties of model predictive control [38] is that the resulting controller may, by design, guarantee stability of the closed-loop system, which together with the guarantee of constraints satisfaction, yields safe operation of the system.

Model predictive control has a strong dependence on the quality of the predictions. In order to improve closed-loop performance, predictive controllers able to adapt their models online have been developed, both linear [13] and nonlinear [40]. Currently, there is ongoing research on this trending topic, as shown for example in the excellent survey paper by Hewing et al. [63]. In contrast to the majority of approaches presented in that review, the work proposed in this thesis will address the problem from a deterministic point of view, instead of stochastically, in order to give ensured guarantees of robust performance, which is a crucial aspect in critical systems.

1.3 Data-based models

This section explains how a regressive model of a system can be derived using only input-output data. Then, the machine learning field is introduced to provide a framework able to learn such regressive models.

1.3.1 Nonlinear autoregressive models

Consider a nonlinear, discrete time, output-feedback, multiple inputs multiple outputs (MIMO) dynamical system in which $y(k) \in \mathbb{R}^{n_y}$ is the vector of outputs, $x(k) \in \mathbb{R}^{n_x}$ the state, and $u(k) \in \mathbb{R}^{n_u}$ the manipulable inputs at time step $k \in \mathbb{N}_0$.

Note that by *system*, one typically refers to the real plant to be controlled. However, it is often not possible to have a perfect knowledge of the real system behaviour, so an approximated mathematical model is used instead. In order to design a controller, the first objective of every control engineer is to obtain such model of the system, to simulate the plant's behaviour, and thus taking into account the possible response of the closed-loop system in the design. A model is even more important in predictive control strategies, where the model is explicitly used by the controller, in order to predict future evolution of the plant.

A first attempt to construct a model may be based on first principles. This approach consists in describing the set of physical, ordinary differential equations that govern the plant. However, as it was stated before, in this thesis we assume that the inner signals of the system are not accessible by the user, and only input-output samples are available. For this reason, a regressive form of the state must be constructed in order to describe the system effectively.

For the case of linear dynamics, autoregressive exogenous (ARX) models provide an appropriate way to accurately describe the behaviour from input-output information, such that

$$y(k+1) + a_1 y(k) + \ldots + a_{n_a} y(k-n_a) = b_1 u(k) + \ldots + b_{n_b} u(k-n_b). \quad (1.1)$$

Here, the model is said to be *autoregressive* since $y(k+1)$ depends on previous outputs, up to the memory horizon n_a, and since it depends on the external previous inputs $u(k-1)$ up to $u(k-n_b)$, it is said to be *exogenous*.

The extension to nonlinear systems is not straightforward (see [67]). An overview of nonlinear black-box identification methods by the end of last

century was done by Sjöberg et al. in [141]. Nonlinear ARX models were introduced in the late 1980's [41, 76], such that

$$y(k+1) = f\big(y(k), \dots, y(k-n_a), u(k), \dots, u(k-n_b)\big). \tag{1.2}$$

Previous outputs and inputs have an effect on the following output, so they can be grouped in the state

$$x(k) = (y(k), \dots, y(k-n_a), u(k-1), \dots, u(k-n_b)), \tag{1.3}$$

where $n_a \in \mathbb{N}_0$ represents the memory horizon for the outputs, and $n_b \in \mathbb{N}_0$ the memory horizon for the inputs. Although the state is fully defined for $n_b \geq 1$, with a slight abuse of notation $n_b = 0$ indicates that no past term of the inputs appears in the state. Note that $x \in \mathbb{R}^{n_x}$, with $n_x = (n_a+1)n_y + n_b n_u$. Taking into account (1.3), (1.2) can be rewritten as

$$y(k+1) = f(x(k), u(k)). \tag{1.4}$$

The conditions under which a system can be described as a NARX model were presented in [105]. These conditions have been widely studied during the last 40 years. One of the first results on this topic was given by Sontag [144], relating the existence of this model to the observability property of the system. Later, Chen and Billings [41] proved that local NARX models can be obtained if the system is locally observable. A comprehensive study of this problem, for local and global estimators, was presented by Levin and Narendra [77]. Further extensions can be found in [158], where global input-output models are proven valid via transversality theory.

In [77, Theorem 3] it was proven that if a linearised model at an equilibrium point (x_s, u_s) is observable, then the dynamics of the system can be locally described by a NARX model. When a global NARX model is required, the strong observability property must be ensured, from the linearised system, for instance. However, in [77, Theorem 6] the authors proved that only generic observability is necessary, which is a property that most systems enjoy in practice. From this result, the conditions required to ensure the existence of this model for a sampled continuous-time system can be derived:

Theorem 1.1. *Consider that each component of the system function, f_i, with $i \in \mathbb{I}_1^{n_y}$, is continuous and such that for all $\tilde{x}$ where the gradient is null, i.e., $\nabla_x f_i(\tilde{x}, u) = 0$, the Hessian matrix $\nabla_{xx} f_i(\tilde{x}, u)$ is non-singular. Then, a NARX model can be determined as valid for almost every input sequence. Besides, the model function $\hat{f}_i$ is continuous and the horizons can be taken as $n_a = n_b = 2n$, where n is the minimum number of states of the system.*

Note that nothing is mentioned here regarding the *quality* of the input-output data available to derive the model. This fact has a crucial effect on the determination of the NARX model function using, for example, ML techniques, and it is analysed in Appendix A. Methods to study the model order and structure from input-output data are presented in [26], [56] and [59].

Based on the available input-output data, and assuming that they are informative enough, the structure of the NARX model and its (hyper-)parameters must be determined. The field of machine learning offers a wide range and well-established methods to cope with this problem, as it will be summarized next.

1.3.2 Machine learning

Machine learning [113] is a whole area of computer science, often said to be a subset of artificial intelligence. The objective of machine learning is to develop algorithms that automatically improve through experience. Here, *automatically* means without being explicitly programmed to do so; and *to improve* means that they should be able to perform better according to some metric/objective. Typically, this objective translates into being able to predict function values of given queries, from functions that are unknown a priori, and *performance* implies accuracy in the predictions. This thesis will be centered in the kinky inference method, which is a supervised regression technique. The static function to be learnt is defined as follows.

Definition 1.1 (Ground truth function)**.** *Given two Hilbert spaces, $\mathcal{W} \subseteq \mathbb{R}^{n_w}$ and $\mathcal{Y} \subseteq \mathbb{R}^{n_y}$, referred to as input and output space, respectively, the ground truth function is defined as the mapping $f : \mathcal{W} \to \mathcal{Y}$ such that*

$$y = f(w). \tag{1.5}$$

Unless otherwise indicated, we will consider Euclidean metrics for both spaces, although J. Calliess proposed to use pseudo-metrics in the general kinky inference method [34], in order to account, e.g., for periodic signals.

As it was stated before, supervised regression algorithms are going to be considered. Hence, it is assumed that a set of $N_{\mathcal{D}}$ input-output observations of the ground truth function is available. They may be also be referred to as *samples*, *measurements*, or *data points*. Note that in a real scenario, the measurements of the output are subject to possible noise $\varepsilon(k) \in \mathbb{R}^{n_y}$, and hence the noise-corrupted observation is denoted $\tilde{y}$, such that

$$\tilde{y}(k+1) = f(w(k)) + \varepsilon(k). \tag{1.6}$$

Then, the observations are gathered in the data set

$$\mathcal{D} = \{(\tilde{y}_i, w_i), i \in \mathbb{I}_1^{N_\mathcal{D}}\}. \tag{1.7}$$

The data set containing only inputs will be denoted $\mathcal{W}_\mathcal{D}$. From now onwards, it is assumed that the possible noise is bounded.

Assumption 1.1. *The output noise $\varepsilon \in \mathbb{R}^{n_y}$ is upper-bounded by some constant $\bar{\mathfrak{e}} \in \mathbb{R}^{n_y}$, that is,*

$$|\varepsilon| \leq \bar{\mathfrak{e}}. \tag{1.8}$$

In other words, the noise lies in a set defined as $\mathcal{E} \subset \mathbb{R}^{n_y} = \{\varepsilon : |\varepsilon| \leq \bar{\mathfrak{e}}\}$, that is, $\varepsilon \in \mathcal{E} = \mathcal{B}(\bar{\mathfrak{e}})$.

This is considered a sensible assumption and besides, notice that the discrete nature of every real system forbids the measurements to become infinite. In addition, the research in this thesis will often be applied to compact spaces, where every signal is bounded.

Then, a machine learning algorithm will be used to build the predictor

$$\hat{y} = \hat{f}(w; \theta, \mathcal{D}), \tag{1.9}$$

where $\hat{y}$ stands for predicted outputs and $\theta \in \mathbb{R}^{n_\theta}$ stands for the set of (hyper-)parameters of the predictor. This predictor may use the set of training data $\mathcal{D}$ to make predictions, or only use it during the learning (tuning) process of θ.

This thesis will frequently make use of the term *prediction error*, which is defined as follows:

Definition 1.2 (Prediction error)**.** *Given the ground truth function f and a predictor $\hat{f}$ for this function, the prediction error $d \in \mathbb{R}^{n_y}$ for a given query w is computed as $d(w) = |f(w) - \hat{f}(w; \theta, \mathcal{D})|$.*

Besides, we will denote the maximum prediction error as follows:

Definition 1.3 (Worst-case prediction error)**.** *The maximum prediction error using the predictor $\hat{f}$, given the parameters θ and data set $\mathcal{D}$ is denoted μ, such that*

$$|\hat{f}(q; \theta, \mathcal{D}) - f(q)| = d(q) \leq \mu \in \mathbb{R}^{n_y},\ \forall q \in \mathcal{W}. \tag{1.10}$$

Then, the last step in order to fully identify a dynamical NARX model consists in fixing a structure and tuning the model's parameters accordingly. Hence, the dynamical system is transformed into a static function, suitable to be processed by machine learning algorithms, via regression. To this end, we define the *regressor* as follows:

Definition 1.4 (Regressor)**.** *The state x and the control input u are aggregated into the so-called* regressor $w \in \mathbb{R}^{n_w}$*, that is, given the definition of the state x in (1.3),*

$$w(k) = \big(x(k), u(k)\big) = \big(y(k), \dots, y(k-n_a), u(k-1), \dots, u(k-n_b), u(k)\big). \tag{1.11}$$

Therefore, a NARX model of a dynamical system, defined by (1.2), can be rewritten as

$$y(k+1) = f\big(w(k)\big). \tag{1.12}$$

With this transformation of a dynamical system into a static function, a prediction model can be learnt as per (1.9), using any machine learning technique; for example, linear regression [116]. This is usually an iterative task in the identification field. For instance, different data sets are used: while $\mathcal{D}$ is used for predicting, others are used for validating the predictor, denoted $\mathcal{D}_{\text{test}}$. The tuning process of θ (see (1.9)) typically aims to minimize the prediction error over $\mathcal{D}_{\text{test}}$. Consider the following example to sum up the identification procedure of regression models:

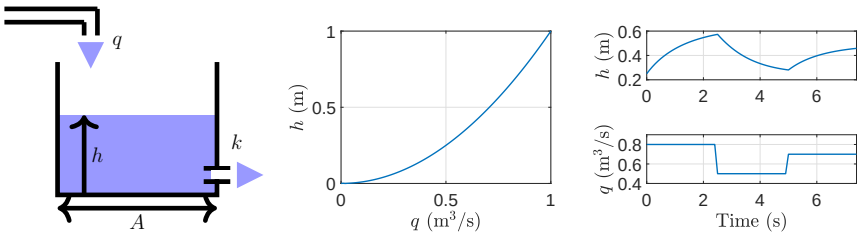
Example 1.1. *Consider a tank in which the liquid height h, measured in* m*, evolves according to the input flow q ($\mathrm{m}^3\,\mathrm{s}^{-1}$) and the discharge flow, following the equation*

$$A\frac{dh(t)}{dt} = q(t) - k\sqrt{h(t)}, \tag{1.13}$$

where $A = 1\,\mathrm{m}^2$ is its area and $k = 1$ the discharge coefficient. A scheme of the system is represented in Figure 1.2a. Note that the square root makes it nonlinear, as shown in the set of equilibrium points in Figure 1.2b. The integration time is fixed to $0.1\,\mathrm{s}$*.*

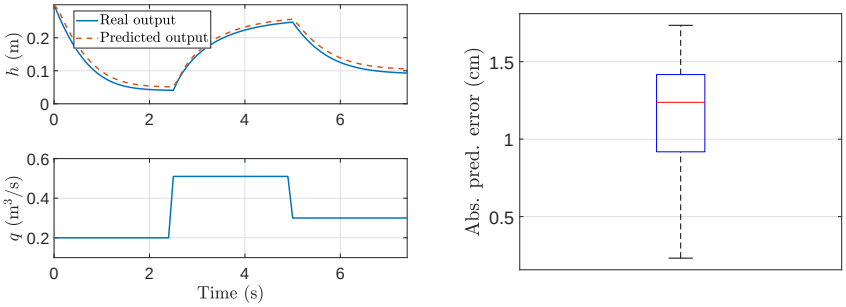
Using a data set of $N_{\mathcal{D}} = 75$ points gathered from the input-output trajectory shown in Figure 1.2c, a linear regression model is built with $n_a = n_b = 0$, and tuned using least squares, yielding $\theta = [-0.031,\ 0.92,\ 0.1]$. The closed-loop prediction for a new input sequence is shown in Figure 1.3, as well as a boxplot of the prediction error.

Once the prediction model is obtained, it can be used in a predictive controller to forecast future evolution of the plant, in order to choose the control action that minimizes a given cost. However, if a robust design of this controller is desired, one needs to take the uncertainty into account, so information of the possible prediction error is required.



(a) Tank scheme (b) Equilibrium points (c) Training data

Figure 1.2: Simulation of a tank.



(a) Closed-loop response (b) Absolute prediction error

Figure 1.3: Closed-loop performance of a NARX model identified via least squares.

Gaussian processes [131] and kinky inference [34] are ML methods that provide an estimation of the worst-case prediction error, the former in a stochastic description and the latter in a deterministic one. The following section introduces the the kinky inference class, together with the properties that justify its choice as learning method in this thesis.

1.4 Kinky inference

This section presents the machine learning technique used in the thesis. The reasons why it is of special interest for the design of robust deterministic predictive controllers will be analysed, as well as its main drawbacks and the methods that are proposed to tackle them. This section is an introductory analysis to the method in its standard form. Besides, the existing methods to

learn the hyper-parameters as well as the properties of the resulting predictor are presented in Appendix B. Then, the proposed modifications to improve the class of learning rules, which are a contribution of this thesis, will be presented in following chapters.

The term *kinky inference* (KI) was coined by Jan-Peter Calliess in [34], making reference to the *kinks* (that is, sudden changes in the first derivative) that the resulting predictions exhibit. There, Calliess laid the foundations of the class of learning rules under the name of kinky inference, and the reader is referred to it for further analysis of its derivation and applications.

Kinky inference evolves from Lipschitz interpolation, as it will be analysed below. Lipschitz interpolation techniques were first described, as far as the author of this thesis is concerned, by A. G. Sukharev in [149]. Further relevant work on Lipschitz interpolation is presented by G. Beliakov in [15]. In addition, the method has also been referred to as nonlinear set membership (NSM) identification, as derived by M. Milanese and C. Novara [112].

The learning method of Lipschitz interpolation is based on Lipschitz continuity of the ground truth function. This property, named after the mathematician R. Lipschitz [108], describes the functions that satisfy the following condition.

Definition 1.5 (Lipschitz continuity)**.** *A function $f : \mathcal{W} \to \mathcal{Y}$ is Lipschitz continuous if there exist a real constant $L_f \geq 0$ such that for all $w_1, w_2 \in \mathcal{W}$,*

$$\|f(w_1) - f(w_2)\| \leq L_f \|w_1 - w_2\|. \tag{1.14}$$

Remark 1.1. *Note that if the condition is satisfied for L_f, it will also be satisfied for any $L' > L_f$. Some authors call every L' a Lipschitz constant, and refer to L_f (the smallest of them) as the* best *Lipschitz constant. For simplicity, we will just refer to L_f as the Lipschitz constant.*

Remark 1.2. *Note that the Lipschitz property is more powerful than the definition above, if other metrics were considered. For simplicity, this thesis will stick to the Euclidean metric, u.o.i., but the framework holds with an easy switch of metric spaces, provided the equivalence of the norms. The extension to consider input pseudo-metrics was analysed in [34].*

If $L < 1$ the mapping f is said to be a contraction [142]. Notice that every continuously differentiable function is Lipschitz, but not vice versa, e.g. $f(w) = |w|$. Note also that, for example, $f(w) = \sqrt{w}$ is not Lipschitz in the origin. However, there exists an extension of the Lipschitz continuity, named after the mathematician O. Hölder:

Definition 1.6 (Hölder continuity)**.** *A function $f : \mathcal{W} \to \mathcal{Y}$ is Hölder continuous if there exist two real constants $L_f \geq 0$ and $0 < p \leq 1$ such that for all $w_1, w_2 \in \mathcal{W}$,*

$$\|f(w_1) - f(w_2)\| \leq L_f \|w_1 - w_2\|^p. \tag{1.15}$$

Here, p (which is also often denoted α in the literature) is called the Hölder exponent. Note that if $p = 1$ it implies Lipschitz continuity. If the condition holds for a given p, the function is said to be p-Hölder continuous.

In contrast with Lipschitz interpolation o NSM, the kinky inference class extends the learning method to Hölder functions. It requires the following assumption.

Assumption 1.2. *The ground truth function f is p-Hölder continuous, with Hölder constant L_f.*

Note that this is not considered a limiting assumption, since the application of the kinky inference method to control will be done mainly in compact sets, and in general, piecewise differentiable functions are Hölder in a compact set (consider for example $f(w) = w^2$).

The value of the Hölder constant of the ground truth, L_f, is unknown. While in several works, a priori knowledge of the correct Hölder constant is assumed [40, 160], other works have proposed methods of adapting these parameters to the data [36, 37, 112]. In this thesis, following the assumption that nothing is known from the ground truth (apart from Hölder continuity), an estimation method independent of L_f has be to derived, which is shown in Appendix B.

Having in mind the learning framework presented in Section 1.3.2, the standard kinky inference method yields the predictor sought in (1.9) as follows:

Definition 1.7 (Standard KI predictor)**.** *Given a data set of $N_{\mathcal{D}}$ input-output observations, $\mathcal{D} = \{(\tilde{y}_i, w_i), i \in \mathbb{I}_1^{N_{\mathcal{D}}}\}$, and certain Hölder parameters grouped in $\theta = (L, p)$, the prediction of an unseen query point q is computed as:*

$$\hat{\mathfrak{f}}(q; \theta, \mathcal{D}) = \frac{1}{2} \underbrace{\min_{i \in \mathbb{I}_1^{N_{\mathcal{D}}}} (\tilde{y}_i + L\|w_i - q\|^p)}_{\mathfrak{u}(q;\theta,\mathcal{D})} + \frac{1}{2} \underbrace{\max_{i \in \mathbb{I}_1^{N_{\mathcal{D}}}} (\tilde{y}_i - L\|w_i - q\|^p)}_{\mathfrak{l}(q;\theta,\mathcal{D})}. \tag{1.16}$$

Analysing (1.16) gives an insight of the simple procedure of the KI algorithm. The fact that the ground truth is Hölder (or Lipschitz) continuous

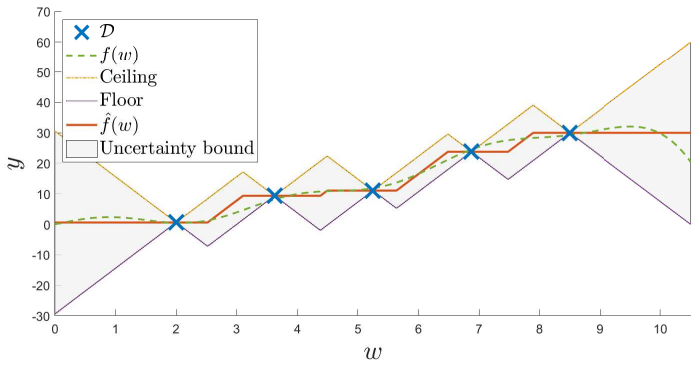


Figure 1.4: Illustration of the kinky inference algorithm.

provides a bound on the domain where the function is going to lie, at the given query q. This bound is given by the *ceiling* $\mathfrak{u}$ and *floor* $\mathfrak{l}$ functions, computed from every observed data point applying the Hölder inequality. Then, the method is as simple as sticking to the mean value between the minimum ceiling and the maximum floor.

Example 1.2. *Consider the (unknown) function*

$$f(w) = 2\sin(2w) + \frac{w^2}{2} - \frac{\exp(w)}{1000},$$

with random noise following a normal distribution of mean 0 *and standard deviation of* 1*. Consider a data set of* 5 *measurements.*

The resulting prediction is shown in Figure 1.4, using the kinky inference algorithm with $L = 15$ *and* $p = 1$*. Note that each data point in* $\mathcal{D}$ *yields a ceiling and a floor over* $\mathcal{W}$*, according to the bounded slope* L*. Then, KI computes the minimum ceiling and the maximum floor, obtaining the prediction as the mean value between them, for every* $q \in \mathcal{W}$*.*

In Appendix B it is shown that the resulting predictor is Hölder continuous and that the uncertainty is bounded. Some other benefits of KI for control are its learning and predicting simplicity, its numerical stability and its parallel and online configurability, among others. On the other hand, there are also some drawbacks, including its conservatism, nondifferentiabilities and its high computational cost, common to most non-parametric methods. In this thesis, new KI formulations are developed to overcome these issues.

1.5 Model predictive control

This section introduces the reader to the model predictive control (MPC) field, with a brief overview of its origins and motivation, as well as the current state-of-the-art.

It is usually convenient to formulate MPC problems in state-space prediction models, because it is a general formulation that simplifies the stability analysis. It was already stated that NARX models can be formulated in state-space, such that

$$x(k+1) = F(x(k), u(k)). \tag{1.17}$$

Model predictive control is a technique that arose to give practical and implementable solution to optimal controllers, also known as infinite horizon optimal control problems. The control action computed depends on the current state of the plant, yielding the control law $u(k) = \kappa\big(x(k)\big)$.

Recall that the objective is to steer the system to the reference, denoted x^{ref}. For notational simplicity, we will assume that the origin, that is, $x = \mathbb{0}_{n_x}$, is the reference to where the system must be driven. This is just a notational simplification, since the problem can be rewritten, w.l.o.g., by a change of variables $\tilde{x}(t) = x(t) - x^{\text{ref}}$. Besides, it is assumed that the origin is an equilibrium point of the system, that is,

$$0 = f(0, 0). \tag{1.18}$$

Optimal control problems (OCP) [120] minimize a cost of driving the system to the reference during its operation. This cost is defined as [1]

$$V_\infty(x(k)) = \sum_{j=1}^{\infty} \ell\left(x(j|k), \kappa_\infty\big(x(j|k)\big)\right), \tag{1.19}$$

where $x(j|k)$ denotes the prediction of the state at time step j, given the measurement of $x(k)$, and $\ell(x, u)$ is called the *stage cost*, and it is the static function that measures the efficiency or economic cost of the system at the current state $x(k)$, for a control action $u(k)$. The optimal control law is denoted $\kappa_\infty(\cdot)$. Hence, an optimization problem has to be solved, in contrast

[1] The cost in optimal control problems or in model predictive control frameworks is sometimes denoted J in the literature (see, e.g. [100]). We will denote it V, in relation with Lypaunov functions.

to more basic approaches, where an explicit solution can be computed (like PID [12]).

In addition, the plant may be subject to operation constraints, restricting both the control actions to $u(k) \in \mathcal{U}$ and the states to $x(k) \in \mathcal{X}$, for all k. The resulting optimization problem in optimal control schemes is denoted $P_\infty(x(k))$:

$$\begin{aligned} \min_{\kappa_\infty(x)} \quad & V_\infty(x(k)) && \text{(1.20a)} \\ \text{s.t.} \quad & x(j|k) = F(x(j|k), \kappa_\infty(x(j|k))), \quad j \in \mathbb{I}_0^\infty, && \text{(1.20b)} \\ & \kappa_\infty(x(j|k)) \in \mathcal{U}, \quad j \in \mathbb{I}_0^\infty, && \text{(1.20c)} \\ & x(j|k) \in \mathcal{X}, \quad j \in \mathbb{I}_0^\infty. && \text{(1.20d)} \end{aligned}$$

This yields the best possible controller, considering the stage cost, and it guarantees stability of the closed-loop system and constraints satisfaction for every initial state for which (1.20) is feasible, as proven in Appendix C.

However, the solution to this problem may be impossible to compute, given the potential infinite number of decision variables. Note that equation (1.20b) implies a differential constraint defined by the systems' transition function, so it leads to a dynamical optimization problem, which is known as a variational problem. The solution to this optimization problem is known for especial cases, such as unconstrained linear systems with quadratic cost functions (which leads to the so-called linear quadratic regulator (LQR) [31]). In general, this variational problem can be solved by determining the solution of certain set of equations derived from the Hamilton-Jacobi-Bellman equations [16] or from Pontryagin's minimum principle [127], which are very difficult to solved.

This impossibility to solve optimal control problems led to model-based predictive controllers (MPC), which are based on two premises. First, a fix, *finite prediction horizon* is used, typically denoted N, allowing numerical solving of the OCP. Second, a *receding horizon* policy is used, such that given the obtained control sequence of N terms

$$\mathbf{u}(k) = (u(k), u(k+1), \ldots, u(k+N)), \tag{1.21}$$

only the first one is applied to the system, yielding a new state and computing the problem again every time step. This receding horizon technique provides feedback and continuous updating of the cost. The standard MPC problem,

denoted $P_N(x(k))$, is the following:

$$\min_{\mathbf{u}(k)} \quad V_N(x(k), \mathbf{u}(k)) \tag{1.22a}$$

$$\text{s.t.} \quad x(j|k) = F(x(j|k), u(k+j)), \quad j \in \mathbb{I}_0^{N-1}, \tag{1.22b}$$

$$u(k+j) \in \mathcal{U}, \quad j \in \mathbb{I}_0^{N-1}, \tag{1.22c}$$

$$x(j|k) \in \mathcal{X}, \quad j \in \mathbb{I}_0^{N-1}, \tag{1.22d}$$

$$x(N|k) \in \mathcal{X}_f, \tag{1.22e}$$

where the optimization cost is

$$V_N(x(k), u(k)) = \sum_{j=0}^{N-1} \ell(x(j|k), u(k+j)) + V_f(x(N|k)), \tag{1.23}$$

which includes a *terminal cost* $V_f(\cdot)$. Besides, the *terminal constraint* (1.22e) is considered, forcing the system to be at the terminal region $\mathcal{X}_f$ at time step $k+N$.

MPC has had an enormous success in industry applications [129], given its optimal nature and its ability to manage constraints. However, taking a finite horizon may loose the well-established properties of optimal controllers. For this reason, stability of MPC has been thoroughly studied since the 1980's. Authors that have laid the foundations of model predictive control are D.Q. Mayne and J.B. Rawlings [133], E.F. Camacho and C. Bordons [38], M. Morari [117], and J. Maciejowski [92], among others. The extension to nonlinear MPC has been studied, together with the previous authors, by F. Allgöwer and R. Findeisen [7], L. Grüne [55] or D. Limon [79], among others. Mayne et al. [107] proved that both stability and constraints satisfaction can be ensured adding an appropriate terminal cost and terminal constraint, as shown in Appendix C. Recently, it has been proven that an MPC without terminal conditions can be stable for large memory horizons [55].

In contrast to more basic control strategies, the method requires an accurate prediction model of the system and to solve an optimization problem each sampling time, so it demands higher computational capacity and an advanced knowledge of the optimization field. Some optimization techniques that have been studied for solving the MPC formulation have been proposed by D.P. Bertsekas [20], L.T. Biegler [22], S. Boyd [27] or M. Diehl [64].

Several sub-fields in MPC have also been of interest to the research community, like *distributed* MPC, which has been studied in depth by E. Camponogara [39], P.D. Christofides, R. Scattolini and D. Muñoz de la Peña [42], among others; or *economic* MPC, which can be found in the

works by M. Diehl [44], R. Amrit [10], D. Angeli [11, 132], M. Müller [118] , T. Faulwasser [47] or S. Lucia [89]. The problem of *tracking* MPC, for changing references has been studied by D. Limon, A. Ferramosca, A.H. González or M. Müller [84, 85].

Further literature review, together with an introduction to the stability and robustness analysis of model predictive control is presented in Appendix C.

1.5.1 Learning-based MPC

As it was introduced in Section 1.2, recently, machine learning techniques are being used in the control field to enhance the performance of the resulting closed-loop. Many control methodologies that use data in their processes have gained attention in the last few years. Among others, it is worth mentioning *adaptive control*, for which the data-driven approach was recently reviewed by M. Benosman in [17]. Some adaptive controllers that merge data and MPC techniques can be found in the works by Adetola et al. [1] or Tanaskovic et al. [153]; the former for systems linear in their parameters and the latter for linear time-varying systems.

Likewise, *dual control* seeks to identify and to control the plant at the same time. The derivation of data-based approaches for dual control can be found in the recent survey by A. Mesbah [110]. Apart from that, *reinforcement learning* is another useful technique that benefits from the experience through data observation. As we will discuss below, the learning-based approach usually enhances the performance of the controllers, although few results address the safety and stability analysis. Regarding reinforcement learning, Berkenkamp et al [19] recently proposed a safe approach with stability guarantees, able to improve the performance and to explore the safe region of the state-space, under some assumptions. Gros & Zanon have recently proposed a RL scope tailored to economic MPCs [54].

Other techniques do not require a model of the system, for instance, the *direct* approach is a data-driven technique in which a set of past trajectories is stored, so the control law is learnt, as shown in [125]. Later, Salvador et al. [137] proposed an offset-free controller based on these historian databases. Among other advantages, there is no tuning step, and new data may easily be incorporated into the process.

Identification via machine learning provides tools that favour the determination and improvement of prediction models. Model predictive control

provides the methodologies and theoretical support for the practical implementation of optimal control strategies. Therefore, apart from the techniques mentioned above, the encounter of both fields has recently gained increasing attention in the research community. MPC can benefit from the use of machine learning algorithms, since data-driven techniques have proven useful to various purposes, like learning (or improving) the prediction model or learning the control laws, as reviewed in [63]. Learning the system model versus learning the control law does not necessarily have to be opposite approaches. Piga et al. recently proposed a method that takes into account the control objective in the model learning step [124]. In the last few years, machine learning techniques have proven valid to describe wide ranges of classes. Consider for example the autonomous driving applications that are currently being developed by the groups of F. Borrelli [135], S. Di Cairano [43] or M. Zeilinger [70], among others.

One of the most challenging issues of using learning-based MPC controllers is how to explicitly recover the safe-by-design properties of nominal formulations. Most off-the-shelf results do not provide stabilizing designs or do not consider modelling errors or constraints, forbidding the deployment of a stability analysis of the closed-loop system. The inherent uncertainty in the ML-based prediction model brings one of the most challenging problems in learning-based control: safety. When the system to be controlled is subject to limits in the manipulable inputs or in the measurable outputs, the resulting controller must make a decision coping with this uncertainty in order to provide a safe evolution of the system. This requires the ML method to provide a description of the uncertainty or prediction error between the real evolution of the plant and the estimated one. Within this topic, there are two approaches that arise when working with data-based schemes: deterministic and stochastic. Then, many applications could benefit from the stability analysis, as for example the recent works on robotic systems by Fisac et al. [52] using the former, or Ostafew et al. [122] using the latter.

Dealing with the uncertainty following a stochastic approach is an open problem in the MPC community. During the last few years several research groups have proposed excellent techniques to address this problem, either studying safe control methodologies (see for example [88]) or using identification methods that provide a probabilistic distribution on the uncertainty of the predictions, like Gaussian processes regression [131]. The estimation on the prediction error offers a probabilistic bound, used to derive safe controllers, typically according to a given confidence interval. For example, the group of M. Zeilinger uses Gaussian processes to design stochastic robust bounds for MPC, via designing safety filters [156] or cautious controllers [62].

Other couple of research lines that consider the stability analysis of Gaussian process-based MPCs are those recently proposed by Koller et al. [74] and Maiworm et al. [93]. Besides, both controllers are able to adapt themselves, enhancing the performance by taking into account new measurements.

On the other hand, the problem can be addressed from a deterministic approach. As it was mentioned before, the safe and robust design of controllers requires bounded errors, but there are few methods that guarantee an ensured bound on the uncertainty. Lipschitz interpolation methods [15] provides a bound on the prediction error, as the robust design requires. They are also referred to as nonlinear set membership (NSM), used by Canale et al. to derive nonlinear data-based MPCs [40]. Lorenzen et al. have recently proposed an online framework that ensures stability of linear systems with parametric uncertainty and additive disturbance [87]. Besides, Terzi et al. also proposed recently a general framework for state-estimation using NSM [154], for linear systems as well. Another robust predictive controller for linear systems was recently proposed by Soloperto et al [143], considering state-dependent uncertainties. Lucia & Karg presented in [90] the power of deep-learning approaches to approximate robust NMPC laws.

Summing up, on the one hand, stochastic approaches yield less rigorous analysis of the stability results [109], so deterministic scopes are needed when ensured robust guarantees are sought. On the other hand, there are few results that address the problem of controlling nonlinear black-box systems while taking into account the uncertainty, in order to consider a robust stability scenario. This thesis aims to contribute to this topic, offering the results that are sketched in the following section.

1.6 Main results

This section presents the main contributions of this thesis, whose structure can be found in the following section. As it was stated before, these contributions belong to two fields: machine learning and model predictive control.

- An improved kinky inference algorithm, named *projected* KI, has been developed in order to reduce computational times of the prediction method.
- A new approach on Hölder continuity of functions has been proposed, which explicitly takes into account the contribution of each input onto each output, named *componentwise* Hölder continuity.

- Based on the latter, an novel KI predictor has been proposed: the componentwise Hölder kinky inference (CHoKI) method, which significantly reduces estimation errors.
- A new method to estimate the Hölder constant from the available data has been proposed. It is an hybrid method that merges the two most commonly used methods: LACKI and POKI.
- The online version of the KI algorithms has been studied. Several policies to update the data set recursively have been proposed.
- Two new control laws have been derived for learning-based model predictive controllers subject to uncertainty: a controller for systems with constraint on the inputs and its extension to systems with constraints also in the outputs. These novel control laws do not require a terminal constraint, avoiding the need for the calculation of any kind of invariant set, which is cumbersome in learning-based control. The procedure and analysis to achieve stability by design are provided.
- Based on the CHoKI predictor, predictive controllers with larger domain of attractions have been presented, benefiting from two facts: lower and more accurate uncertainty bounds and larger feasibility regions.
- An online learning MPC based on two different prediction models has been proposed. A safe model guarantees robust stability, while an updated model improves the performance by means of including new measurements during the operation.
- An online learning MPC for tracking based on an exploration-exploitation criteria has been proposed. Even in the case of low-dense data sets, the closed-loop system is able to explore unseen regions of the space, driving the system to the best reachable reference while robustly satisfying constraints.

1.7 Thesis overview

The rest of the thesis is structured as follows:

Chapter 2 proposes a robust predictive controller that uses a model derived with the class of kinky inference. The controller formulation considers

soft constraints in the outputs, hard constraints in the inputs and guarantees closed-loop robust stability as well as performance by means of the use of different control and prediction horizons and a weighted terminal cost, without the need of a terminal region. With respect to the learning method, a modification of the standard KI algorithm is proposed: the projected KI, which aims to reduce computational times, in order to allow the application to real-time systems.

Chapter 3 addresses the problem of robust and stable control of systems subject to hard constraints in the outputs. A learning-based MPC and a design procedure to guarantee that the closed-loop system satisfies the constraints while steering the system to the reference without the calculation of a terminal invariant set are proposed. This is one of the key contributions of the thesis.

Chapter 4 presents an improvement of kinky inference, which is able to explicitly take into account the different effects of the inputs of the ground truth function into different outputs, in the so-called *componentwise Hölder* approach. The proposed new predictor yields smaller prediction errors than the standard KI. The controller presented in Chapter 3 is extended to the proposed CHoKI prediction model, enhancing the performance and with larger domains of attraction than the previous ones.

Chapter 5 exploits the fact that during the operation of the system, new measurements become available. Hence, incorporating these new measurements into the model, in an online fashion, could improve the prediction and therefore the performance. To this end, in this chapter an online learning MPC is proposed, based on a double prediction model: an initial safe model is in charge of the constraints satisfaction and feasibility of the controller, while an updated model is used to improve the performance of the closed-loop.

Chapter 6 the online learning approach presented in the previous chapter, but taking an exploration-exploitation approach that cautiously forces the system to stay in a safe region in which stability is guaranteed, while preventing the prediction model from becoming computationally intractable. A model predictive control for tracking is proposed which, together with the exploration policy, allows the system to move to unexplored regions while satisfying constraints, even in the case of low-dense initial data sets.

In addition, three appendices are included. **Appendix A** analyses several facts that must be taken into account when dealing with data-based methods. **Appendix B** presents further results of the kinky inference class of learning rules, and **Appendix C** addresses the stability analysis of standard MPC.

1.8 Publications

Part of the research developed during the elaboration of this thesis has been published in the following papers:

Journal papers

- Manzano, J. M., Limon, D., Muñoz de la Peña, D., and Calliess, J.-P. Robust learning-based MPC for nonlinear constrained systems. *Automatica 117* (2020), 108948
- Manzano, J. M., Limon, D., Muñoz de la Peña, D., and Calliess, J.-P. Output feedback MPC based on smoothed projected kinky inference. *IET Control Theory & Applications 13*, 6 (2019), 795–805
- Manzano, J. M., Limon, D., Muñoz de la Peña, D., and Calliess, J.-P. Online learning constrained MPC based on double prediction. *Accepted in the International Journal of Robust and Nonlinear Control* (2020)
- Manzano, J. M., Muñoz de la Peña, D., Calliess, J.-P., and Limon, D. Componentwise Hölder inference for robust learning-based MPC. *Submitted to the IEEE Transactions on Automatic Control* (2020)
- Manzano, J. M., Muñoz de la Peña, D., and Limon, D. Oracle-based economic predictive control. *Submitted to Computer Applications in Chemical Engineering* (2020)

Book chapters

- Alamo, T., Manzano, J. M., and Camacho, E. Robust design through probabilistic maximization. In *Uncertainty in Complex Networked Systems*. Springer, 2018, pp. 247–274

Conference papers

- Manzano, J. M., Limon, D., Muñoz de la Peña, D., and Calliess, J.-P. Data-based robust MPC with componentwise Hölder

kinky inference. In *2019 IEEE 58th Conference on Decision and Control (CDC)* (2019), IEEE, pp. 6449–6454

- Manzano, J. M., Nadales, J. M., Limon, D., and Muñoz de la Peña, D. Oracle-based economic predictive control. In *2019 IEEE 58th Conference on Decision and Control (CDC)* (2019), IEEE, pp. 4246–4251

- Blaas, A., Manzano, J. M., Limon, D., and Calliess, J.-P. Localised kinky inference. In *2019 18th European Control Conference (ECC)* (2019), IEEE, pp. 985–992

- Manzano, J. M., Limon, D., and Muñoz de la Peña, D. Técnicas de aprendizaje automatizado para la operación económica basada en datos de sistemas ciberfísicos. *XVII Simposio CEA de Ingeniería de Control y V Seminario de Innovación Docente en Automática* (2019)

- Manzano, J. M., Limon, D., Muñoz de la Peña, D., and Calliess, J.-P. Robust data-based model predictive control for nonlinear constrained systems. *IFAC-PapersOnLine 51*, 20 (2018), 505–510

- Maiworm, M., Limon, D., Manzano, J. M., and Findeisen, R. Stability of Gaussian process learning based output feedback model predictive control. *IFAC-PapersOnLine 51*, 20 (2018), 455–461

- Manzano, J. M., Limon, D., Alamo, T., and Callies, J.-P. Control predictivo basado en datos. *Actas de las XXXVIII Jornadas de Automática* (2017)

- Nadales, J. M., Manzano, J. M., Barriga, A., and Limon, D. FPGA parallel implementation of Lipschitz interpolation techniques. In *Submitted to the 59th IEEE Conference on Decision and Control* (2020), IEEE

- Manzano, J. M., Calliess, J.-P., Muñoz de la Peña, D., and Limon, D. Online learning robust MPC: an exploration-exploitation approach. In *Accepted for publication in the 21st IFAC World Congress* (2020), IFAC

Chapter 2

Output feedback MPC based on smoothed projected kinky inference

2.1 Introduction

As a non-parametric regression method, the computational effort when using KI for making predictions grows with the sample size. In MPC, the prediction model is frequently evoked to compute large numbers of predictions. Hence, the computational complexity of non-parametric methods often forbid their applicability in MPC frameworks. Therefore, reducing computational effort for prediction is crucial. To this end, in this chapter we propose a variation of kinky inference that is designed to speed up predictions.

Instead of making predictions on the basis of all observed data, our approach is based on using a different reduced subset of data for each region of a partition of the state space, to implement the smoothed kinky inference method presented in [34]. The use of different reduced data sets depending on the current state decreases the computational effort of the inference method. The smoothed predictions yield smooth control actions that result in better performance. The whole prediction method is called *smoothed projected kinky inference* (SPKI).

In addition, a new controller formulation that guarantees closed-loop robust stability by means of the use of different control and prediction horizons and a weighted terminal cost is also presented. This controller is an extension of the controller proposed in [83], which provides a stabilizing design without

the use of a terminal constraint, tailored to data-based models. Note that the calculation of a terminal robust invariant set may be very difficult to obtain, even for linear systems. This calculation could be impossible to derive in the context of black-box systems that concerns us, so avoiding its need is a key contribution in the considered scenario.

The result is an output-feedback predictive controller which, under the assumption that the model of the system is Hölder continuous, it is proven that the closed-loop system is input-to-state stable with respect to the measurement noises and prediction errors. Because of the generality of its formulation (output-feedback control of input constrained systems), it can be applied to a wide set of control problems. The proposed controller has been validated in simulation using the control of a continuous stirred tank reactor case study.

The results of this chapter have been published in [100].

2.2 Smoothed projected kinky inference

As motivated in Chapter 1, KI has some properties that makes it suitable for the estimation of prediction models for control purposes. However, the computational effort depends linearly on the size of the dataset. Furthermore, the prediction may not be smooth, which may limit its real time applications. This motivates the study of methods to reduce the computation cost and to smooth the predictions. This is particularly relevant in the derivation of prediction models for model predictive control, because, as it was mentioned before, the control techniques require the solution of an optimization problem at each sampling time.

2.2.1 Smoothed kinky inference

As stated when introducing the KI method, the resulting prediction may exhibit *kinks*, that is, sudden changes in the derivative of $\hat{\mathfrak{f}}(w)$. These non-differentiabilities are usually not desired, especially when working with optimizers that seek to minimize a target function.

In particular, the cost to be minimized in the model predictive controllers that will be presented in this thesis will be based on the resulting predictor, and the kinks may cause gradient-based optimization methods to present solving issues. To avoid these problems, we propose to use a modified version

of the predictor, known as *smoothed kinky inference* (SKI) [34]. In this predictor, a convex combination of various points surrounding the query q is used, as it is shown below:

$$\hat{\mathfrak{f}}_S(q;\theta,\mathcal{D}) = \sigma_0\hat{\mathfrak{f}}(q) + \sum_{i=1}^{n_w}\frac{\sigma_i}{2}\big(\hat{\mathfrak{f}}(q+e_i\delta) + \hat{\mathfrak{f}}(q-e_i\delta)\big), \tag{2.1}$$

with weights $\sum_{i=0}^{n_w}\sigma_i = 1$, where e_i denotes the vector with a 1 in the i-th coordinate and 0's elsewhere, and δ is the incremental factor. [1] The new predictor $\hat{\mathfrak{f}}_S$ is Hölder continuous and its Hölder parameters are the same as in the prediction function $\hat{\mathfrak{f}}$ [34].

The KI predictions at each one of the dimensional components of w can be used to calculate an approximation of the gradient of $\hat{\mathfrak{f}}_S$, and by extension, to estimate the gradient of the cost. This gradient can be given to the optimizer in order to increase the iteration speed. Such gradient is approximated by:

$$\frac{\partial\hat{\mathfrak{f}}_S}{\partial w_i} \simeq \frac{\hat{\mathfrak{f}}(w+e_i\delta) - \hat{\mathfrak{f}}(w-e_i\delta)}{2\delta}. \tag{2.2}$$

2.2.2 Projected kinky inference

In its standard form, the computation time to evaluate a prediction using KI grows linearly with the number of training data points contained in the data set. As predictions occupy a large amount of time during the repeated optimizations performed by our MPC controller, this property constitutes a serious computational bottleneck. In this section, we will address this issue. To this end, we partition the input workspace $\mathcal{W}$ into disjoint subsets, and base the prediction of any given query point $q \in \mathcal{W}$ only on sample points contained in the same (and neighbouring) subsets.

The input space $\mathcal{W} \subset \mathbb{R}^{n_w}$ of the data set $\mathcal{D}$ is divided into several partitions $\mathcal{W}_i$, such that the union of them conform the original space $\mathcal{W} = \cup(\mathcal{W}_i)$, and their intersection is null: $\cap_{i\neq j}\mathcal{W}_i = \emptyset$, $\forall j$. This partition is calculated offline, and then a classification algorithm is built to locate the partition $\mathcal{W}_i$ to which a query point belongs to. This partition can be calculated taking into account different objectives, as for instance, to ensure a regular distribution of data points.

Once the set of the partition where the query $q \in \mathcal{W}$ is located is found, the prediction is computed as

$$\hat{\mathfrak{f}}_{SP}(q;\theta,\mathcal{D}) := \hat{\mathfrak{f}}_S(q;\theta,\mathcal{D}_i), \tag{2.3}$$

[1] Note that the dependence of $\hat{\mathfrak{f}}$ w.r.t. to θ and $\mathcal{D}$ is omitted in favour of conciseness.

only taking into account sample points of a local subset of data points $\mathcal{D}_i$, where

$$\mathcal{D}_i = \{\mathcal{D} | w \in \mathcal{W}_{\mathcal{D}} : w \in (\mathcal{W}_i \oplus \mathcal{B}(R_i))\}, \tag{2.4}$$

and " $\oplus$ " represents the Minkowski sum. The selection of the data set $\mathcal{D}_i$ aims to include points in a ball of radius R_i around any query state, to make predictions better for points in the frontiers between partitions. We call the prediction rule $\hat{\mathfrak{f}}_{SP}(\cdot)$ smoothed projected kinky inference (SPKI).

Note that the cardinality of the set $\mathcal{D}_i$ is in general much lower than the cardinality of the original data set $\mathcal{D}$, and hence obtaining a prediction is less cumbersome. The radius R_i must be chosen appropriately, depending on the density of the data set, increasing it to ensure a minimum number of data points in $\mathcal{D}_i$. The Hölder constant is kept the same for every partition data set, although further extension to local parameters was proposed in [24].

It can be proven that if the partitions' data sets are chosen appropriately (that is, the value of R_i is large enough to capture all relevant data points), the prediction error can be equal to the prediction error of the inference obtained using the full data set.

The partition can be done using machine learning techniques such us the K-means algorithm for clustering or performing principal component analysis, useful given the recursive characteristic of the NARX model. Some references regarding this topic may be found in [34], [157]. In the case study presented in Section 2.4, the set of the partitions are hypercubes, that is, they are done independently through some dimensions of $\mathcal{W} \subset \mathbb{R}^{n_w}$.

Computational complexity

In what is to follow, we will describe this particular approach in greater detail and analyse the computational complexity. To this end, we modify the prediction to (with high probability) only depend on a finite number of data points that does not grow with the number of data points, while avoiding loss of consistency.

For simplicity, assume the sample inputs were drawn i.i.d. and are contained in a hypercube $H = \{w : w_i \in [0,1], \forall i = 1, \ldots, n_w\}$ (once the samples are drawn, we can always ensure this by appropriate rescaling of the workspace). We partition H into m^{n_w} sub-hypercubes H_i, indexed by the tuple $i := (i_1, ..., i_{n_w})$, where $i_j \in \{1, ..., m\}$ $(j = 1, ..., n_w)$. We define

$$H_{i_j} = \Big\{ w : i_j = \min\{m, \max\{1, \lfloor w_j m \rfloor + 1\}\}, \forall j = 1, ..., n_w \Big\}.$$

For simplicity, for now, assume predictions are based exclusively on one of the hypercubes H_i, i.e. without overlap, such that $H_i = \mathcal{W}_i$. We will consider the following question. Given a fixed partition of H into m^d hypercubes and assuming we have drawn a sample of $N_{\mathcal{D}}$ data points i.i.d. uniformly from H, what is the computational complexity of making a prediction? That is, how many basic computational steps $T_{N_{\mathcal{D}},m}$ are required to locate the partition to which the query belongs, that is, computing i and subsequently $\hat{\mathfrak{f}}_p(q;\theta,\mathcal{D})$ for any given query $q \in \mathcal{W}$?

To answer this question, we note that, for a given query point $q \in H_i$ computing the index $i = (i_1, ..., i_{n_w})$ can be done in $\mathcal{O}(n_w)$ basic computational steps (including potential rescaling if necessary). Once i is determined, we can compute the prediction $\hat{\mathfrak{f}}(q;\theta,\mathcal{D}_i)$ in $\mathcal{O}(N_i)$ computational steps, where N_i is the number of samples contained in $\mathcal{W}_i$. So, predicting the value of the query can be done in at most

$$T_{N,m} := aN_i + b\, n_w + c \in \mathcal{O}(N_i + n_w) \tag{2.5}$$

computational steps for some (algorithm-dependent) parameters a, b, c.

Now, N_i is a random variable whose distribution depends on the number of cubes m^{n_w} and the number of samples $N_{\mathcal{D}}$. Assume the data contain N_i samples $(w_1, ..., w_N)$. Let

$$E_{i,k} = \{(w_1, ..., w_N) : |\{w_1, ..., w_N\} \cap H_i| = k\}$$

be the event that hypercube H_i contains exactly $N_i = k$ sample points and let $p := \mathbb{P}(H_i) = m^{-n_w}$. We consider this situation as a sequence of N Bernoulli trials with p being the success probability. Given our assumption of i.i.d. uniformity,

$$\mathbb{P}(E_{i,k}) = \binom{N}{k} p^k (1-p)^{N-k}.$$

Therefore, the probability that N_i is within some desired interval $[\underline{k}, \bar{k}]$ is

$$\mathbb{P}(N_i \in \{\underline{k}, ..., \bar{k}\}]) = \sum_{k=\underline{k}}^{\bar{k}} \binom{N}{k} p^k (1-p)^{N-k}.$$

In summary, to answer our first question, given a partition into m^{n_w} hypercubes, we have

$$\mathbb{P}(T_{N,m} \in \{a\underline{k} + b\, n_w + c, ..., a\bar{k} + b\, n_w + c\}) = \sum_{k=\underline{k}}^{\bar{k}} \binom{N}{k} m^{-n_w k} (1 - m^{-n_w})^{N-k}. \tag{2.6}$$

With N_i following a binomial distribution, the expected value of the computational effort is

$$\begin{aligned}\mathbb{E}(T_{N,m}) &= apN + bn_w + c \\ &= am^{-n_w}N + bn_w + c \in \mathcal{O}(m^{-n_w}N + n_w). \end{aligned} \tag{2.7}$$

These results can be utilised to answer the question of how to choose parameters m to control the average and maximal computational effort per query point. This is of particular importance in MPC, where predictions are made repeatedly during an optimization process that needs to terminate within a given time span that is smaller than the sampling period length. Of course, in practise, we may furthermore cap the per query prediction time by organising the data in $\mathcal{D}_i$ in a list, and only utilising the data of that list to make a prediction that can be processed within the prescribed computational budget.

In our analysis, we have assumed that we only take H_i into account when computing $\mathcal{D}_i$. However, to facilitate smoothness of the prediction surface we also consider a selection of adjacent hypercubes. In the full setting we include samples from all neighbouring hypercubes in the set

$$\{H_{i+j} | j \in \{-1, 0, 1\}^{n_w}\}.$$

Of course, this means that the computational complexity increases by a factor of $\mathcal{O}(3^{n_w})$. If one desires to avoid this exponential dependence on n_w, for high-dimensional problems in particular, it may instead be desirable to achieve a compromise by only including those neighbouring hypercubes that share a face with H_i. That is, we only take into account samples from the union of the hypercubes in the set

$$\Big\{H_{i+j} | j \in \{-1, 0, 1\}^{n_w} \wedge \|j\|_1 \leq 1\Big\}.$$

Since the latter set only contains $\mathcal{O}(n_w)$ hypercubes, with this choice, the overall complexity increases merely by a factor that grows linearly with the input space dimensionality, n_w.

As a final remark, it should be emphasised that several extensions of our analysis might have to be made to adjust it to better match conditions that might arise in reality. Firstly, the data might not have been collected i.i.d. from an uniform distribution. Indeed, the distribution might be unknown (or distributional assumption questionable). In that case, an approach would be to fit a distribution to the collected data via standard density estimation

techniques. In that case, our hypercubes will have varying success probabilities p_i that have to be taken into account when computing the expected computational effort and the probabilistic bounds. Secondly, the inputs may follow a different distribution or none at all. In that case, depending on the case, one should marginalise over the input distribution or might resort to worst-case analysis.

2.3 Data-based predictive control

This section presents a model predictive controller designed to ensure robust stability of the real system in closed-loop, considering soft constraints in the outputs and hard constraints in the inputs, i.e., $y \in \mathcal{Y}_{\text{soft}}$, $u \in \mathcal{U}$. In the proposed controller, a prediction horizon larger than the control horizon has been considered in order to enhance the closed-loop performance and to increase the domain of attraction. This is particularly interesting when the system to be controlled is nonlinear.

The NARX models described in Section 1.3.2 (on page 12) are used. For a given state of the plant and a sequence of future control actions $\mathbf{u}$ at sampling time k, the prediction model to be used in the MPC is converted from the prediction function $\hat{\mathfrak{f}}_{SP}$ to state-space as follows:

$$\begin{aligned} \hat{x}(j+1|k) &= \hat{F}(\hat{x}(j|k), u(k+j); \theta, \mathcal{D}), && \text{(2.8a)} \\ \hat{y}(j|k) &= M\hat{x}(j|k), && \text{(2.8b)} \end{aligned}$$

where the predicted state

$$\begin{aligned} \hat{x}(j|k) = \; & (\hat{y}(j|k), \cdots, \hat{y}(1|k), y(k), \cdots, y(k+j-n_a), \\ & u(k+j-1), \cdots, u(k), \cdots, u(k+j-n_b)), \end{aligned}$$

includes past measurements y if $n_a \geq j$ (or $n_b > j$ for the u), or only estimated future values $\hat{y}$ otherwise. The model is:

$$\begin{aligned} \hat{F}(\hat{x}(j|k), u(k+j); \theta, \mathcal{D}) = \; & (\hat{\mathfrak{f}}(\hat{x}(j|k), u(k+j); \theta, \mathcal{D}), \\ & \hat{y}(j|k), \cdots, y(k), \cdots, \\ & y(k+j-n_a+1), \\ & u(k+j), \cdots, u(k+j-n_b+1)), \end{aligned}$$

and the output matrix is $M = [I_{n_y}, \mathbb{0}, \ldots, \mathbb{0}] \in \mathbb{R}^{n_x \times n_x}$, being I_{n_y} the identity matrix.

To derive the predictive controller, a positive definite stage cost function $\ell(y,u)$ is defined as:

$$\ell(y,u) = \ell_t(y,u) + \ell_b(y),$$

where the term $\ell_t(y,u)$ penalizes the tracking error of inputs and outputs w.r.t. the reference set-point (assumed to be given by (0,0)) and $\ell_b(y)$ is a barrier function that implements soft constraints on the outputs. The barrier cost function $\ell_b(\cdot)$ must satisfy that $\ell_b(y) = 0$ for all $y \in \mathcal{Y}_{\text{soft}}$ and that there exists two $\mathcal{K}$-functions α_b and β_b such that

$$\alpha_b(\mathfrak{d}(y, \mathcal{Y}_{\text{soft}})) \geq \ell_b(y) \geq \beta_b(\mathfrak{d}(y, \mathcal{Y}_{\text{soft}})) \; \forall y,$$

where $\mathfrak{d}(y, \mathcal{Y}_{\text{soft}})$ is a measure of the distance from y to $\mathcal{Y}_{\text{soft}}$.

Assumption 2.1. *There exist two $\mathcal{K}$-functions α_y and α_u such that*

$$\ell(y,u) \geq \alpha_y(\|y\|) + \alpha_u(\|u\|).$$

Besides, $\ell(y,u)$ is assumed to be continuous.

The optimization problem $P_{N_c,N_p}(x(k), \mathcal{D})$ to be solved is the following:

$$\begin{aligned}
\min_{\mathbf{u}} \quad & V_{N_c,N_p}(x(k), \mathbf{u}) = \sum_{i=0}^{N_c-1} \ell(\hat{y}(i|k), u(k+i)) \\
& + \sum_{i=N_c}^{N_p-1} \ell(\hat{y}(i|k), \kappa_f(\hat{x}(i|k)) + \lambda V_f(\hat{x}(N_p|k)) && \text{(2.9a)} \\
\text{s.t.} \quad & \hat{x}(0|k) = x(k) && \text{(2.9b)} \\
& \hat{x}(j+1|k) = \hat{F}(\hat{x}(j|k), u(k+j); \theta, \mathcal{D}),\ j \in \mathbb{I}_0^{N_p-1} && \text{(2.9c)} \\
& \hat{y}(j|k) = M\hat{x}(j|k),\ j \in \mathbb{I}_0^{N_p-1} && \text{(2.9d)} \\
& u(k+j) \in \mathcal{U},\ j \in \mathbb{I}_0^{N_p-1}, && \text{(2.9e)}
\end{aligned}$$

where $\lambda \geq 1$ is a design parameter of the controller, $N_p \in \mathbb{N}$ is the prediction horizon, $N_c \leq N_p$ is the control horizon and $\mathcal{U}$ is the compact set defined by the constraints on the inputs. Asymptotic stability is ensured by designing a suitable terminal cost function $V_f(\cdot)$ and a terminal control law $\kappa_f(\cdot)$. Notice that no terminal constraint is considered, since the terminal cost function V_f is weighted by a factor $\lambda \geq 1$, following the procedure proposed in [83], and extended for a prediction horizon larger than the control horizon.

The MPC control law is $u(k) = \kappa_{\text{MPC}}(x(k); \theta, \mathcal{D}) = u^*(0)$, which depends on the NARX state of the system and the collected historical data set $\mathcal{D}$.

2.3.1 Stability analysis of the controller

In order to demonstrate the stability of the proposed controller, first the asymptotic stability of the *nominal case* will be proven, that is, when the proposed controller is used to regulate a system whose model is equal to the prediction one.

Theorem 2.1. *Suppose there exists a terminal control law* $\kappa_f(\cdot) : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$, *a terminal cost function* $V_f(\cdot) : \mathbb{R}^{n_x} \to \mathbb{R}$ *and a region* $\Omega_\gamma = \{x : V_f(x) \leq \gamma\}$, *where* γ *is a positive constant such that for all* $x \in \Omega_\gamma$: [2]

$$\begin{aligned} \alpha_1(\|x\|) \leq V_f(x) \leq \alpha_2(\|x\|), \\ V_f(x^+) - V_f(x) \leq -\ell_t(y, \kappa_f(x)), \\ Mx \in \mathcal{Y}_{\text{soft}}, \\ \kappa_f(x) \in \mathcal{U}. \end{aligned}$$

Consider the so-called domain of attraction*, defined as*

$$X_{N_c,N_p}(\lambda) = \left\{ x : V^*_{N_c,N_p}(x(k)) \leq (N_p - n)\phi + \lambda\gamma \right\},$$

where ϕ *is a positive constant and* $n = \max(n_a, n_b)$.

Then, given $N_p \geq n$, *for all* $\lambda \geq 1$ *and all* $x(0) \in X_{N_c,N_p}(\lambda)$, *the closed-loop system* $x^+ = \hat{F}(x, \kappa_{\text{MPC}}(x; \theta, \mathcal{D}); \theta, \mathcal{D})$ *is asymptotically stable.*

Proof.

Notice that if $x \in \Omega_\gamma$, $y \in \mathcal{Y}_{\text{soft}}$ so $\ell_b = 0$, and thus $\ell(y, u) = \ell_t(y, u)$. For the sake of clarity it may be used x_j or $x(j|k)$ indistinctly. The proof uses the following three lemmas:

Lemma 2.1. *If the assumptions of Theorem 2.1 hold, then*

$$V^*_{N_c,N_p}(x(k)) \leq \lambda V_f(x(k)),\ \forall x(k) \in \Omega_\gamma.$$

Proof. For any system controlled by $u(k + j) = \kappa_f(x(j|k))$,

$$V_f(x(j|k)) - V_f(x(j + 1|k)) \geq \ell(y(j|k), \kappa_f(x(j|k))).$$

[2] The notation x^+ implies $x(k + 1)$, given $x(k)$.

Summing for the whole sequence:

$$\begin{aligned}\sum_{j=0}^{N_p-1} V_f(x(j|k)) - V_f(x(j+1|k)) &= V_f(x(k)) - V_f(x(N_p|k)) \\ &\geq \sum_{j=0}^{N_p-1} \ell(y(j|k), \kappa_f(x(j|k)))\end{aligned}$$

and by optimality, provided that $\lambda \geq 1$, $\forall N_p \geq N_c \geq 1$

$$\begin{aligned}\lambda V_f(x(k)) &\geq \sum_{j=0}^{N_p-1} \ell(y(j|k), \kappa_f(x(j|k))) + \lambda V_f(x(N_p|k)) \\ &\geq V^*_{N_c,N_p}(x(k)). \end{aligned} \tag{2.10}$$

□

Lemma 2.2. *If $x^*(j|k) \notin \Omega_\gamma$, $\forall j = 0, \cdots, N_p - 1$, then*

$$V^*_{N_c,N_p}(x(k)) \geq (N_p - n)\phi + \lambda\gamma.$$

Proof. Let define $\ell_{\text{ext}}(x, u)$ as the stage cost extended to consider a positive definite function of the whole state:

$$\ell_{\text{ext}}(x(k), u(k)) = \frac{1}{n}\sum_{i=0}^{n} \ell(y(k-n+i), u(k-n+i)),$$

where $n = \max(n_a, n_b)$, and with a slight abuse of notation, the terms given by $y(k-n+i)$ and $u(k-n+i)$ are set to 0 if $y(k-n+i)$ or $u(k-n+i)$ are not included in $x(k)$, respectively.

Then, ϕ is defined as a positive constant such that $\forall x \notin \Omega_\gamma$, $\ell_{\text{ext}}(x, u) \geq \phi$. Hence,

$$\begin{aligned}V^*_{N_c,N_p}(x(k)) &= \sum_{j=0}^{N_p-1} \ell(y^*(j), u^*(j)) + \lambda V_f(x^*(N_p)) \\ &\geq \sum_{k=0}^{\left\lfloor \frac{N_p}{n} \right\rfloor - 1} \sum_{i=0}^{n} \ell(y^*(k+n-i), u^*(k+n-i)) + \lambda V_f(x^*(N_p)) \\ &\geq \left\lfloor \frac{N_p}{n} \right\rfloor n\phi + \lambda\gamma \\ &\geq (N_p - n)\phi + \lambda\gamma.\end{aligned}$$

□

Lemma 2.3. *If the assumptions of Theorem 2.1 hold, and if* $x^*(N_p|k) \notin \Omega_\gamma$, *then*

$$x^*(j|k) \notin \Omega_\gamma,\ \forall j = 0, \ldots, N_{p-1}.$$

Proof. This lemma is proven by contradiction:

(i) Assume that there exists any instant $i < N_c$ in which $x_i^* \in \Omega_\gamma$. From Lemma 2.1 it is inferred that for a problem of horizons $N_c - i,\ N_p - i$,

$$\lambda V_f(x_i^*) \geq V^*_{N_c-i,N_p-i}(x_i^*).$$

From Bellman's optimality principle, and provided that $\ell(y, u)$ is positive definite, it is derived that

$$V^*_{N_c-i,N_p-i}(x_i^*) \geq \lambda V_f(x^*_{N_p}),$$

and given that $x^*_{N_p} \notin \Omega_\gamma$, then

$$\lambda V_f(x^*_{N_p}) > \lambda\gamma.$$

Hence, $\lambda V_f(x_i^*) > \lambda\gamma$, which is a contradiction, so $x_i^* \notin \Omega_\gamma$.

(ii) Contrary, assume there exists an instant $i \geq N_c$ in which $x_i^* \in \Omega_\gamma$. Then, $x^*_{j+1} = \hat{F}(x_j^*, k_f(x_j^*))$ for $j = i, \ldots, N_p$. Since Ω_γ is a positive invariant set for this system, we have that $x_j^* \in \Omega_\gamma$ for $j = i, \ldots, N_p$, leading again to a contradiction. □

Given these three lemmas, we have that $\forall x \in X_{N_c,N_p}(\lambda)$, if $x^*_{N_p} \notin \Omega_\gamma$, then $x_j^* \notin \Omega_\gamma\ \forall j = 0, \ldots, N_p - 1$, and therefore

$$V^*_{N_c,N_p}(x(k)) \geq (N_p - n)\phi + \lambda\gamma.$$

So on the contrary, if $V^*_{N_c,N_p}(x(k)) \leq (N_p - n)\phi + \lambda\gamma$, then $x^*_{N_p} \in \Omega_\gamma$. Next, we are ready to prove the statement of the theorem. First, recursive feasibility is proven:

Assume that $x(k) \in X_{N_c,N_p}(\lambda)$, then $x^*_{N_p} \in \Omega_\gamma$. Define the sequence of future control inputs for $x(k + 1)$ as

$$\tilde{u}(k + j + 1) = \begin{cases} u^*(k + j) & j = 0, \ldots, N_c - 1 \\ \kappa_f(\tilde{x}(j|k + 1)) & j = N_c, \ldots, N_p - 1, \end{cases}$$

and define $\tilde{x}(j|k+1)$ as the predicted trajectory. Then, from standard arguments of MPC proofs, this sequence is feasible for the optimization problem,

since $u^*(k)$ is feasible. Besides, the difference between the costs associated to these trajectories is:

$$\tilde{V}_{N_c,N_p}(x(k+1)) - V^*_{N_c,N_p}(x(k)) = -\ell(y(k),u^*(k)) + \Big[\ell(\tilde{y}(N_p|k+1),\kappa_f(\tilde{x}(N_p|k+1))) + \lambda V_f\left(\tilde{x}(N_p|k+1)\right) - \lambda V_f(x^*(N_p|k))\Big].$$

The sum among brackets is negative, since $x^*_{N_p} \in \Omega_\gamma$. Thus,

$$\tilde{V}_{N_c,N_p}(x(k+1)) - V^*_{N_c,N_p}(x(k)) \leq -\ell(y(k),u^*(k)),$$

and by optimality,

$$V^*_{N_c,N_p}(x(k+1)) \leq \tilde{V}_{N_c,N_p}(x(k+1)),$$

$$V^*_{N_c,N_p}(x(k+1)) - V^*_{N_c,N_p}(x(k)) \leq -\ell(y(k),u^*(k)). \tag{2.11}$$

So since $x(k) \in X_{N_c,N_p}(\lambda)$ implies that $V^*_{N_c,N_p}(x(k)) \leq (N_p-n)\phi + \lambda\gamma$, then $V^*_{N_c,N_p}(x(k+1)) \leq (N_p-n)\phi + \lambda\gamma$, implying that $x(k+1) \in X_{N_c,N_p}(\lambda)$. Therefore the closed-loop system is recursively feasible.

Then, asymptotic stability is proven. Given the definition of $x(k)$,

$$\begin{aligned}\sum_{j=k-n}^{k} \ell(y(j),u(j)) &\geq \sum_{j=0}^{n} \alpha_y(\|y(k-j)\|) + \alpha_u(\|u(k-j)\|)\\ &\geq \alpha_x(\|x(k)\|) + \alpha_u(\|u(k)\|)\\ &\geq \alpha_x(\|x(k)\|),\end{aligned} \tag{2.12}$$

where $n = \max(n_a, n_b)$ and α_y, α_u, and α_x are $\mathcal{K}$-functions. Defining

$$W(x_k) = \sum_{j=0}^{n} V^*_{N_c,N_p}(x(k-j)),$$

we can infer that

$$\begin{aligned}W(x(k+1)) - W(x(k)) &= V^*_{N_c,N_p}(x(k+1)) - V^*_{N_c,N_p}(x(k-n))\\ &\leq -\sum_{j=k-n}^{k} \ell(y(j),u(j)) \leq -\alpha_x(\|x(k)\|),\end{aligned}$$

where the first inequality is derived by recursion of (2.11), and the second inequality comes from (2.12). Besides,

$$W(x) \geq \sum_{j=k-n}^{k} \ell(y(j),u(j)) \geq \alpha_x(\|x\|),$$

and, by Prop. B.25 in [133], there exists a $\mathcal{K}$-function $\alpha_2(\cdot)$ such that $W(x) \leq \alpha_2(\|x\|)$. Hence, $W(x(k))$ is a Lyapunov function of the system, which is asymptotically stable. □

Remark 2.1. *Whereas N_p and λ are design parameters, ϕ and γ are problem-dependent parameters. They require solving three global optimization problems, whose ingredients are sketched below.*

According to the definition of γ in Theorem 2.1, let us define γ_1 as:

$$\begin{aligned} \gamma_1 \;&=\; \arg\min_{x} V_f(x) \\ \text{s.t.} \quad & \lambda V_f(\hat{F}(x, \kappa_f(x); \theta, \mathcal{D})) - \lambda V_f(x) \geq -\ell(Mx, \kappa_f(x)). \end{aligned} \tag{2.13}$$

It must also be guaranteed that for all $V_f(x) \leq \gamma$, $\kappa_f(x) \in \mathcal{U}$, so we define

$$\begin{aligned} \gamma_2 \;&=\; \max_{\gamma} \gamma \\ \text{s.t.} \quad & \kappa_f(x) \in \mathcal{U} \\ & Mx \in \mathcal{Y}_{\text{soft}},\ \forall x \in \{x : V_f(x) \leq \gamma\}. \end{aligned} \tag{2.14}$$

Then, $\gamma = \min(\gamma_1, \gamma_2)$. In addition, ϕ is obtained as:

$$\begin{aligned} \phi \;&=\; \min \ell_{\text{ext}}(x, \kappa_f(x)) \\ \text{s.t.} \quad & V_f(x) \geq \gamma, \end{aligned} \tag{2.15}$$

where the extended stage cost ℓ_{ext} is defined in Lemma 2.2 (on page 36). Note that these calculations are done offline.

Corollary 2.1. *Under the assumptions of Theorem 2.1, it can be proven that:*

$$(i) \quad X_{N_c,N_p}(\lambda_1) \subseteq X_{N_c,N_p}(\lambda_2)\ \forall \lambda_1 \leq \lambda_2 \tag{2.16}$$

$$(ii) \quad X_{N_c,N_1}(\lambda) \subseteq X_{N_c,N_2}(\lambda)\ \forall N_1 \leq N_2 \tag{2.17}$$

Proof.

First, it is proven that if $\lambda_1 \leq \lambda_2$ and considering that $x \in X_{N_c,N_p}(\lambda_1)$, then $x \in X_{N_c,N_p}(\lambda_2)$. Define $\hat{u}^*$, $\hat{x}^*$ and $\hat{V}^*_{N_c,N_p}$ as the optimal solution for $x \in X_{N_c,N_p}(\lambda_1)$ and $\tilde{u}^*$, $\tilde{x}^*$ and $\tilde{V}^*_{N_c,N_p}$ as the optimal solution for $x \in X_{N_c,N_p}(\lambda_2)$.

Taking into account that $\hat{u}^*$ is a feasible solution of the optimization problem for λ_2,

$$\begin{aligned}
\tilde{V}^*_{N_c,N_p}(x) &= \sum_{i=0}^{N_p-1} \ell(\tilde{x}^*_i, \tilde{u}^*_i) + \lambda_2 V_f(\tilde{x}^*_{N_p}) \\
&\leq \sum_{i=0}^{N_p-1} \ell(\hat{x}^*_i, \hat{u}^*_i) + \lambda_2 V_f(\hat{x}^*_{N_p}) \\
&= \sum_{i=0}^{N_p-1} \ell(\hat{x}^*_i, \hat{u}^*_i) + \lambda_1 V_f(\hat{x}^*_{N_p}) + \lambda_2 V_f(\hat{x}^*_{N_p}) - \lambda_1 V_f(\hat{x}^*_{N_p}) \\
&= \hat{V}^*_{N_c,N_p}(x) + (\lambda_2 - \lambda_1) V_f(\hat{x}^*_{N_p}) \\
&\leq \hat{V}^*_{N_c,N_p}(x) + (\lambda_2 - \lambda_1)\gamma \\
&\leq (N_p - n)\phi + \lambda_1\gamma + (\lambda_2 - \lambda_1)\gamma \\
&= (N_p - n)\phi + \lambda_2\gamma,
\end{aligned}$$

and hence $x \in X_{N_c,N_p}(\lambda_2)$. □

Now it is proven that if $N_1 \leq N_2$ and $x \in X_{N_c,N_1}(\lambda)$, then $x \in X_{N_c,N_2}(\lambda)$.

Provided that $V^*_{N_c,N_1}(x(k)) \leq (N_1 - n)\phi + \lambda\gamma$, $\forall x \in X_{N_c,N_1}(\lambda)$ and $N_1 \leq N_2$, from (2.10) it is derived that

$$V_f(x^*_{N_1}) \geq \sum_{j=N_1}^{N_2-1} \ell(y^*_j, \kappa_f(x^*_k)) + V_f(x^*_{N_2}),$$

which implies that $V^*_{N_c,N_1} \geq V^*_{N_c,N_2}$, so

$$V^*_{N_c,N_2} \leq V^*_{N_c,N_1} \leq (N_1 - n)\phi + \lambda\gamma \leq (N_2 - n)\phi + \lambda\gamma,$$

and hence $X_{N_c,N_1} \subseteq X_{N_c,N_2}$ $\forall N_1 \leq N_2$. □

Remark 2.2. *From standard theory of MPC, it is well known that increasing the control horizon N_c increases the domain of attraction $X_{N_c,N_p}(\lambda)$, with the respective increase in the number of decision variables. Corollary 2.1 provides other two ways of increasing the domain of attraction of the MPC: (i) increasing the weighting factor λ and (ii) increasing the prediction horizon N_p. A larger weighting factor provides a larger domain of attraction, but the performance might be worse. On the other hand, a larger prediction horizon does not only increase the domain of attraction, but it also improves the performance, since $V^*_{N_c,N_2} \leq V^*_{N_c,N_1}$. The number of decision variables does not increase with N_p, but it increases the number of constraints in the optimization problem, which might lead to a mild increase of the computational cost.*

Once the nominal asymptotic stability is proven, then it is shown that the real plant, controlled with the proposed predictive control law, is input-to-state stable w.r.t. the model mismatches under the following assumption.

Assumption 2.2. *It is assumed that the radius R_i in equation (2.4) is large enough to ensure that the function $\hat{\mathfrak{f}}_{SP}(\cdot,\cdot)$ is Hölder continuous.*

Definition 2.1. *A system $x^+ = f(x) + d$ is input-to-state stable (ISS) w.r.t. d if there exists a $\mathcal{KL}$ function β and a $\mathcal{K}$ function α such that $\|x(k)\| \leq \beta(\|x(0)\|, k) + \sup_{j\in[0,k]} \alpha(\|d(j)\|)$.*

Now, the stability result is stated in the following theorem.

Theorem 2.2. *Assume that the stage cost function $\ell(\cdot,\cdot)$, the terminal control law $\kappa_f(\cdot)$ and the terminal cost function $V_f(\cdot)$ are uniformly continuous functions in the feasibility region $X_{N_c,N_p}(\lambda)$. Assume that the hypotheses of Theorem 2.1 hold. Then, the real plant, controlled by $u(k) = \kappa_{MPC}(x(k); \theta, \mathcal{D})$ is input-to-state stable w.r.t. the model mismatch signal:*

$$d(k) = |f(x(k), u(k)) - \hat{\mathfrak{f}}(x(k), u(k); \theta, \mathcal{D})|.$$

Proof.

First, notice that the model of real plant can be posed as $f(x(k), u(k)) = \hat{\mathfrak{f}}(x(k), u(k)) + d(k)$ and the signal $d(k)$ is bounded by μ. Given that the function $\hat{\mathfrak{f}}(\cdot,\cdot)$ is Hölder continuous, the model function $\hat{F}(\cdot,\cdot)$ is also Hölder continuous. Since the stage cost function $\ell(\cdot,\cdot)$ and the terminal cost function $V_f(\cdot)$ are continuous functions and $X_{N_c,N_p}(\lambda)$ is a compact set, then they are uniformly continuous. From Proposition 1, case C1, in [81], it is inferred that the closed-loop system is ISS w.r.t. the signal that describes the model mismatch, $d(k)$. □

Remark 2.3. *This theorem resorts on Assumption 2.2, which may be difficult to demonstrate. Notice that the assumption can always be satisfied since for $\mathcal{D}_i = \mathcal{D}$ the model function is Hölder continuous. In case one cannot ensure Hölder continuity, stability is not lost. Instead, according to the definition of input-to-state practical stability (ISpS, see Definition 6 in [81]), it can be proven that if the discontinuity jump is upper bounded by c, then*

$$\|\hat{\mathfrak{f}}_{SP}(w + \Delta w, u) - \hat{\mathfrak{f}}_{SP}(w, u)\| \leq \sigma(c) + \hat{L}\|\Delta w\|^p,$$

where σ is a $\mathcal{K}$-function. Using similar arguments to the proof of Theorem 2.2 and Definition 6 in [81], it can be proven that the closed-loop system is ISpS, since there exists a $\mathcal{KL}$-function β and $\mathcal{K}$-functions α and ω such that:

$$\|x(k)\| \leq \beta(\|x(0)\|, k) + \sup_{j\in[0,k]} \alpha(\|d(j)\|) + \omega(c).$$

Remark 2.4. *The calculation of the control action requires the solution of a mathematical programming problem on line. This is typically a gradient-based algorithm that may be sensitive to abrupt variations of the gradients, which may be derived from discontinuity of the derivative of the models. As it is illustrated in the case study, this may produce convergence issues leading to spikes in the evolution of the control action. To avoid this effect, the tracking stage cost function can be modified, adding a term penalizing the variation of the control input, i.e.* $\|u(k+i) - u(k+i-1)\|_S^2$. *The resulting tracking stage cost can be recasted, w.l.o.g., as a function of the state (that is, the regressor),* $\ell_t(x, u)$*, provided that* $n_b \geq 1$*. The stability analysis presented can be extended to this case.*

2.4 Case study

In this section we consider the control of the continuously-stirred tank reactor presented in [139]. The experiments carried out to generate the data sets and the procedure to tune the proposed controller are described in detail.

The input of the reactor is the reference of the coolant temperature T_r (K) and the output is the concentration of the reactant, C_A ($\mathrm{mol\,l^{-1}}$), in the reaction $A \rightarrow B$. The temperature of the tank and the coolant are given by T and T_c (K), respectively. It is assumed that the evolution of the plant is given by the following set of differential equations:

$$\frac{dC_A(t)}{dt} = \frac{q_0}{V} \cdot (C_{Af} - C_A(t)) - k_0 \cdot \exp\left(-\frac{E}{R \cdot T(t)}\right) \cdot C_A(t) \tag{2.18a}$$

$$\frac{dT(t)}{dt} = \frac{q_0}{V} \cdot (T_f - T(t)) + \frac{(-\Delta H_r) \cdot k_0}{\varrho \cdot C_p} \cdot \exp\left(-\frac{E}{R \cdot T(t)}\right) \cdot C_A(t) + \frac{U \cdot A}{V \cdot \varrho \cdot C_p} \cdot (T_c(t) - T(t)) \tag{2.18b}$$

$$\frac{dT_c(t)}{dt} = \frac{T_r(t) - T_c(t)}{\tau_c} \tag{2.18c}$$

The parameters of the model are given in Table 2.1. Note that the model is only used to carry out simulations, no information is used to design the controller. It is also assumed that the concentration sensor adds an error of 2.5% of the measurement. The error is generated randomly for each measurement using an uniform distribution. The constraints in the input are $T_r^{\min} = 335\,\mathrm{K}$ and $T_r^{\max} = 372\,\mathrm{K}$.

Param.	Definition	Value	Units
q_0	Input flow of the reactive	10	$\mathrm{l\,min^{-1}}$
V	Liquid volume in the tank	150	l
k_0	Frequency constant	6×10^{10}	$\mathrm{min^{-1}}$
E/R	Arrhenius constant	9750	K
$-\Delta H_r$	Enthalpy of the reaction	10000	$\mathrm{J\,mol^{-1}}$
UA	Heat transfer coefficient	70000	$\mathrm{J\,min^{-1}\,K^{-1}}$
ϱ	Density	1100	$\mathrm{g\,l^{-1}}$
C_p	Specific heat	0.3	$\mathrm{J\,g^{-1}\,K^{-1}}$
τ	Time constant	1.5	min
C_{Af}	C_A in the input flow	1	$\mathrm{mol\,l^{-1}}$
T_f	Temperature (input flow)	370	K

Table 2.1: Parameters of the system

First, the set of equilibrium points of the system is estimated using a sequence of steps in the input, from $T_r^{\min}$ to $T_r^{\max}$ with 0.5 K increments, and each step long enough to reach a steady sate. The result is shown in Figure 2.1. Apart from obtaining the equilibrium points of the system, this test is used to adjust the sampling time, which is set to $\tau_s/20 = 30\,\mathrm{s}$, where τ_s stands for the mean settling time of the sequence of steps applied.

This test is also used to bound the output variable C_A, which is defined by $C_A^{\max} = 0.8053\,\mathrm{mol\,l^{-1}}$ and $C_A^{\min} = 0.2\,\mathrm{mol\,l^{-1}}$ (note that the system has an inverse static characteristic). This restriction is treated by means of a soft constraint, using a barrier cost. In addition, an equilibrium point is chosen to be the reference operating point, given by $C_A^{\mathrm{ref}} = 0.439\,\mathrm{mol\,l^{-1}}$, $T_r^{\mathrm{ref}} = 356\,\mathrm{K}$.

After defining the set of equilibrium points, a set of experiments are carried out to obtain the data for the predictor. The experiments are designed using the methodologies presented in [134] to identify the dynamics of a system within a workspace: a relay test is carried out to estimate the crossover frequency of the system around the operating point [12], with limits $T_r^{\max}$ and $T_r^{\min}$. The crossover frequency obtained is 3.7 mHz. Figure 2.2 shows the simulation.

Then, a sequence of chirp signals covering the workspace are applied to generate the raw data set containing the trajectories of concentrations and temperatures, $\mathcal{D}_{\mathrm{raw}}$. The parameters of the chirp signals (length of the signal τ_f, initial and final frequency f_0, f_f, center T_{r_0} and amplitude A) are shown in Table 2.2. The simulation is shown in Figure 2.3. The set of

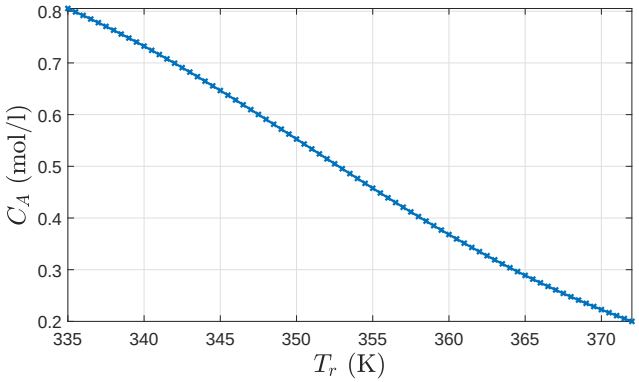


Figure 2.1: Set of equilibrium points of the CSTR.

Signal	τ_f (min)	f_0 (mHz)	f_f (Hz)	T_0 (K)	A (K)
1	1000	3.7	0.37	341.17	6.1667
2	1000	3.7	0.37	347.33	6.1667
3	1000	3.7	0.37	353.5	6.1667
4	1000	3.7	0.37	359.67	6.1667
5	1000	3.7	0.37	365.83	6.1667
6	5000	1.7	0.15	353.5	18.5
7	5000	3.7	0.37	353.5	18.5

Table 2.2: Parameters of the chirp signals.

equilibrium points are added to the data set obtained with the chirp signals (the resulting data points are represented in Figure 2.4).

In addition, several tests with random input signals are carried out to obtain data sets for cross-validation (see Figure 2.5). The same input signal is applied several times in order to estimate a bound of the measurement error of C_A. This error is estimated as $\bar{\mathfrak{e}} = 0.02\,\mathrm{mol\,l^{-1}}$, which is the expected value, given the 2.5% measurement error used to generate the simulation.

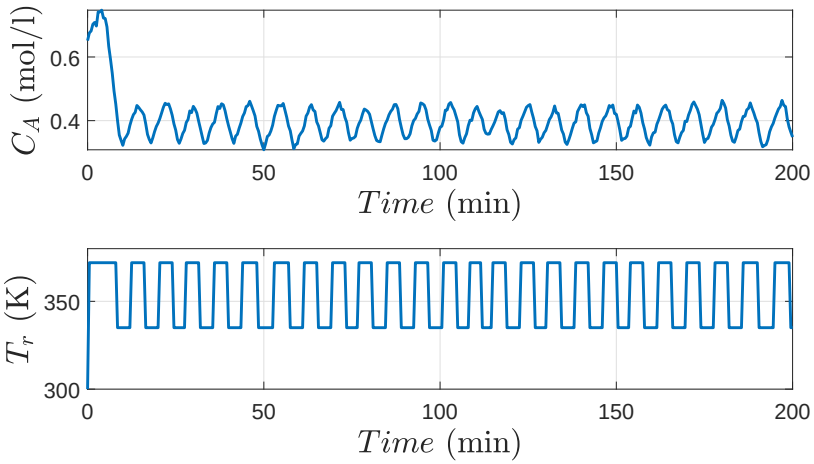


Figure 2.2: Relay test response.

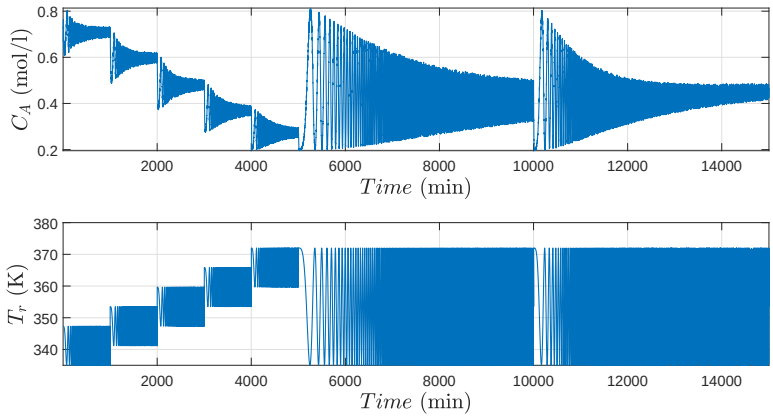


Figure 2.3: Chirp signals used to obtain the data set.

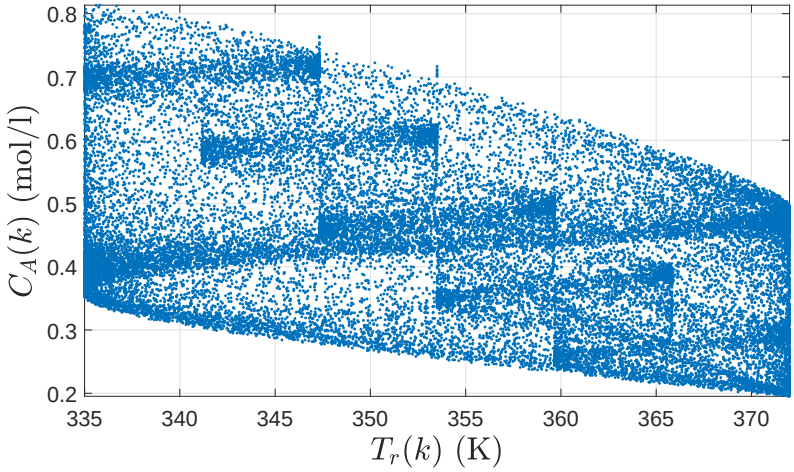


Figure 2.4: 2-D representation of $\mathcal{D}_{\text{raw}} : \{(C_A(k), T_r(k)), N_{\mathcal{D}} = 30000\}$.

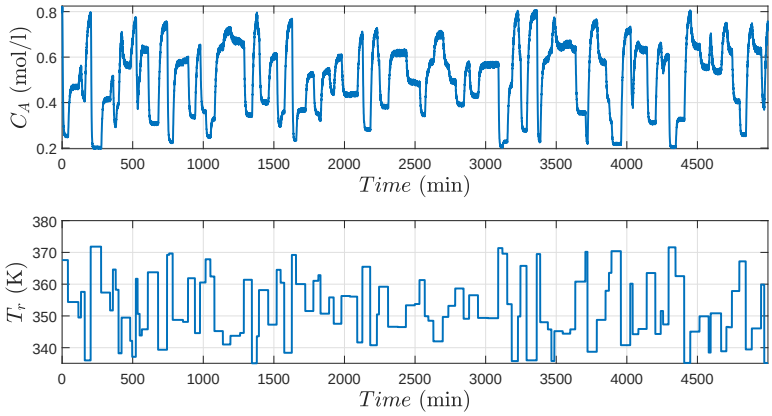


Figure 2.5: Test used for cross-validation.

In order to implement the prediction and optimization problem, it is desirable that the input-output variables are suitably scaled, so every component is expressed in the range [0,1]. The data set $\mathcal{D}_{\text{raw}}$ is scaled using the following expressions (also used for the maximum, minimum and reference values):

$$u = \frac{T_r - T_r^{\min}}{T_r^{\max} - T_r^{\min}}, \qquad y = \frac{C_A - C_A^{\min}}{C_A^{\max} - C_A^{\min}}.$$

To obtain the SPKI model, the parameters n_a and n_b of the NARX regressor have to be chosen. To this end, for various values of these parameters, the predictor is designed, and its corresponding validation error is computed in order to choose the best set of parameters. For each set of parameters, a data set $\mathcal{D} = \{(\tilde{y}(k), w(k)), k \in \mathbb{I}_1^{N_\mathcal{D}}\}$ is obtained from the experimental data.

Thus, for various combinations of the memory horizons, the Hölder parameters of the system are obtained using the POKI method (see Appendix B). Using the obtained constant and exponent, the SPKI predictor can be defined as per (2.3).

The SKI filtering of equation (2.1) is done predicting with increment δ set to 0.015, and $\sigma_0 = 2/(n_w + 2)$, $\sigma_i = 1/(n_w + 2)$. For this reason, the computation time used for prediction increases linearly with the number of regression terms n_a, n_b.

Finally, the data from the other tests are used to calculate the prediction error, as the difference between the real output and the concentration estimated by the predictor, for the same input signal. As it is represented in Figure 2.6, for $n_a = 2$ and $n_b = 1$, the maximum prediction error is minimized. With these values the Hölder constant results in $L = 1.5015$, and the exponent $p = 0.86$. Figure 2.7 shows the histogram of the validation error for this configuration, whose absolute maximum is $\mu = 0.061\,\text{mol}\,\text{l}^{-1}$.

In order to reduce the computational effort, the regressor space $\mathcal{W}$ is divided into a set of overlapping regions, following the procedure presented in Section 2.2.2. Given the recursive characteristic of the NARX model, data points defined by a sequence of temporal signals (e.g. $(y(k), y(k-1))$) are highly correlated, due to the small variance of the signals between sequent time steps. For this reason, the regions $\mathcal{W}_i$ are not defined in the whole $\mathcal{W}$ space, but instead, only the current values of the output and the input (i.e. $(y(k), u(k))$) are taken into account. The vector space $(y(k), u(k))$ is divided into a grid of *squares* of 0.01 side length (this corresponds to dividing each component into $m = 100$ partitions, provided that they are both scaled between 0 and 1). Instead of using a ball for the overlapping, to each

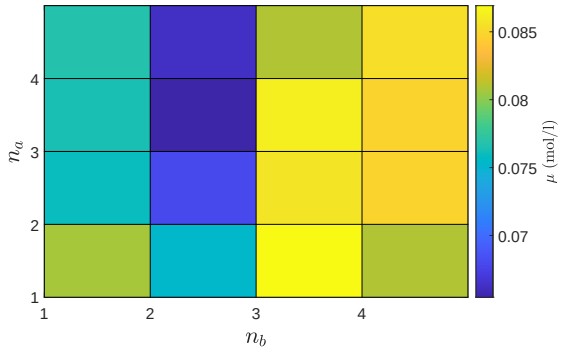


Figure 2.6: Dependence of the the prediction error w.r.t. the parameters n_a and n_b.

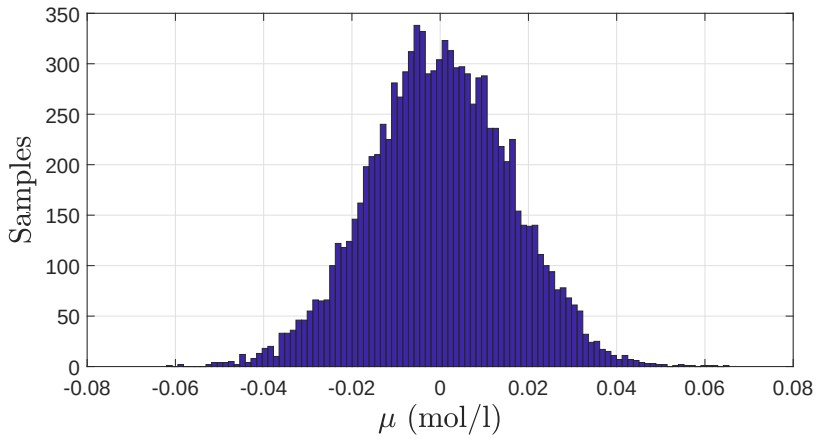


Figure 2.7: Histogram of the validation error for $n_a = 2$ and $n_b = 1$.

region, its corresponding data are obtained using also data of neighbouring regions (see Figure 2.8).

The prediction for 100 random query points in an Intel® Core™ i7-6700HQ CPU @ 2.60GHz 12GB RAM without the partitions division takes around 41.826 s, while using partition-based approach takes only 0.0717 s, for $n_a = 2$ and $n_b = 1$, with prediction difference less than $0.001\,\mathrm{mol\,l^{-1}}$.

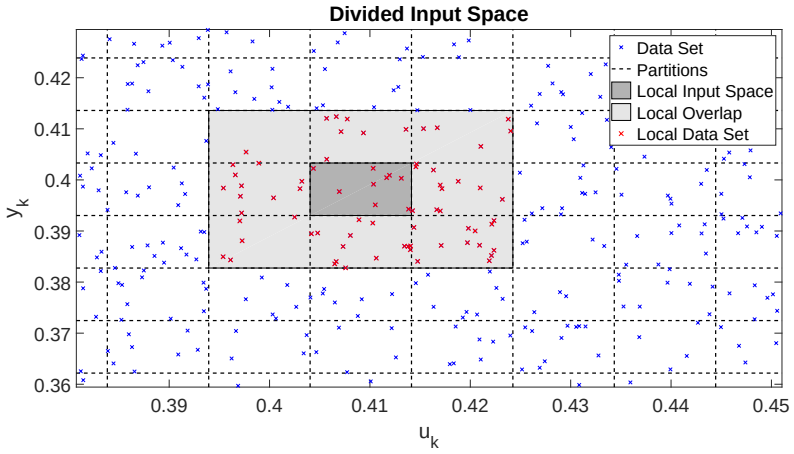


Figure 2.8: Partition of the input space $\mathcal{W}$.

Control of the reactor

The optimization problem $P_{N_c,N_p}(x(k),\mathcal{D})$ (2.9) is solved in MATLAB®, using the optimization function `fmincon`. The stage cost is defined as the sum of a quadratic cost ℓ_t and the barrier cost ℓ_b: [3]

$$\ell_t(x_k,u_k) = \|y_k - y_r\|_Q^2 + \|u_k - u_r\|_R^2 + \|u_k - u_{k-1}\|_S^2,$$

$$\ell_b(y_k) = \Lambda\left(1 - \exp\left(\frac{-\mathfrak{d}(y_k,\mathcal{Y}_{\text{soft}})}{\epsilon}\right)\right).$$

The parameters are set to $Q = 10$, $R = 5$, $S = 10$ (Remark 2.4), $\Lambda = 9999$ and $\epsilon = 3\times10^{-3}$. The terminal control law $\kappa_f(x) = K(x - x_r) + u_r$ and terminal cost $V_f(x) = \|x - x_r\|_P^2$ are considered, with $\lambda = 1$. K and P are calculated from the solution of the LQR for a linear model around the reference. This linearized model, calculated numerically from the input-output data, is the following:

$$x(k+1) = \begin{bmatrix} 0.448 & 0.299 & -0.002 & -0.011 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x(k) + \begin{bmatrix} -0.0983 \\ 0 \\ 0 \\ 1 \end{bmatrix} u(k).$$

This results in:

[3] The subindex in x_k denotes $x(k)$.

$$P = \begin{bmatrix} 183.48 & 30.828 & -0.2224 & -1.1638 \\ 30.828 & 14.994 & -0.1079 & -0.5647 \\ -0.2224 & -0.1079 & 0.0008 & 0.0041 \\ -1.1638 & -0.5647 & 0.0041 & 11.021 \end{bmatrix},$$

$$K = \begin{bmatrix} 0.4856 & 0.2278 & -0.0016 & -0.0086 \end{bmatrix}.$$

Regarding Theorem 2.1, the parameters ϕ and γ have been computed. Setting the control horizon to $N_c = 4$ and the prediction horizon to $N_p = 14$ results in $\phi = 1.2046$ and $\gamma = 0.0294$. If $x(0) = [0.6\ 300\ 300]^T$, then the optimal cost is $V^*_{N_c,N_p}(x(0)) = 13.721$, so $x(0) \in X_{N_c,N_p}(\lambda)$. Regarding Theorem 2.2, as far as we analysed the outcomes of the predictor, Assumption 2.2 holds for every query point observed, making $\ell(\cdot,\cdot)$, $\kappa_f(\cdot)$ and $V_f(\cdot)$ uniformly continuous.

Once the controller is designed, a hundred simulations are carried out, with the same initial state, but a hundred different realizations of the 2.5% random noise. The following figures represent the output and the control input. Within each subplot, the constraints appear in black, the reference in red, and the 100 trajectories are represented by the gray band, whose mean value is plotted in blue.

A first experiment is carried out in which the model used by the MPC is the real plant (defined by its set of ODEs (2.18)). The simulations provide a measure of the best closed-loop behaviour, achievable assuming perfect knowledge of the model. This result is shown in Figure 2.9.

Secondly, an MPC controller based on the standard KI prediction is applied, using the data set $\mathcal{D}$. Notice that neither the data set was divided into partitions, nor the filtered KI was implemented. The result of this simulation is shown in Figure 2.10. These results are not satisfactory, since the solver is failing to converge due to the fact that the KI prediction may be non-differentiable or due to the large number of iterations.

Finally, the SPKI-MPC is applied, to address the issues mentioned above. The result is shown in Figure 2.11. The behaviour is smoother, with a smaller band than the previous experiment, which indicates greater robustness and a performance closer to the ideal one (Figure 2.9) than the simpler KI set.

Apart from that, in order to compare the performance of the presented results, a hundred different simulations of the MPC (with random initial state) are carried out, whose results are compared in Figure 2.12. The boxplots represent maximum, minimum, median and deviation of the performance costs and computational times of the experiments. The performance index of each

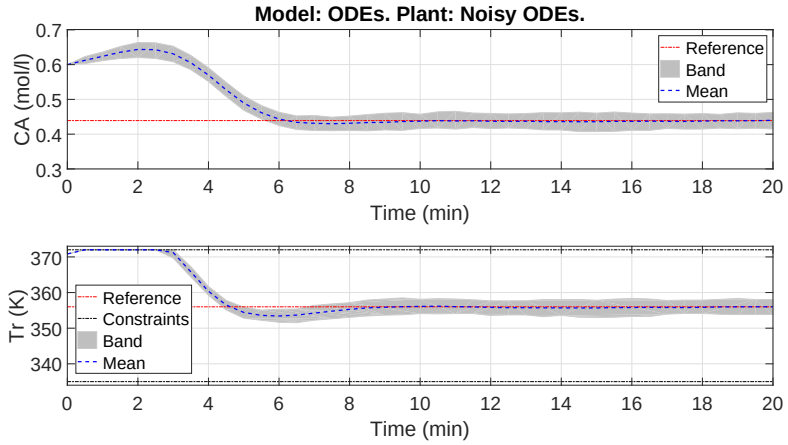


Figure 2.9: Ideal ODEs-MPC.

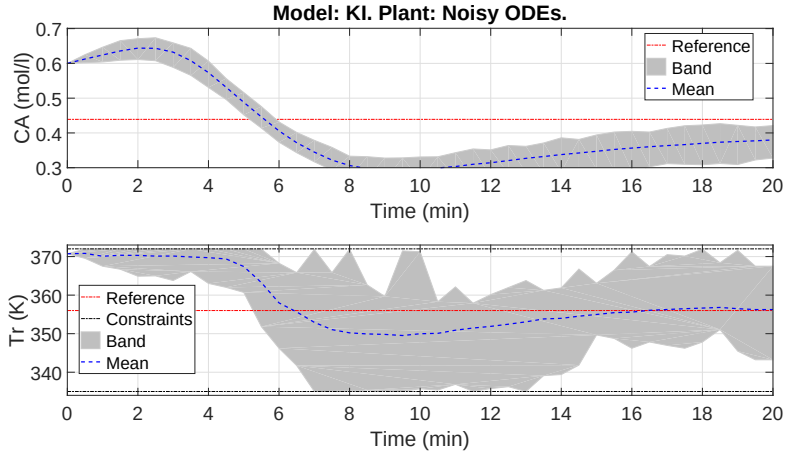


Figure 2.10: Primary KI-MPC.

simulation is defined as:

$$\Phi = \sum_{i=1}^{t_{\text{sim}}} \ell_t(x(i), u(i)) + \ell_b(y(i)).$$

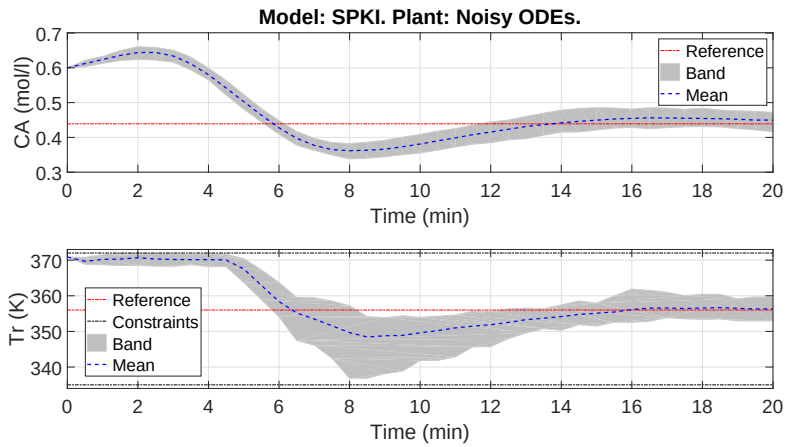
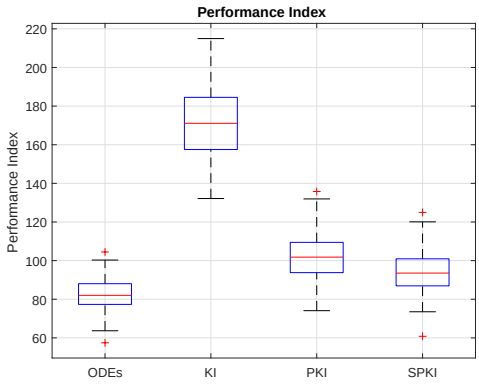
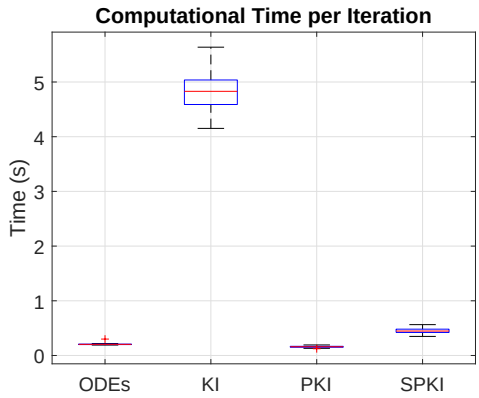


Figure 2.11: Data-based SPKI predictive control applied to a CSTR.



(a) Performance



(b) Computational time

Figure 2.12: Boxplot comparison of the different prediction models for 100 simulations of the MPC.

Chapter 3

Robust learning-based MPC for nonlinear constrained systems

3.1 Introduction

Robust constraint satisfaction is particularly important for safe operation of systems, and challenging for learning-based controllers, as they have to consider not only the possible perturbations of the system, but also the uncertainty introduced by the inference method. The main limitation of the controller presented in the previous chapter is that it cannot guarantee robust satisfaction of output constraints.

In this chapter, the main contribution is a new robust predictive controller formulation for systems also subject to hard constraints on the outputs. A design method that takes into account the prediction error bounds in an explicit way to tighten the problem constraints and that guarantees closed-loop constraint satisfaction and input-to-state stability (ISS) [81] is provided. One of the main characteristics of this design is that it is not based on a terminal region constraint. In general, this terminal constraint is based on robust invariant sets which, for the class of systems considered (that is, unknown systems, possibly nonlinear, for which a priori only input-output data are available), are difficult to obtain.

Given the hard feature of the constraints, inference methods that provide a guaranteed bound on the prediction error must be used. Besides, knowledge of the Lipschitz constant allows the estimation of the reachable sets, in order to account for a robust design of the controller. For these two reasons, kinky inference is an useful method to address this problem. The procedure

to define the tightened constraints of the MPC optimization problem is provided. This procedure is specifically tailored to the KI model used, in order to obtain the least conservative possible bounds.

The results in this chapter were published in [98, 102].

3.2 Problem setting

The problem setting is inherited from Chapter 2, with the additional hard constraint in the output, that is,

$$y(k) \in \mathcal{Y}, \forall k, \tag{3.1}$$

where $\mathcal{Y} \subset \mathbb{R}^{n_y}$ is a compact set.

MPC requires repeated optimization of the predicted control inputs subject to constraints. Therefore, in order to give guarantees on the controller's closed-loop performance, recursive feasibility and constraint satisfaction must be ensured. That is, it is necessary to ensure that all constraints remain satisfiable during runtime, or equivalently, to guarantee that the controlled system will not leave the feasibility region. However, since the controller will not be based on the ground-truth dynamics f, but on the learnt model $\hat{\mathfrak{f}}$ inferred from samples of the ground-truth, recursive feasibility can only be guaranteed if a bound on the discrepancy between f and $\hat{\mathfrak{f}}$ is known a priori, and taken into account by the controller.

The LACKI method explained in Appendix B ensures that if the model function is Hölder and the noise is bounded, then the estimation error is bounded, which is required to design a deterministic robust controller to regulate the plant. Any worst-case guarantee inevitably requires a priori knowledge. Hence, in the following hypothesis, it is assumed that the maximum prediction error μ is available for the design of the controller.

Assumption 3.1. *It is assumed that for L, p, and $\mathcal{D}$, a bound on the error between the estimated output and the real output is known, denoted $\mu \in \mathbb{R}^{n_y}$, such that*

$$|\hat{\mathfrak{f}}_i(x, u; \theta, \mathcal{D}) - f_i(x, u)| \leq \mu_i, \tag{3.2}$$

for all $i \in \mathbb{I}_1^{n_y}$, $x \in \mathcal{Y}^{n_a+1} \times \mathcal{U}^{n_b}$, and $u \in \mathcal{U}$.

Remark 3.1. *From a practical point of view, the problem of how to calculate the error bound must be addressed. Kinky inference methods enjoy the property of providing a deterministic error bound if the Lipschitz constant and an*

upper bound of the noise are known (see Appendix B). Moreover, if a bound on the second derivative is known, it is also possible to derive an estimation error bound. If these parameters are not known a priori, which is usual in practice, then they must be estimated from experimental data. Consequently, the validity of the results presented in this section is conditioned to the validity of the estimated error bound. This is the reason why this is considered as a standing assumption.

3.3 Stabilizing data-based NMPC

In this section, a model predictive controller is derived based on a prediction model learnt from data of the plant. Since the prediction model is not accurate, the effect of the estimation error on the predictions must be analysed, to be taken into account in the design of the controller. For this analysis, it is convenient to define the NARX model of the plant in a state–space form, which is done as in Chapter 2, in equations (2.8) (on page 33).

Then, the proposed robust MPC is based on nominal predictions and tightened constraints. To guarantee robustness, a bound on the propagation of the prediction error (see Figure 3.1) is calculated from the following lemma:

Lemma 3.1. *Assume that at sampling time k, the state of the plant is $x(k)$ and a sequence of future control inputs $u(k+j)$ for $j \in \mathbb{I}_0^{N-1}$ is given. Let $\hat{x}(j|k)$ and $\hat{y}(j|k)$ be the predicted states and outputs, respectively, derived from (2.8a) for the given sequence of future control inputs and the current state $x(k)$, i.e. $\hat{x}(0|k) = x(k)$.*

Assume that at sampling time $k+1$, the current output $y(k+1)$ is measured, and hence the current state $x(k+1)$ is known. Based on these new measurements, an updated sequence of states and outputs $\hat{x}(j|k+1)$ and $\hat{y}(j|k+1)$ is predicted based on (2.8a) with $\hat{x}(0|k+1) = x(k+1)$ and the remaining sequence of the given future control inputs.

Let $c_1 \in \mathbb{R}^{n_y}$ be a vector such that

$$|y(k+1) - \hat{y}(1|k)| \leq c_1. \tag{3.3}$$

Then, the mismatch between the predictions satisfies [1]

$$|\hat{y}(j-1|k+1) - \hat{y}(j|k)| \leq c_j,\ j \in \mathbb{I}_1^N, \tag{3.4a}$$
$$\|\hat{x}(j-1|k+1) - \hat{x}(j|k)\|_{\mathcal{X}} \leq r_j,\ j \in \mathbb{I}_1^N, \tag{3.4b}$$

[1] $\|\cdot\|_{\mathcal{X}}$ is a norm for the state–space such that $\|x\|_{\mathcal{X}} = \|(x,0)\|_{\mathcal{W}}$.

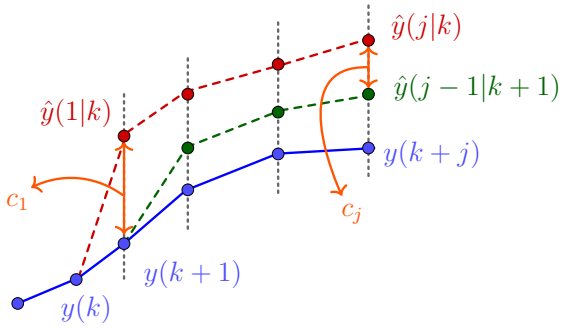


Figure 3.1: Propagation of the prediction error.

where $c_j \in \mathbb{R}^{n_y}$ *and* $r_j \in \mathbb{R}$ *are obtained from the recursion*

$$c_{j+1,i} = L_i r_j^{p_i}, \tag{3.5}$$

and $r_j = \|\Xi_j\|_{\mathcal{X}}$, $j \in \mathbb{I}_1^{N-1}$, $i \in \mathbb{I}_1^{n_y}$, *where*

$$\Xi_j = \mathcal{B}(c_j) \times \cdots \times \mathcal{B}(c_{\sigma(j)}) \times \underbrace{\{0\} \times \cdots \times \{0\}}_{n_b+1-\sigma(j-1) \text{ times}} \subseteq \mathbb{R}^{n_x},$$

with $\sigma(j) = \max(1, j - n_a)$.

Proof.

Provided that

$$\hat{y}(j-1|k+1) = \hat{\mathfrak{f}}(x(j-2|k+1, u(k+j-1))$$

and $\hat{y}(j|k)) = \hat{\mathfrak{f}}(\hat{x}(j-1|k), u(k+j-1))$, it can be derived that $\forall i \in \mathbb{I}_1^{n_y}$

$$|\hat{y}_i(j-1|k+1) - \hat{y}_i(j|k))| \leq L_i \|\hat{x}(j-2|k+1) - \hat{x}(j-1|k)\|_{\mathcal{X}}^{p_i}.$$

Note that

$$\begin{aligned}\hat{x}(j-2|k+1) - \hat{x}(j-1|k) = \quad & \big[\hat{y}(j-2|k+1) - \hat{y}(j-1|k), \ldots, \\ & \hat{y}(\sigma(j-1)-1|k+1) - \hat{y}(\sigma(j-1)|k), \\ & 0, \ldots, 0\big].\end{aligned}$$

Then, $\hat{x}(j-2|k+1) - \hat{x}(j-1|k) \in \Xi_{j-1}$. Assuming that c_{j-1} is known,

$$\|\hat{x}(j-2|k+1) - \hat{x}(j-1|k)\|_{\mathcal{X}} \leq \|\Xi_{j-1}\|_{\mathcal{X}} = r_{j-1},$$

which implies the stated result. □

Remark 3.2. *Note that the formulation of the previous Lemma and in what is to follow treats each of the n_y outputs of f as different functions, considering different L_i, p_i for each $i \in \mathbb{I}_1^{n_y}$.*

Remark 3.3. *If the infinity norm is chosen as the norm of the input space, then*

$$r_j = \max_{s \in \mathbb{I}_{\sigma(j)}^{j}} \|c_s\|_\infty.$$

Based on the derived bounds on the prediction error, the problem of robust constraint satisfaction is addressed, by means of a set of tightened constraints on the outputs [133], computed offline for the maximum possible prediction error, i.e. taking $c_1 = \mu$. These sets are defined as follows:

$$\mathcal{Y}_j = \mathcal{Y} \ominus \mathcal{B}(d_j), \tag{3.6}$$

where d_j, known as *back-off* in MPC literature, is obtained as

$$d_j = \sum_{s=1}^{j} c_s. \tag{3.7}$$

These constraints sets will be used to prove recursive feasibility of the controller, following standard procedures.

Lemma 3.2. *The sets $\mathcal{Y}_j$ are such that for all $y \in \mathcal{Y}_j$ and for all deviation $\Delta y \in \mathcal{B}(c_j)$, $y + \Delta y \in \mathcal{Y}_{j-1}$.*

Proof.

Since for $j \geq 1$, $d_j = d_{j-1} + c_j$, it follows that

$$\mathcal{B}(d_j) = \mathcal{B}(d_{j-1}) \oplus \mathcal{B}(c_j).$$

By definition,

$$y + \Delta y \in \mathcal{Y}_j \oplus \mathcal{B}(c_j) = \mathcal{Y} \ominus \mathcal{B}(d_j) \oplus \mathcal{B}(c_j),$$

and hence $\mathcal{Y}_j = \mathcal{Y} \ominus \mathcal{B}(d_j) = \mathcal{Y} \ominus \mathcal{B}(d_{j-1}) \ominus \mathcal{B}(c_j)$, so

$$\begin{aligned} y + \Delta y &\in \mathcal{Y}_j \oplus \mathcal{B}(c_j) \\ &= \mathcal{Y} \ominus \mathcal{B}(d_{j-1}) \ominus \mathcal{B}(c_j) \oplus \mathcal{B}(c_j) \\ &\subseteq \mathcal{Y} \ominus \mathcal{B}(d_{j-1}) = \mathcal{Y}_{j-1}. \end{aligned}$$

□

In order to ensure that the proposed controller is feasible, the tightened set of constraints must be non-empty along the prediction horizon, as stated in the following assumption.

Assumption 3.2. *The prediction horizon N and estimation error bound μ are such that the set $\mathcal{Y}_N$ is non-empty.*

Based on the previous definitions, the optimization problem $P_N(x(k); \mathcal{D})$ of the proposed predictive controller is:

$$\begin{aligned}
\min_{\mathbf{u}} \quad & V_N(x(k), \mathbf{u}) \\
& = \sum_{i=0}^{N-1} \ell(\hat{x}(i|k), u(k+i)) + \lambda V_f(\hat{x}(N|k)) && \text{(3.8a)} \\
\text{s.t.} \quad & \hat{x}(0|k) = x(k) && \text{(3.8b)} \\
& \hat{x}(j+1|k) = \hat{F}(\hat{x}(j|k), u(k+j)), \quad j \in \mathbb{I}_0^{N-1} && \text{(3.8c)} \\
& \hat{y}(j|k) = M\hat{x}(j|k), \quad j \in \mathbb{I}_0^{N-1} && \text{(3.8d)} \\
& u(k+j) \in \mathcal{U}, \quad j \in \mathbb{I}_0^{N-1} && \text{(3.8e)} \\
& \hat{y}(j|k) \in \mathcal{Y}_j, \quad j \in \mathbb{I}_0^{N-1}, && \text{(3.8f)}
\end{aligned}$$

where $\lambda \geq 1$ is a weighting parameter and $\hat{F}$ and M define the state-space (see equations (2.8)). Note that this problem is nonlinear, non-convex and non-differentiable. Its ingredients are required to meet the following assumption, which is similar to the standard MPC ones [133]:

Assumption 3.3.

1. *The stage cost function $\ell(\cdot, \cdot)$ is a Hölder continuous positive definite function such that $\ell(\cdot, \cdot) \geq \alpha_y(\|x\|_{\mathcal{X}})$ for a certain $\mathcal{K}$-function α_y, and its Hölder parameters are L_x and p_x.*

2. *There exists a control law $u = \kappa_f(x)$, a function V_f and a level set*

$$\Omega_\gamma = \{x : V_f(x) \leq \gamma\} \subseteq \mathbb{R}^{n_x}$$

for some $\gamma > 0$ such that for all $x \in \Omega_\gamma$ the following conditions hold:

   (a) *V_f is a Hölder continuous positive definite function, with Hölder constants L_{V_f}, p_{V_f}. Given two $\mathcal{K}$-functions α_f, β_f,*

$$\begin{aligned}
\alpha_f(\|x\|_{\mathcal{X}}) \leq V_f(x) &\leq \beta_f(\|x\|_{\mathcal{X}}), \\
V_f(\hat{F}(x, \kappa_f(x))) - V_f(x) &\leq -\ell(x, \kappa_f(x)).
\end{aligned}$$

(b) $\kappa_f(x) \in \mathcal{U}$, $Mx \in \mathcal{Y}_N$.

The controller is derived from the receding horizon solution of (3.8). It follows a standard robust approach in which the cost of the nominal predictions is minimized, while taking into account a tightened set of constraints to guarantee recursive feasibility. The main difference with off-the-shelf robust ISS formulations for nonlinear systems [81] is that in these, either there are no constraints on the states in the optimization problem, or a terminal constraint, based on a certain robust positive invariant set, is added. In this controller, although a terminal cost (based on a local controller for the nominal model) is taken into account in the cost function, no terminal constraint is included. Thus, its design is notably simplified since the calculation of a robust invariant set is avoided, which was a hard task, as shown in [51]. In this case, the calculation could have been even more difficult, provided the lack of an explicit expression of the model of the system.

Furthermore, an additional tuning parameter λ is added, modifying the weight of the terminal cost in the objective function. It is proven that this controller guarantees that the closed-loop systems is ISS in an explicitly defined region of the state space, which is enlarged by this weight.

Define the function

$$\nu(c_1) = \sum_{j=1}^{N} L_x r_j^{p_x} + \lambda L_{V_f} r_{N+1}^{p_{V_f}}, \tag{3.9}$$

where r_j is defined in Lemma 3.1 for c_1, and $L_x, p_x, L_{V_f}, p_{V_f}$ in Assumption. 3.3.

Assumption 3.4. *The bound μ is such that the set $\Upsilon = \{x : \ell(x,0) \leq \nu(\mu)\}$ is contained in Ω_γ. The positive constants λ and ϕ are such that $\lambda \geq 1$ and $\ell(x,0) > \phi$ for all $x \notin \Omega_\gamma$.*

Remark 3.4. *In a general setting, a condition to check if the level set Υ is contained in Ω_γ could be derived using the supply $\mathcal{K}_\infty$-functions that bound the cost functions given in Assumption 3.3. In this case the condition would be [79]:*

$$\nu(\mu) \leq \alpha_y(\beta_f^{-1}(\gamma)). \tag{3.10}$$

Another method could be using probabilistic validation by means of randomized algorithms [32].

Lemma 3.3. *Under Assumption 3.4, $\phi \geq \nu(\mu)$.*

Proof.

Since $\Upsilon \subseteq \Omega_\gamma$, the constant ϕ can be taken as

$$\phi \geq \min_{x \in \mathcal{X} \setminus \Omega_\gamma} \ell(x, 0) \geq \min_{x \in \mathcal{X} \setminus \Upsilon} \ell(x, 0) \geq \nu(\mu).$$

□

Let Γ define the following level set of the optimal cost function

$$\Gamma = \{x : V_N^*(x) \leq N\phi + \lambda\gamma\}. \tag{3.11}$$

Notice that this set is compact and non-empty. It is next stated that this set defines the region in which ISS is guaranteed.

Theorem 3.1 (ISS stability)**.** *Suppose that Assumptions 3.1, 3.2, 3.3 and 3.4 hold for the optimization problem P_N (3.8). Let $\kappa_N(x)$ be the control law derived from the solution of $P_N(x; \mathcal{D})$ applied using a receding horizon policy. Then, for any initial $x(0) \in \Gamma$, the system controlled by the control law $u(k) = \kappa_N(x(k))$ is input-to-state stable w.r.t. the estimation error; and the constraints are always satisfied, i.e. $u(k) \in \mathcal{U}$, $y(k) \in \mathcal{Y}$ and $x(k) \in \Gamma, \forall k$.*

Proof.

Assume that $x(k) \in \Gamma$. Then, it can be shown that $x^*(N|k) \in \Omega_\gamma$ [83]. Define the shifted sequence as $\bar{\mathbf{u}}(k+1)$ such that $\bar{u}(j|k+1) = u^*(j+1|k)$ for $j \in \mathbb{I}_0^{N-2}$ and $\bar{u}(N-1|k+1) = \kappa_f(x^*(N|k))$.

Recursive feasibility: Assuming that $x(k) \in \Gamma$, it will be proven that also $x(k+1) \in \Gamma$. In that case, since Γ is a subset of the feasibility region of the optimization problem $P_N(x; \mathcal{D})$, the system is recursively feasible.

Firstly, it will be shown that the solution $\bar{\mathbf{u}}(k+1)$ is a feasible solution for $x(k+1)$. Given that $x^*(N|k) \in \Omega_\gamma$, from the feasibility of $\mathbf{u}^*(k)$ and Assumption 3.3, it is immediate to state that $\bar{u}(j|k+1) \in \mathcal{U}$ for all $j \in \mathbb{I}_0^{N-1}$.

From Lemma 3.1, $\hat{y}(j|k+1) - y^*(j+1|k) \in \mathcal{B}(c_{j+1})$, $\forall j \in \mathbb{I}_0^{N-1}$. Besides, from the feasibility of $\mathbf{u}^*(k)$, it is obtained that $y^*(s|k) \in \mathcal{Y}_s$ for $s \in \mathbb{I}_0^{N-1}$ and $x^*(N|k) \in \Omega_\lambda$, which implies that $y^*(N|k) = Mx^*(N|k) \in \mathcal{Y}_N$ in virtue of Assumption 3.3.2b. Then, from Lemma 3.2, for all $j \in \mathbb{I}_0^{N-1}$,

$$\hat{y}(j|k+1) \in \mathcal{Y}_{j+1} \oplus \mathcal{B}(c_{j+1}) \subseteq \mathcal{Y}_j.$$

Therefore the problem $P_N(x(k+1); \mathcal{D})$ is feasible.

Next, it will be proven that $x(k+1) \in \Gamma$. Since $x(k) \in \Gamma$, it can be stated that $V_N^*(x(k)) \leq N\phi + \lambda\gamma$. Following standard arguments in MPC [133] it can be proven that

$$\begin{aligned} V_N(x^*(1|k), \bar{\mathbf{u}}(k+1)) &\leq V_N^*(x(k)) - \ell(x(k), u(k)) \\ &\leq N\phi + \lambda\gamma - \ell(x(k), u(k)). \end{aligned}$$

On the other hand, given $\tilde{x} = \hat{F}(x^*(N|k), \kappa_f(x^*(N|k))$,

$$\begin{aligned} V_N(x(k+1),&\bar{\mathbf{u}}(k+1)) - V_N(x^*(1|k), \bar{\mathbf{u}}(k+1)) = \\ &\sum_{i=0}^{N-1} \Big(\ell(\hat{x}(i|k+1), \bar{u}(k+i+1)) - \ell(x^*(i+1|k), \bar{u}(k+i+1))\Big) \\ &+ \lambda\Big(V_f(\hat{x}(N|k+1)) - V_f(\tilde{x})\Big). \end{aligned}$$

According to Lemma 3.1, it can be derived the upper bound on the norm for $s \in \mathbb{I}_1^N$, $\|\hat{x}(s-1|k+1) - x(s|k)\|_{\mathcal{X}} \leq r_s$, with r_s obtained for a given c_1. Then, for $j \in \mathbb{I}_1^N$ and given c_1 satisfying (3.3),

$$\ell(\hat{x}(j-1|k+1), \bar{u}(k+j)) - \ell(x^*(j|k), \bar{u}(k+j)) \leq L_x r_j^{p_x},$$

and $V_f(\hat{x}(N|k+1)) - V_f(\tilde{x}) \leq L_{V_f} r_{N+1}^{p_{V_f}}$.

Therefore,

$$V_N(x(k+1), \bar{\mathbf{u}}(k+1)) - V_N(x^*(1|k), \bar{\mathbf{u}}(k+1)) \leq \nu(c_1). \tag{3.12}$$

To prove robust invariance the worst possible case has to be considered, for which $c_1 = \mu$ is taken. Hence, it has been proven that

$$V_N(x(k+1), \bar{\mathbf{u}}(k+1)) \leq \nu(\mu) + N\phi + \lambda\gamma - \ell(x(k), u(k)).$$

Consider the case where $x(k) \in \Gamma \setminus \Upsilon$. Then $\ell(x(k), u(k)) > \nu(\mu)$. Hence, $V_N(x(k+1), \bar{\mathbf{u}}(k+1)) \leq \nu(\mu) + N\phi + \lambda\gamma - \ell(x(k), u(k)) \leq N\phi + \lambda\gamma$.

Given that $V_N^*(x(k+1)) \leq V_N(x(k+1), \bar{\mathbf{u}}(k+1))$, then $x(k+1) \in \Gamma$.

Consider now the case that $x(k) \in \Upsilon$. Since $\Upsilon \subseteq \Omega_\gamma$, $x(k) \in \Omega_\gamma$. With standard arguments in MPC [133], it can be shown that the optimal cost $V_N^*(x(k)) \leq \lambda V_f(x(k)) \leq \lambda\gamma$. Hence,

$$\begin{aligned} V_N(x(k+1), \bar{\mathbf{u}}(k+1)) &\leq \nu(\mu) + V_N^*(x(k)) - \ell(x(k), u(k)) \\ &\leq \nu(\mu) + \lambda\gamma - \ell(x(k), u(k)), \end{aligned}$$

since $\nu(\mu) \leq \phi$, $V_N(x(k+1), \bar{\mathbf{u}}(k+1)) \leq N\phi + \lambda\gamma$. Thus, $x(k+1) \in \Gamma$.

Input-to-state stability: Equation (3.12) can be rewritten as follows, taking $c_1 = d(k+1) := |y(k+1) - \hat{y}(1|k)|$:

$$V_N(x(k+1), \bar{\mathbf{u}}(k+1)) - V_N(x^*(1|k), \bar{\mathbf{u}}(k+1)) \leq \nu(d(k+1)). \tag{3.13}$$

Then, following the previous steps, it can be derived that

$$\begin{aligned} V_N^*(x+1) &\leq V_N(x(k+1), \bar{\mathbf{u}}(k+1)) \\ &\leq \nu(d(k+1)) + V_N^*(x(k)) - \ell(x(k), u(k)). \end{aligned} \tag{3.14}$$

Thus, $V_N^*(x)$ is an ISS Lyapunov function [81] w.r.t. d.

□

Remark 3.5 (Suboptimal case)**.** *The stability analysis can be extended to the case in which the optimal solution of the control problem is not found. Given an initial feasible solution of the control problem, if the optimizer is able to improve the cost (even for a suboptimal solution of the problem), then the controller is able to maintain robust stability while satisfying the constraints [133].*

Remark 3.6 (Violation of Assumption 3.1)**.** *If the bound on the prediction error is estimated from data (e.g. via cross-validation), then the error could take a value larger than μ for a certain period of time. In this case, the ISS property (3.14) still holds as long as $x(k) \in \Gamma$ for that period of time. Notice that the ISS condition is derived from the smoothness of the optimal cost function, which is an inherent property of the proposed optimal control problem.*

Corollary 3.1. *If the obtained bound on the prediction error μ holds with confidence ρ, that is,*

$$\mathbb{P}(|\hat{\mathfrak{f}}(x, u; \theta, \mathcal{D}) - f(x, u)| \leq \mu) \geq \rho, \tag{3.15}$$

then Theorem 3.1 holds with probability greater or equal than ρ.

Proof.

Denote T the event satisfying Theorem 3.1, and denote A the event satisfying Assumption 3.1. Theorem 3.1 proves that if Assumption 3.1 is met, then the system is ISS. Therefore, $\mathbb{P}(T|A) = 1$. Besides, suppose that the probability of an underestimation of μ is bounded by $1-\rho$. Hence, $\mathbb{P}(A) \geq \rho$.

Additionally, as stated in Remark 3.6, there may exist cases in which Theorem 4.3 holds even if Assumption 3.1 is not met. Thus, $\mathbb{P}(A|T) \leq 1$. Applying Bayes' theorem

$$\mathbb{P}(T) = \frac{\mathbb{P}(T|A)\mathbb{P}(A)}{\mathbb{P}(A|T)} \geq \rho. \tag{3.16}$$

□

Remark 3.7 (Stability margin)**.** *Most of the robust controllers for constrained systems exhibit an upper bound on the estimation error to be solvable [81]. This is the so-called stability margin. One of the main drawbacks of robust predictive controllers is that this margin is typically quite conservative, due to the open-loop nature of the predictions. As a robust controller, our approach inherits this drawback.*

3.4 Case study

In this section, the proposed controller is applied to the continuously-stirred tank reactor that was presented in Chapter 2 (on page 42). However, a different configuration is used. The set of ordinary differential equations that govern the plant are those in (2.18), although the parameters change, as presented in Table 3.1. The constraints are given by $0.38 \leq C_A \leq 0.954\,\mathrm{mol\,l^{-1}}$ and $280 \leq T_r \leq 310\,\mathrm{K}$. The reference equilibrium point is $C_A^{\mathrm{ref}} = 0.62\,\mathrm{mol\,l^{-1}}$ and $T_r^{\mathrm{ref}} = 304.5\,\mathrm{K}$.

In order to identify the system, several data sets are generated. The training data set is obtained using the following input sequence of chirp signals: five chirp signals of length 1000 min, initial and final frequencies of 1 mHz and 0.15 Hz (respectively), amplitude 5 K, and centres starting from 285 K with 5 K interval; followed by two chirp signals of length 5000 min, centered in 295 K, 15 K of amplitude, and initial and final frequencies of 10 and 200 mHz and 1 and 90 mHz, respectively, as represented in Figure 3.2.

Another data set is obtained to calculate μ via cross-validation, applying a pseudorandom input sequence, where T_r switches randomly between 280 and 310 K with random switching periods between 12 and 75 min.

Following standard cross-validation procedures [86], these data sets are used to define a predictor for different values of the memory horizons n_a and n_b. Setting the prediction horizon to $N = 4$ and applying LACKI (see Appendix B) to calculate L while fixing $p = 1$, the optimal n_a, n_b are obtained

Param.	Definition	Value	Units
q_0	Input flow of the reactive	10	$\mathrm{l\,min^{-1}}$
V	Liquid volume in the tank	150	l
k_0	Frequency constant	7.2×10^{10}	$\mathrm{min^{-1}}$
E/R	Arrhenius constant	8750	K
$-\Delta H_r$	Enthalpy of the reaction	50000	$\mathrm{J\,mol^{-1}}$
UA	Heat transfer coefficient	50000	$\frac{\mathrm{J}}{\mathrm{min\,K}}$
ϱ	Density	1000	$\mathrm{g\,l^{-1}}$
C_p	Specific heat	0.239	$\frac{\mathrm{J}}{\mathrm{g\,K}}$
τ	Time constant	1.5	min
C_{Af}	C_A in the input flow	1	$\mathrm{mol\,l^{-1}}$
T_f	Temperature (input flow)	370	K

Table 3.1: Parameters of the system

minimizing d_4, which is calculated using (3.7). The minimum is obtained for $n_a = n_b = 2$, for which $L = 1.38$, $\mu = 0.032\,\mathrm{mol\,l^{-1}}$ and $d_4 = 0.22\,\mathrm{mol\,l^{-1}}$.

Both the stage and the terminal cost of the MPC are defined as follows, with $x^{\mathrm{ref}} = (y^{\mathrm{ref}}, \ldots, y^{\mathrm{ref}}, u^{\mathrm{ref}}, \ldots, u^{\mathrm{ref}})$:

$$\ell(x, u) = \|x - x^{\mathrm{ref}}\|_Q^2 + \|u - u^{\mathrm{ref}}\|_R^2, \tag{3.17a}$$

$$V_f(x) = \|x - x^{\mathrm{ref}}\|_P^2. \tag{3.17b}$$

The parameter Q is set to $100I_5$, $R = 0.1$, and $\lambda = 10$. Using the model with $n_a = n_b = 2$ the terminal cost is obtained solving a LQR for the linearised model around the reference point. To ensure robust stability (Theorem 3.1), Assumptions 3.1, 3.2, 3.3 and 3.4 must hold true. The prediction error was obtained via cross-validation. The value of d_N results in $\mathcal{Y}_N = \{y : 0.60 \leq y \leq 0.73\}$. Following the procedure in Section 3.3 and in Chapter 2 results in $\gamma = 16796$, $\nu(\mu) = 53.553$ and $\phi = 9.2189 \times 10^5$, which satisfy all the assumptions.

The proposed controller is applied in 100 closed-loop simulations, subject to random noise. The results are shown in the last row of Figure 3.3. Note that the output is steered to the reference while the constraints are satisfied. The optimization problem is solved in MATLAB® on an Intel® Core™ i7-6700HQ CPU @ 2.60 GHz 12GB RAM, and each iteration takes less than one second to compute, much shorter than the sampling time of 30 s.

In order to compare the proposed controller to other methods, the same setup is simulated with two different MPCs. First, a controller derived

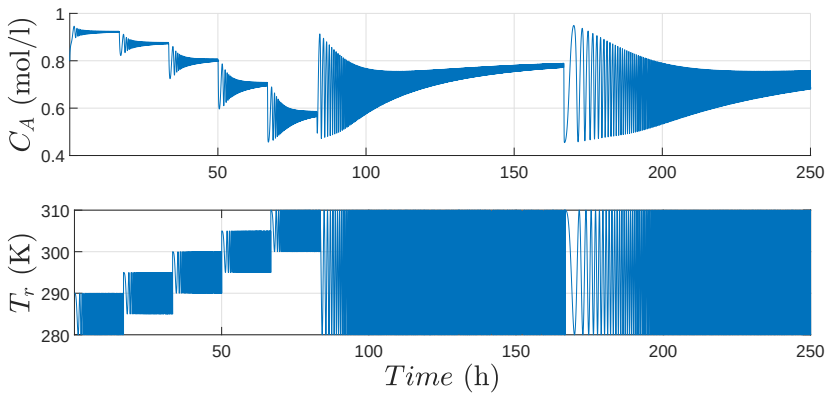


Figure 3.2: Simulation obtained applying a sequence of chirp signals.

from (3.8), but with the set of ODEs (2.18) as the state-feedback prediction model, for which μ is the maximum noise, $0.025\,\mathrm{mol\,l^{-1}}$. This aims to resemble the ideal case of perfect knowledge of the plant. The result is shown in the first row of Figure 3.3. As expected, the data-based controller performs slower than the ideal since, unlike the latter, an output-feedback uncertain framework is considered.

Second, the controller proposed in Chapter 2 is applied. It is based on KI and guarantees closed-loop stability, but does not take into account output constraints, so as shown in the second row of Fig. 3.3, the minimum C_A limit is violated. To sum up, the controller proposed in this chapter is able to robustly satisfy hard constraints in the outputs, learning the model from input-output data with a closed-loop performance similar to the ideal case.

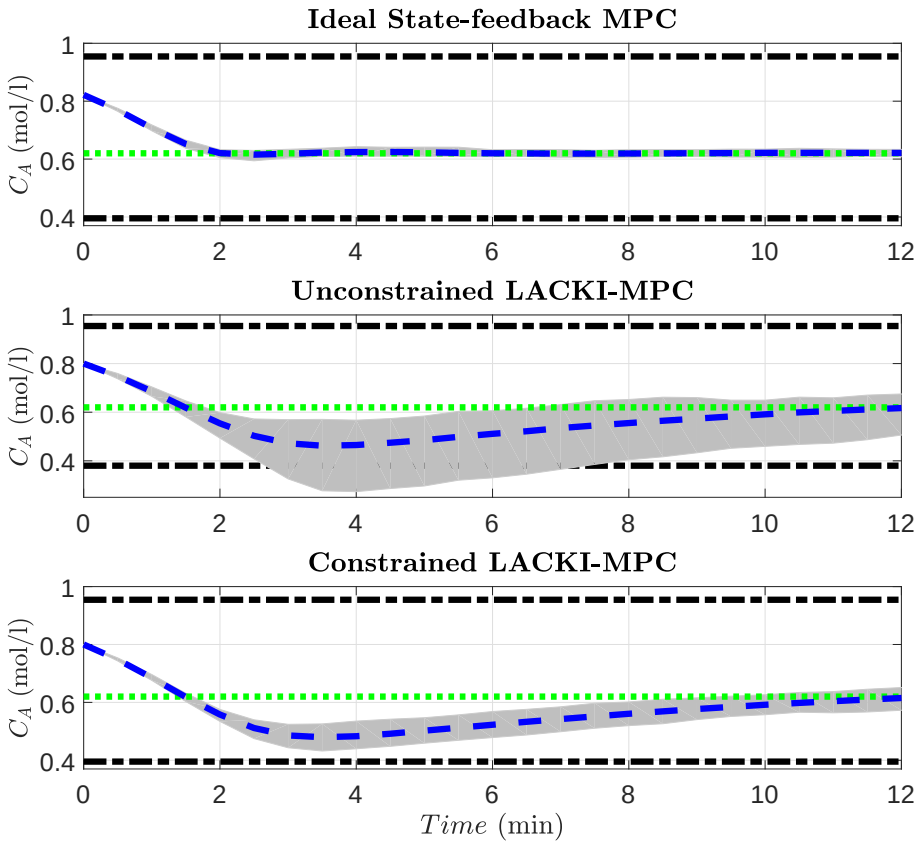


Figure 3.3: Closed-loop output for 100 simulations of the ideal MPC (top), the robust learning MPC for systems without output constraints (middle) and the proposed robust constrained learning MPC (bottom). The grey band groups the trajectories, the blue dashed line represents its mean, the green dotted one the reference and the black dash-dotted one the constraints.

Chapter 4

Componentwise Hölder inference for robust learning-based MPC

4.1 Introduction

The objective of this chapter is to introduce a novel learning methodology that extends and improves KI, able to significantly reduce the prediction error. To this end, an extended version of continuity of functions is used, namely *componentwise Hölder continuity*. Then, leveraging on this notion, the KI is extended to exploit this property, yielding the *componentwise Hölder kinky inference* (CHoKI) predictor. This improved inference method could be used in the two MPCs proposed in chapters 2 and 3.

In Hölder continuity, the Hölder constant describes the aggregated contribution of the variation of the inputs on the outputs. The idea of componentwise Hölder continuity is to disaggregate this contribution, considering an individual constant and exponent for the effect of each input on each output. These parameters can be described by a gain matrix and an exponent matrix. Hence, componentwise Hölder continuity can be regarded as if the scalar Hölder constant and exponent were extended to matrices. It is first proven that the proposed learning method can estimate any Hölder continuous function, by proving that componentwise Hölder continuity over a compact domain is equivalent to Hölder continuity.

Besides, a non-parametric estimation method and the procedure to calculate its hyper-parameters from measured (possibly noisy) input-output data

is presented. It is also shown that the proposed method encompasses and extends existing methods, such as kinky inference or NSM. It is rigorously proven that the proposed learning method ensures a bounded estimation of the prediction error and that the method is a learning algorithm. This is a key aspect, since it allows one to relate the accuracy of the predictions to the density of the available observations. Besides, it renders the online version of such algorithm applicable, that is, makes it prone to decrease the prediction error as more data points are taking into consideration by the predictor.

Based on the properties of the proposed estimation method, the predictive control law presented in Chapter 3 has been extended to prediction models based on CHoKI. All the results presented in this chapter encompass the standard KI approach, which could be considered as a sub-case of the more general CHoKI framework. Hence, the method proposed here is in general more precise and yields smaller prediction errors without loss of generality. The properties of the proposed estimation method could also be used to enhance existing controllers, such as [13] or [40].

The main advantage of using CHoKI as prediction method in the design of safe LBMPC, in contrast to standard KI, is two-fold. First, the prediction error bounds obtained are decreased. This is particularly important for the design of robust MPC controllers, since this reduces the size of the sets used to tighten the constraints to guarantee safety. Second, the accuracy in the estimation of the reachable sets is improved, by reducing the expansion rate of the reachable sets, leading also to a significant reduction of the conservativeness of the tightened constraints. A procedure is proposed to calculate the reachable sets and it is proven that the result yields the tightest sets that can be computed using the componentwise Hölder condition. Various case studies illustrate that CHoKI notably reduces the conservatism of the design when compared to standard KI methods, thus enhancing the performance and enlarging the domain of attraction of the proposed controller.

The results in this chapter were presented in [99, 103].

4.2 Extension of Hölder properties

4.2.1 Componentwise Hölder continuity

The standard Hölder continuity condition allows one to estimate the effect on the outputs of a variation on the inputs, bounding the worst case. In this section, a new approach on the Hölder property is considered. It exploits the

fact that a function may vary sharply along one dimension of the input, while varying smoothly with respect to other input dimensions. Besides, if the function maps the inputs to several outputs, it seems intuitive to distinguish different parameters of the Hölder continuity for each output, as if they were different functions.

Hence, we propose to use matrices as the parameters of the Hölder property: $\mathcal{L}, \mathcal{P} \in \mathbb{R}^{n_y \times n_w}$, yielding the following definition:

Definition 4.1 (Componentwise Hölder continuity)**.** *Given the matrices $\mathcal{L}$ and $\mathcal{P}$, a function $f : \mathcal{W} \to \mathcal{Y}$ is componentwise $\mathcal{L}$-$\mathcal{P}$-Hölder continuous if $\forall w_1, w_2 \in \mathcal{W}$ and $\forall i \in \mathbb{I}_1^{n_y}$*

$$|f_i(w_1) - f_i(w_2)| \leq \sum_{j=1}^{n_w} \mathcal{L}_{ij} |w_{1,j} - w_{2,j}|^{\mathcal{P}_{ij}}. \tag{4.1}$$

In Hölder continuity, L aggregates the effects of the inputs on the outputs into a single constant. On the contrary, the proposed componentwise approach uses each $\mathcal{L}_{i,j}$ to take into account separately the effect of each input on each output.

This componentwise Hölder continuity condition may be rewritten in a more compact form. To this end, new notation is introduced. Given a vector $w \in \mathbb{R}^{n_w}$ and two matrices $\mathcal{L}, \mathcal{P} \in \mathbb{R}^{n_y \times n_w}$, we define

$$\mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(w) := \left(a : a_i = \sum_{j=1}^{n_w} \mathcal{L}_{i,j} w_j^{\mathcal{P}_{i,j}}, \forall i \in \mathbb{I}_1^{n_y} \right). \tag{4.2}$$

Then, componentwise Hölder continuity (cf. (4.1)) can be written as

$$|f(w_1) - f(w_2)| \leq \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|w_1 - w_2|), \ \forall w_1, w_2. \tag{4.3}$$

The following theorem states under which conditions Hölder continuity and componentwise Hölder continuity are equivalent.

Theorem 4.1. *Let $f : \mathcal{W} \subseteq \mathbb{R}^{n_w} \to \mathcal{Y} \subseteq \mathbb{R}^{n_y}$.*

1. *If f is Hölder continuous in $\mathcal{W}$, then f is componentwise Hölder continuous in $\mathcal{W}$.*
2. *If $\mathcal{W}$ is compact and f is componentwise Hölder continuous in $\mathcal{W}$, then f is Hölder continuous in $\mathcal{W}$.*

Proof (point 1).

Consider two parameters L and p such that $f : \mathcal{W} \to \mathcal{Y}$ is Hölder continuous. Provided that for a given $w \in \mathbb{R}^{n_w}$, $\|w\| \leq \sqrt{n_w}\|w\|_\infty$, define $L_\infty = L n_w^{(p/2)}$. Then, we have that

$$\begin{aligned} \|y_1 - y_2\| &\leq L\|w_1 - w_2\|^p && (4.4a)\\ &\leq L_\infty \|w_1 - w_2\|_\infty^p && (4.4b)\\ &= L_\infty \max_j (|w_{1,j} - w_{2,j}|)^p && (4.4c)\\ &\leq L_\infty \sum_{j=1}^{n_w} |w_{1,j} - w_{2,j}|^p, && (4.4d) \end{aligned}$$

where inequality (4.4d) holds provided that $|w_{1,j} - w_{2,j}|^p \geq 0$, $\forall j$.

Then, defining $\mathcal{L} = L_\infty \mathbb{1}_{n_y \times n_w}$ and $\mathcal{P} = p\mathbb{1}_{n_y \times n_w}$, the function f is componentwise Hölder continuous. □

Proof (point 2).

Given that for any $w_1, w_2, j \in \mathbb{I}_1^{n_w}$, $|w_{1,j} - w_{2,j}| \leq \|w_1 - w_2\|_\infty$,

$$\begin{aligned} |y_{i,1} - y_{i,2}| &\leq \sum_{j=1}^{n_w} \mathcal{L}_{i,j} |w_{1,j} - w_{2,j}|^{\mathcal{P}_{i,j}} \\ &\leq \sum_{j=1}^{n_w} \mathcal{L}_{i,j} \|w_1 - w_2\|_\infty^{\mathcal{P}_{i,j}}. \end{aligned}$$

If $\mathcal{W}$ is a compact space, there exists a c such that $\|w\|_\infty \leq c$, $\forall w \in \mathcal{W}$, so $\|w_1 - w_2\| \leq 2c$, $\forall w_1, w_2 \in \mathcal{W}$. Besides, if $x \in [0,\, 1]$ and $p_1 \geq p_2$, with $p_1, p_2 \in (0, 1]$, then $x^{p_1} \leq x^{p_2}$.

Let $p = \min_{i,j} \mathcal{P}_{i,j}$ and let $L = \max_i \sum_{j=1}^{n_y} \mathcal{L}_{i,j} (2c)^{\mathcal{P}_{i,j} - p}$. Then

$$\begin{aligned} \mathcal{L}_{i,j} \|w_1 - w_2\|_\infty^{\mathcal{P}_{i,j}} &\leq \mathcal{L}_{i,j} (2c)^{\mathcal{P}_{i,j}} \left(\frac{\|w_1 - w_2\|}{2c} \right)^{\mathcal{P}_{i,j}} \\ &\leq \mathcal{L}_{i,j} (2c)^{\mathcal{P}_{i,j}} \left(\frac{\|w_1 - w_2\|_\infty}{2c} \right)^{p} \\ &= \mathcal{L}_{i,j} (2c)^{\mathcal{P}_{i,j} - p} \|w_1 - w_2\|_\infty^p. \end{aligned}$$

Hence,

$$\begin{aligned} |f_i(w_1) - f_i(w_2)| &\leq \sum_{j=1}^{n_y} \mathcal{L}_{i,j} \|w_1 - w_2\|_\infty^{\mathcal{P}_{i,j}} \\ &\leq \sum_{j=1}^{n_y} \mathcal{L}_{i,j} (2c)^{\mathcal{P}_{i,j}-p} \|w_1 - w_2\|_\infty^p \\ &\leq L\|w_1 - w_2\|_\infty^p. \end{aligned}$$

And since

$$\exists i^* : |f_{i^*}(w_1) - f_{i^*}(w_2)| = \|f(w_1) - f(w_2)\|_\infty,$$

we have that

$$\|f(w_1) - f(w_2)\|_\infty \leq L\|w_1 - w_2\|_\infty^p.$$

□

Corollary 4.1. *If $\mathcal{P} = p\mathbb{1}_{n_y \times n_w}$, the equivalence in Theorem 4.1 holds for any space, even if it is not compact.*

With a slight abuse of notation, in what is to follow the mapping in (4.2) may also be used for sets, such that for a given set $A \subseteq \mathbb{R}^{n_w}$,

$$\mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(A) := \{\mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(x) | x \in A\}. \tag{4.5}$$

Corollary 4.2. *In the Lipschitz case and for every output component $i \in \mathbb{I}_1^{n_y}$, if the Lipschitz constant is such that $L = \| \mathcal{L}_i\|_\infty$, then*

$$L\|w_i\|_\infty \geq \mathfrak{d}_{\mathcal{L}_i}(|w_i|).$$

Proof.

Using the Cauchy-Schwarz inequality,

$$\mathfrak{d}_{\mathcal{L}}(|w|) = \langle \mathcal{L}, |w| \rangle \leq \| \mathcal{L}\|_\infty \|w\|_\infty = L\|w\|_\infty,$$

where $\langle \cdot, \cdot \rangle$ denotes the canonical inner product. □

4.2.2 Componentwise Hölder kinky inference

Proceeding as in the standard KI approach, assuming that f is Hölder continuous, and given a data set of input-output observations, this section presents the *componentwise Hölder kinky inference* (CHoKI) predictor.

Provided that the componentwise Hölder condition (4.1) holds in virtue of Theorem 4.1, the proposed estimation method is:

$$\hat{\mathfrak{f}}_{\mathrm{C}}(q;\Theta,\mathcal{D}) = \frac{1}{2}\left(\min_{i=1,\ldots,N_{\mathcal{D}}}(\tilde{y}_i + \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|q-w_i|)) + \max_{i=1,\ldots,N_{\mathcal{D}}}(\tilde{y}_i - \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|q-w_i|))\right), \tag{4.6}$$

where $\Theta = (\mathcal{L}, \mathcal{P})$.

The resulting prediction function is componentwise Hölder continuous, with the same parameters, as stated in the following lemma.

Lemma 4.1. *For a given $\Theta = (\mathcal{L}, \mathcal{P})$, the CHoKI predictor $\hat{\mathfrak{f}}_{\mathrm{C}}$ given by (4.6) is componentwise $\mathcal{L}$-$\mathcal{P}$-Hölder continuous.*

Proof.

Given two scalars $x \geq 0$ and $0 < p \leq 1$, x^p is concave, so

$$\|x_1\|^p - \|x_2\|^p \leq \|x_1 - x_2\|^p.$$

Then, given $\mathcal{L}$, $\mathcal{P}$, $\mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|w|)$ is Hölder continuous, since for each i-th component

$$\sum_{j=1}^{n_w} \mathcal{L}_{i,j}|w_{1,j}|^{\mathcal{P}_{i,j}} - \sum_{j=1}^{n_w} \mathcal{L}_{i,j}|w_{2,j}|^{\mathcal{P}_{i,j}} \leq \sum_{j=1}^{n_w} \mathcal{L}_{i,j}(|w_{1,j} - w_{2,j}|)^{\mathcal{P}_{i,j}}.$$

The sum of two componentwise Hölder continuous functions f, g with Hölder parameters $\mathcal{L}$ and $\mathcal{P}$ is also componentwise Hölder, since

$$\begin{aligned} |f(w_1) + g(w_1) - f(w_2) - g(w_2)| &\leq |f(w_1) - f(w_2)| + |g(w_1) - g(w_2)| \\ &\leq \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|w_1 - w_2|) + \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|w_1 - w_2|). \end{aligned}$$

Finally, the minimum (or, equivalently the maximum) of componentwise Hölder functions is also componentwise Hölder: Let denote $h(w) = \min(f, g)$ and assume, w.o.l.g., that $f(w_1) > g(w_2)$. Then, if $f(w_1) \leq g(w_1)$,

$$\begin{aligned} |h(w_1) - h(w_2)| &= |f(w_1) - g(w_2)| = f(w_1) - g(w_2) \\ &\leq g(w_1) - g(w_2) \leq \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|w_1 - w_2|). \end{aligned}$$

Hence, $\hat{\mathfrak{f}}_{\mathrm{C}}$ (cf. eq. (4.6)) is componentwise Hölder continuous, with Hölder parameters $\mathcal{L}$ and $\mathcal{P}$. □

In case that the parameters $\mathcal{L}$ and $\mathcal{P}$ are unknown a priori, they must be estimated from the available input-output data. In the LACKI method, the

Hölder constant can be derived from $\mathcal{D}$ by a procedure based on sample consistency (see Appendix B). However, in the componentwise case, the LACKI method cannot be used, since $\|w_1 - w_2\|$ provides an aggregated measurement of the effect of the inputs on the outputs, and there is no direct information of the contribution of each input on a particular output. In order to infer this contribution, an optimization method is adopted, extending the results of POKI to obtain the matrices $\mathcal{L}$ and $\mathcal{P}$.

The estimation of the parameters $\Theta = (\mathcal{L}, \mathcal{P})$ is based on solving an optimization problem offline, which depends on a regularization parameter $\eta \in \mathbb{R}^{n_y}$ and two data sets: the data set $\mathcal{D}$, used for estimation, and a data set $\mathcal{D}_{\text{test}}$ used for validation.

Thus, and hybrid method that merges LACKI and POKI is proposed, to compute the parameters $\mathcal{L}$, $\mathcal{P}$ as solutions of the following optimization problem, where a measure of the performance of the prediction over the data set $\mathcal{D}_{\text{test}}$, $g(\Theta, \mathcal{D}, \mathcal{D}_{\text{test}})$, and a regularization term are minimized, subject to a constraint that ensures their consistency with the samples of the data set:

$$\Theta = \arg\min_{\Theta} g(\Theta, \mathcal{D}, \mathcal{D}_{\text{test}}) + \tau_{\mathcal{L}} \| \mathcal{L} - \mathcal{L}_0 \|_1, \tag{4.7a}$$

$$\text{s.t.} \quad |\tilde{y}_i - \tilde{y}_j| - \eta \leq \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|w_i - w_j|) \quad \forall w_i, w_j \in \mathcal{W}_{\mathcal{D}}, w_i \neq w_j, \tag{4.7b}$$

$$0 < \mathcal{P}_{ij} \leq 1, i \in \mathbb{I}_1^{n_y}, j \in \mathbb{I}_1^{n_w}, \tag{4.7c}$$

where $\tau_{\mathcal{L}}$ is a design regularization hyper-parameter used to ensure boundedness of $\mathcal{L}$, and $\mathcal{L}_0$ stands for any possible prior guess of $\mathcal{L}$. Besides, the cost function g must be positive and bounded, for any size of $\mathcal{D}$. A possible choice of the performance measure is the mean squared prediction error:

$$g(\Theta, \mathcal{D}, \mathcal{D}_{\text{test}}) = \frac{1}{N_{\mathcal{D}}} \sum_{w_i \in \mathcal{W}_{\mathcal{D}_{\text{test}}}} \| \hat{\mathfrak{f}}_{\text{C}}(w_i; \Theta, \mathcal{D}) - \tilde{y}_i \|^2. \tag{4.8}$$

Remark 4.1. *For an analysis on the effect of the regularization hyper-parameter η, the reader is referred to [36], where $\eta \geq 2\bar{\mathfrak{e}}$ is taken for each component to effectively smooth out the effect of the noise in the prediction.*

Remark 4.2. *The regularization term of the cost function, $\tau_{\mathcal{L}} \| \mathcal{L} - \mathcal{L}_0 \|_1$, prevents the problem from overfitting the noise, while ensuring boundedness of $\mathcal{L}$. If $\mathcal{D}_{\text{test}}$ is separate from $\mathcal{D}$ and such that g is bounded for all $\mathcal{L}$, then it can be removed, setting $\tau_{\mathcal{L}} = 0$.*

Remark 4.3. *In practice, it may be easier to fix $\mathcal{P}$ a priori, and to optimize over $\mathcal{L}$, provided that the assumptions hold for the chosen $\mathcal{P}$.*

Next, based on T. Mitchell's definition of a learning algorithm [113], it will be proven that CHoKI is a learning method. For this, the following lemma, which proves sample consistency, is needed.

Lemma 4.2 (Sample consistency of CHoKI)**.** *If $\mathcal{L}$ and $\mathcal{P}$ are obtained as in (4.7), the CHoKI predictor (4.6) is sample-consistent (up to $\eta/2+\bar{\mathfrak{e}}$), that is,*

$$|\hat{\mathfrak{f}}_{\mathrm{C}}(w_k) - f(w_k)| \leq \frac{\eta}{2} + \bar{\mathfrak{e}},\ \forall w_k \in \mathcal{W}_{\mathcal{D}}. \tag{4.9}$$

Proof. Let us denote

$$i = \arg\min_n \left(\tilde{y}_n + \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|w_k - w_n|)\right),$$

$$j = \arg\max_n \left(\tilde{y}_n - \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|w_k - w_n|)\right).$$

Then,

$$\hat{\mathfrak{f}}_{\mathrm{C}}(w_k) \;=\; \frac{1}{2}\underbrace{\left(\tilde{y}_i + \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|w_k - w_i|)\right)}_{B} + \frac{1}{2}\underbrace{\left(\tilde{y}_j - \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|w_k - w_j|)\right)}_{A} = \frac{1}{2}B + \frac{1}{2}A.$$

It is first proven that $A \geq \tilde{y}_k$. If $j = k$ this is immediate. Otherwise, note that

$$\begin{aligned} A &= \tilde{y}_j - \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|w_k - w_j|) \\ &\geq \tilde{y}_k - \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|w_k - w_k|) = \tilde{y}_k. \end{aligned}$$

Then, it is proven that $A \leq \tilde{y}_k + \eta$. From (4.7b) we have that

$$|\tilde{y}_k - \tilde{y}_j| \leq \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|w_k - w_j|) + \eta.$$

Provided that $\tilde{y}_j \geq A \geq \tilde{y}_k$,

$$\begin{aligned} A &= \tilde{y}_j - \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|w_k - w_j|) \\ &\leq \tilde{y}_k + |\tilde{y}_j - \tilde{y}_k| - \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|w_k - w_j|) \leq \tilde{y}_k + \eta. \end{aligned}$$

The same procedure is applied to prove that $\tilde{y}_k - \eta \leq B \leq \tilde{y}_k$. Then, $\hat{\mathfrak{f}}_{\mathrm{C}}(w_k)$ is such that

$$\frac{1}{2}(\tilde{y}_k - \eta) + \frac{1}{2}\tilde{y}_k \;\leq \hat{\mathfrak{f}}_{\mathrm{C}}(w_k) \leq\; \frac{1}{2}(\tilde{y}_k + \eta) + \frac{1}{2}\tilde{y}_k,$$

or equivalently,

$$\tilde{y}_k - \frac{\eta}{2} \leq \hat{\mathfrak{f}}_{\mathrm{C}}(w_k) \leq \tilde{y}_k + \frac{\eta}{2},\ |\hat{\mathfrak{f}}_{\mathrm{C}}(w_k) - \tilde{y}_k| \leq \frac{\eta}{2}.$$

Finally, since $\tilde{y}_k \leq f(w_k) + \bar{\mathfrak{e}}$,

$$|\hat{\mathfrak{f}}_{\mathrm{C}}(w_k) - f(w_k)| \leq \frac{\eta}{2} + \bar{\mathfrak{e}}.$$

□

Theorem 4.2. *Let $\Theta = (\mathcal{L}, \mathcal{P})$ be obtained as the solution of (4.7) for a data set $\mathcal{D}$ with $\eta \geq 2\bar{\mathfrak{e}}$, and assume that the function f satisfies the componentwise Hölder condition (4.3) for the pair $(\mathcal{L}_f, \mathcal{P})$ in $\mathcal{W}$. Then, $\mathcal{L}$ is bounded and*

$$|f(w) - \hat{\mathfrak{f}}_{\mathrm{C}}(w; \Theta, \mathcal{D})| \leq \mathfrak{d}^{\mathcal{P}}_{\mathcal{L}_f + \mathcal{L}}(\mathcal{R}_{\mathcal{D}}) + \frac{\eta}{2} + \bar{\mathfrak{e}}, \tag{4.10}$$

where $\mathcal{R}_{\mathcal{D}} = \max_{w \in \mathcal{W}} \min_{w_j \in \mathcal{W}_{\mathcal{D}}} (|w_j - w|)$ measures the maximum radius between a possible query and the data set.

Proof.

First, it is proven that $\mathcal{L}$ as a solution of (4.7) is bounded (i.e. not infinity). To this end, it is proven that $\mathcal{L}_f$ is a possible bounded solution that satisfies the constraint (4.7b). Indeed, note that since f is componentwise Hölder continuous for $(\mathcal{L}_f, \mathcal{P})$, $\mathcal{L}_f$ is bounded and for all $w_i \neq w_j \in \mathcal{W}$

$$|\tilde{y}_i - \tilde{y}_j| - \eta \leq |y_i - y_j| \leq \mathfrak{d}^{\mathcal{P}}_{\mathcal{L}_f}(|w_i - w_j|),$$

which satisfies the condition.

Next, it is proven that the solution of (4.7), i.e.

$$\Theta^* = \arg\min_{\Theta} g(\Theta, \mathcal{D}, \mathcal{D}_{\mathrm{test}}) + \tau_{\mathcal{L}} \| \mathcal{L} - \mathcal{L}_0 \|_1,$$

is bounded even for infinitely dense data sets, as $N_{\mathcal{D}} \to \infty$.

To this end, $g(\Theta, \mathcal{D}, \mathcal{D}_{\mathrm{test}})$ must be bounded for all $w \in \mathcal{W}$. For example consider

$$g(\Theta, \mathcal{D}, \mathcal{D}_{\mathrm{test}}) = \frac{1}{N_{\mathcal{D}}} \sum_{w_i \in \mathcal{W}_{\mathcal{D}_{\mathrm{test}}}} \|\hat{\mathfrak{f}}_{\mathrm{C}}(w_i; \Theta, \mathcal{D}) - \tilde{y}_i\|^2.$$

Since $\mathcal{W}$ is compact, the noise is bounded and f is Hölder, then $\tilde{y}_i$ is bounded. Besides, Hölder continuity of $\hat{\mathfrak{f}}_{\mathrm{C}}$ ensures that $\hat{\mathfrak{f}}_{\mathrm{C}}(w, \Theta, \mathcal{D})$ is bounded for any Θ. Then, $g(\cdot)$ is upper-bounded by some k_1, irrespective of the number of data points.

Hence, assuming that $g(\cdot)$ is bounded by k_1, and taking into account that $\Theta_f = (L_f, P)$ is a feasible solution of the optimization problem, we have that

$$\begin{aligned} g(\Theta^*, \mathcal{D}, \mathcal{D}_{\text{test}}) + \tau_{\mathcal{L}} \| \mathcal{L}^* - \mathcal{L}_0 \|_1 &\leq g(\Theta_f, \mathcal{D}, \mathcal{D}_{\text{test}}) + \tau_{\mathcal{L}} \| \mathcal{L}_f - \mathcal{L}_0 \|_1 \\ &\leq k_1 + \tau_{\mathcal{L}} \| \mathcal{L}_f - \mathcal{L}_0 \|_1 = k_2, \end{aligned}$$

for certain constant k_2. Given that $g(\cdot)$ is positive, we have that

$$\tau_{\mathcal{L}} \| \mathcal{L}^* - \mathcal{L}_0 \|_1 \leq g(\Theta^*, \mathcal{D}, \mathcal{D}_{\text{test}}) + \tau_{\mathcal{L}} \| \mathcal{L}^* - \mathcal{L}_0 \|_1 \leq k_2.$$

Therefore, $\mathcal{L}$ is bounded for any size of the data set.

Finally, the bound given in the theorem is proven. Denote

$$w_n = \arg \min_{w_j \in \mathcal{W}_{\mathcal{D}}} (|w_j - w|)$$

the closest data point to the query w. The following operation is decomposed into three addends:

$$\begin{aligned} |f(w) - \hat{\mathfrak{f}}_{\text{C}}(w)| &= |f(w) - f(w_n)| \\ &\quad + |\hat{\mathfrak{f}}_{\text{C}}(w_n) - \hat{\mathfrak{f}}_{\text{C}}(w)| + |f(w_n) - \hat{\mathfrak{f}}_{\text{C}}(w_n)|. \end{aligned}$$

The first term is less than or equal to $\mathfrak{d}^{\mathcal{P}}_{\mathcal{L}_f}(|w - w_n|)$, which is bounded by $\mathfrak{d}^{\mathcal{P}}_{\mathcal{L}_f}(\mathcal{R}_{\mathcal{D}})$.

The second term is less than or equal to $\mathfrak{d}^{\mathcal{P}}_{\mathcal{L}}(|w - w_n|) + \eta$, which is bounded by $\mathfrak{d}^{\mathcal{P}}_{\mathcal{L}}(\mathcal{R}_{\mathcal{D}}) + \eta$.

The third term is less than or equal to $\eta/2 + \bar{\mathfrak{e}}$, as proven in Lemma 4.2.

Then, the three bounds add together, proving (4.10). □

Based on this theorem, it can be derived that the prediction error is bounded for all query q in a compact space $\mathcal{W}$. Besides, it proves that as more observations are added to the data set, the prediction error decreases, vanishing up to $\eta/2 + \bar{\mathfrak{e}}$ for infinitely dense data sets, when $\mathcal{R}_{\mathcal{D}} \to 0$.

Finally, it is proven that CHoKI enhances the existing methods based on Hölder continuity. This is achieved by proving that KI is a particular case of CHoKI for a certain parameter setting, as it is stated in the following lemma.

Lemma 4.3. *If $\mathcal{L} = L\mathbb{1}^T_{n_w}$ and the Lipschitz case (i.e. $\mathcal{P} = \mathbb{1}_{n_y \times n_w}$) is applied, then the proposed prediction method is equal to the standard KI when the one norm is used, i.e.,*

$$\hat{\mathfrak{f}}(q; L, \mathcal{D}) = \hat{\mathfrak{f}}_{\text{C}}(q;\, \mathcal{L}, \mathcal{D}). \tag{4.11}$$

Proof.

The proof follows developing $\mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|q-w_i|)$ in eq. (4.6) using the definition in eq. (4.2) and bearing in mind that $\mathcal{L}_j = L$, for all column $j \in \mathbb{I}_1^{n_w}$. Hence, using the one norm for the standard KI,

$$L\|q-w_i\| = L\sum_{j=1}^{n_w}|q-w_i| = \sum_{j=1}^{n_w}\mathcal{L}_j|q-w_i| = \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|q-w_i|).$$

□

Note that the method proposed to obtain $\mathcal{L}$, $\mathcal{P}$ adds degrees of freedom with respect to the scalar KI case. So, in general, CHoKI will perform equal or better than standard KI over $\mathcal{D}_{\text{test}}$, since the latter is a particular case of the former.

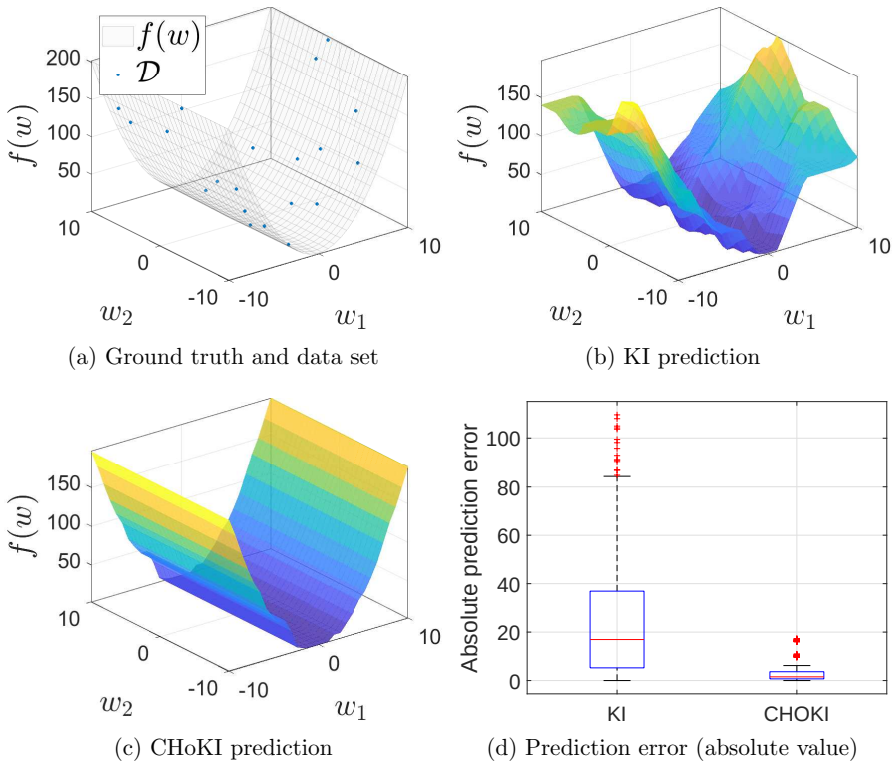
Remark 4.4. *In the case that prior knowledge of the true Lipschitz parameter is available (or assumed to be known), as in nonlinear set membership approaches [40], the CHoKI method proposed here can be applied, improving the performance of the derived controller, provided that the uncertainty bounds will be decreased.*

The overall performance of the proposed method is illustrated in the following example:

Example 4.1. *Consider the function* $f : \mathcal{W} \subset \mathbb{R}^2 \to \mathbb{R}$*:*

$$f(w) = 2w_1^2 + \frac{\sqrt{|w_2|}}{2},$$

within the input space $\mathcal{W} = \{w : -10I_2 \leq w \leq 10I_2\}$*. Note that* f *is not Lipschitz continuous in the origin. Figure 4.1 depicts the prediction errors generated by CHoKI and KI, trained on a set of* $N_{\mathcal{D}} = 20$ *random sampled data, over a grid of 900 query points. The Hölder parameters were obtained as per (4.7) with* $\eta = 0$*, yielding* $L = 34.6$*,* $p = 1$ *and* $\mathcal{L} = [37.5\ 0.2]$*,* $\mathcal{P} = [1\ 0.75]$*. Note that with CHoKI the point-wise prediction error decreases up to* 84%*. Besides, the learning property of CHoKI is illustrated in Figure 4.2, using different grids to obtain the data set and fixing the Hölder parameters such that (4.7b) is satisfied. The prediction error decreases as the density of the data set increases.*



(a) Ground truth and data set (b) KI prediction

(c) CHoKI prediction (d) Prediction error (absolute value)

Figure 4.1: Prediction of $f(w) = 2w_1^2 + 0.5\sqrt{|w_2|}$ using both the standard KI and the CHoKI method, given $N_{\mathcal{D}} = 20$ random data points.

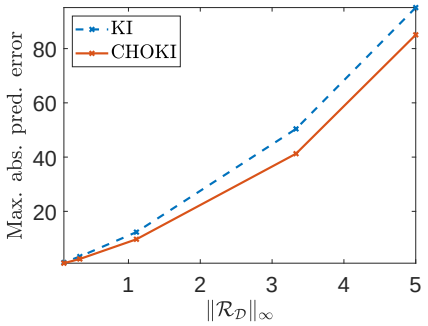


Figure 4.2: Maximum absolute prediction error w.r.t. the data set's density.

4.3 Reachability sets

In order to implement the control law presented in Chapter 3, it is necessary to calculate the reachability sets for the CHoKI predictor. Although the method presented in Section 3.3 could be used to obtain conservative Hölder bounds, in this section we present a method that explicitly takes into account the properties of CHoKI, yielding in general tighter sets. In particular, the sets of constraints is now obtained as

$$\mathcal{Y}_j = \mathcal{Y}_{j-1} \ominus \mathcal{R}_j, \tag{4.12}$$

with $\mathcal{Y}_0 = \mathcal{Y}$. $\mathcal{R}_j$ accounts for the possible deviation of the nominal predictions $\hat{y}(j|k)$ from the real system, j steps ahead.

Next, we present a procedure based on the CHoKI predictor to calculate these sets, which in general provides less conservative results than the ones given by standard procedures or the one presented in Chapter 3. For its calculation, we will make use of the following lemma.

Lemma 4.4. *Consider a sequence of future inputs $u(k+i)$ for $i \in \mathbb{I}_0^{N-1}$. Let $c_1 \in \mathbb{R}^{n_y}$ be a vector such that*

$$|y(k+1) - \hat{y}(1|k)| \leq c_1. \tag{4.13}$$

The mismatch between a prediction at time step $k+j$ given the measurement at time step k and the prediction at that time step given the measurement at time $k+1$ for the same sequence of control inputs is bounded by the sets

$$|\hat{y}(j|k) - \hat{y}(j-1|k+1)| \in \mathcal{M}_j \subseteq \mathbb{R}^{n_y} \tag{4.14a}$$
$$|\hat{w}(j|k) - \hat{w}(j-1|k+1)| \in \mathcal{G}_j \subseteq \mathbb{R}^{n_w}. \tag{4.14b}$$

The sets $\mathcal{M}$ and $\mathcal{G}$ can be obtained from the recursion

$$\mathcal{M}_j = \mathbb{B}(\mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(\mathcal{G}_{j-1})), \tag{4.15a}$$
$$\mathcal{G}_j = \mathcal{M}_j \times \ldots \times \mathcal{M}_{\sigma(j)} \times \{0\} \times \ldots \times \{0\}, \tag{4.15b}$$

with $\sigma(j) = \max(1, j - n_a)$, and $\mathcal{M}_1 = \mathbb{B}(c_1)$.

Proof.

It is first proven that the Cartesian hull of the map $\mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(\mathcal{B}(v))$, that is the tightest ball containing it, is a ball given by $\mathcal{B}(\mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(v))$, i.e.

$$\mathcal{B}(\mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(\mathcal{B}(v))) = \mathcal{B}(\mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(v)). \tag{4.16}$$

Note that $\mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(\mathcal{B}(v)) = \{\mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(x) | x \in \mathcal{B}(v)\}$. For every component $i \in \mathbb{I}_1^{n_y}$ it can be bounded that for all $j \in \mathbb{I}_1^{n_w}$, noticing that $x \in \mathcal{B}(v)$,

$$0 \leq \mathcal{L}_{ij} x_j^{\mathcal{P}_{ij}} \leq \mathcal{L}_{ij} v_j^{\mathcal{P}_{ij}},$$

since $|x_j| \leq v_j$ and $v \in \mathcal{B}(v)$. Equivalently, $\mathcal{B}(\mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(\mathbb{B}(v))) = \mathbb{B}(\mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(v))$, since $\mathbb{B}(v)$ is just the intersection between $\mathcal{B}(v)$ and the first orthant.

Then, (4.18) is proven, i.e., that both $\mathcal{M}_j$ and $\mathcal{G}_j$ are hypercubes and can therefore be defined by the box over a vector, $\mathbb{B}(c_j), \mathbb{B}(d_j)$, respectively. For $j = 1$, $\mathcal{M}_1 = \mathbb{B}(c_1)$ and $\mathcal{G}_1 = \mathcal{M}_1 \times \{0\} \times \ldots \times \{0\} = \mathbb{B}(d_1)$.

For $j > 1$, given that $\mathcal{G}_{j-1} = \mathbb{B}(d_{j-1})$ and (4.16), we have that

$$\mathcal{M}_j = \mathcal{B}(\mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(\mathcal{G}_{j-1})) = \mathbb{B}(\mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(d_{j-1})) = \mathbb{B}(c_j),$$

$$\mathcal{G}_j = \mathcal{M}_j \times \ldots \times \mathcal{M}_{\sigma(j)} \times \{0\} \times \ldots \times \{0\} = \mathbb{B}(d_j).$$

□

This Lemma proves that the smallest sets that contain all possible values of $|\hat{y}(j|k) - \hat{y}(j-1|k+1)|$ and $|\hat{w}(j|k) - \hat{w}(j-1|k+1)|$ are $\mathcal{M}_j$ and $\mathcal{G}_j$, respectively.

The set $\mathcal{R}_j$ is defined as

$$\mathcal{R}_j = \{y : |y| \in \mathcal{M}_j\}, \tag{4.17}$$

for all $j \in \mathbb{I}_1^N$, with $\mathcal{M}_j$ calculated from (4.15) with $\mathcal{M}_1 = \mathbb{B}(\mu)$. The following lemma proves that the sets $\mathcal{M}_j$ and $\mathcal{G}_j$ are balls that can be calculated by a simple recursion.

Lemma 4.5. *Given Lemma 4.4, let $c_j \in \mathbb{R}^{n_y}$ and $d_j \in \mathbb{R}^{n_w}$ be such that*

$$\mathcal{M}_j = \mathbb{B}(c_j), \tag{4.18a}$$
$$\mathcal{G}_j = \mathbb{B}(d_j). \tag{4.18b}$$

The sets $\mathcal{M}_j$ and $\mathcal{G}_j$ can be calculated using the recursion

$$c_j = \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(d_{j-1}), \tag{4.19a}$$
$$d_j = (c_j, \ldots, c_{\sigma(j)}, 0, \ldots, 0). \tag{4.19b}$$

Besides, $\mathcal{R}_j = \mathcal{B}(c_j)$.

Proof.

Provided that $w(k) = (x(k), u(k))$, and that the sequence of future inputs $u(k+i)$ for $i \in \mathbb{I}_0^{N-1}$ is given,

$$|\hat{w}(j|k) - \hat{w}(j-1|k+1)| = \begin{bmatrix} |\hat{y}(j|k) - \hat{y}(j-1|k+1)| \\ |\hat{y}(j-1|k) - \hat{y}(j-2|k+1)| \\ \vdots \\ |\hat{y}(j-n_a|k) - \hat{y}(j-n_a-1|k+1)| \\ |u(k+j-1) - u(k+j-1)| \\ \vdots \\ |u(k+j-n_b) - u(k+j-n_b)| \\ |u(k+j) - u(k+j)| \end{bmatrix}. \quad (4.20)$$

These $\hat{y}$ are predicted values if $j > n_a$, and real measurements if not. Equation (4.20) directly translates into (4.15b), provided that if $\hat{y}(j|k)$ is a real measurement, $|y(j|k) - y(j-1|k+1)|$ equals $\mathbb{0}_{n_y}$, and belongs to $\mathcal{M}_j$ otherwise.

To obtain $\mathcal{M}_j$ we make use of the componentwise Hölder continuity of the predictor $\hat{\mathfrak{f}}_{\mathrm{C}}$. Given equation (4.13), we have that $\mathcal{M}_1 = \mathbb{B}(c_1)$. Note that

$$\hat{y}(j+1|k) = \hat{\mathfrak{f}}_{\mathrm{C}}(\hat{x}(j|k), u(k+j); \Theta, \mathcal{D}),$$

and that

$$|\hat{\mathfrak{f}}_{\mathrm{C},i} - \hat{\mathfrak{f}}_{\mathrm{C},j}| \leq \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(|w_i - w_j|).$$

Hence,

$$\begin{aligned} |\Delta y| &= |\hat{y}(j+1|k) - \hat{y}(j|k+1)| \\ &= |\hat{\mathfrak{f}}_{\mathrm{C}}(\hat{x}(j|k), u(k+j)) - \hat{\mathfrak{f}}_{\mathrm{C}}(\hat{x}(j-1|k), u(k+j))| \\ &\leq \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}} \left(\begin{bmatrix} |\hat{y}(j|k) - \hat{y}(j-1|k+1)| \\ |\hat{y}(j-1|k) - \hat{y}(j-2|k+1)| \\ \vdots \\ |\hat{y}(j-n_a|k) - \hat{y}(j-n_a-1|k+1)| \\ |u(k+j-1) - u(k+j-1)| \\ \vdots \\ |u(k+j-n_b) - u(k+j-n_b)| \\ |u(k+j) - u(k+j)| \end{bmatrix} \right). \end{aligned}$$

From this inequality follows that $|\Delta y| \leq \mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(\mathcal{G}_j)$, so the minimum set that contains all possible values of $|\Delta y|$ is the Cartesian closed topological hull, that is, $\mathcal{M}_{j+1} = \mathcal{B}(\mathfrak{d}_{\mathcal{L}}^{\mathcal{P}}(\mathcal{G}_j))$. □

If the set $\mathcal{Y}$ is a polytope, as it is customary, then the resulting tightened constraints are also polytopes. Notice that these calculations are done only once, offline.

4.4 Stabilizing conditions for CHoKI-MPC

In order to recover the safe-by-design properties of the controller (3.8) based on KI, some of the assumptions are extended to the componentwise framework as follows:

Assumption 4.1. *The stage cost function $\ell(x,u)$ is a componentwise Hölder continuous, positive definite function such that $\ell(x,u) \geq \alpha(\|x\|)$, where α is a $\mathcal{K}$-function. Given $\mathcal{P}$, its Hölder constant is $\mathcal{L}_\ell$.*

Assumption 4.2. *V_f is a componentwise Hölder continuous, positive definite function, with Hölder parameters $\mathcal{L}_f, \mathcal{P}$, such that Assumption 3.3 holds.*

Assumption 4.3. *Assumption 3.4 holds redefining ν as*

$$\nu(\mu) = \sum_{j=1}^{N} \|\mathfrak{d}^{\mathcal{P}}_{\mathcal{L}_\ell}(\mathcal{G}_j)\|_\infty + \lambda \|\mathfrak{d}^{\mathcal{P}}_{\mathcal{L}_f}(\mathcal{G}_{N+1})\|_\infty. \tag{4.21}$$

Next, we present the stability result for the CHoKI-MPC, which follows the same line of reasoning as the stability proof presented in Chapter 3 for KI predictors.

Theorem 4.3 (ISS stability)**.** *Suppose that assumptions 3.1, 4.1-4.3 hold for the optimization problem $P_N(\cdot)$ obtained with the CHoKI predictor. Let $\kappa_N(x)$ be the control law derived from the solution of $P_N(x)$ applied using a receding horizon policy. Then, for any $x(0) \in \Gamma$, the system to be controlled by the control law $u(k) = \kappa_N(x(k))$ is input-to-state stable with respect to the estimation error, $x(k) \in \Gamma$, and the constraints are always satisfied, i.e. $y(k) \in \mathcal{Y}, \forall k$.*

Proof.

The proof of this theorem follows the same steps as the proof of Theorem 3.1 (see page 61). Lemma 3.2 is modified as follows:

Lemma 4.6. *For all $y \in \mathcal{Y}_j$ and for all Δy such that $|\Delta y| \in \mathcal{M}_j$, the sets $\mathcal{Y}_j$ are such that $y + \Delta y \in \mathcal{Y}_{j-1}$.*

Proof By definition, we have that if $|\Delta y| \in \mathcal{M}_j$ then $\Delta y \in \mathcal{R}_j$. Thus, since the origin is contained in $\mathcal{R}_j$,

$$y + \Delta y \in \mathcal{Y}_j \oplus \mathcal{R}_j = \mathcal{Y}_{j-1} \ominus \mathcal{R}_j \oplus \mathcal{R}_j \subseteq \mathcal{Y}_{j-1}.$$

□

Corollary 3.1 can be easily extended to analyse this theorem with respect to the validity of Assumption 3.1. As stated in Remark 3.6, if the prediction error is larger than the estimated μ for a period of time, Theorem 4.3 still holds if $x(k) \in \Gamma$ for that period of time. Furthermore, if Assumption 3.1 does not hold in general, we propose to have a backup controller, similar in spirit to the approach proposed by L. Hewing et al. [61]. Specifically, the controller proposed in Chapter 2 admits the extension to the CHoKI-based model, and it guarantees stability relaxing the hard constraints in the outputs to soft constraints included in the cost of the MPC.

It is important to remark that the benefits of the robust predictive controller based on the componentwise Hölder approach are two-fold, compared to the KI-based MPC of Chapter 3. First, the enhanced learning method potentially provides less conservative estimation errors (see Fig. 4.1 on page 79), yielding more accurate predictions. This leads to an improvement of the closed-loop performance of the controlled system.

Second, note that the recursion used for the set of tightened constraints (cf. equation (4.12)) is obtained using the componentwise Hölder metric, in contrast to the standard version presented in Section 3.3. Hence, recalling Corollary 4.2 (on page 72), even if the maximum prediction errors were the same (which in general they are not), the back-off of the set of tightened constraints are less conservative ($\mathcal{R}_N^{\mathrm{CHoKI}} \subseteq \mathcal{R}_N^{\mathrm{KI}}$), thus yielding larger regions of feasibility of the proposed controller. This double benefit of the proposed method will be illustrated in the following section, using two case studies.

4.5 Case studies

This section considers a couple of case studies to illustrate the performance of the proposed method.

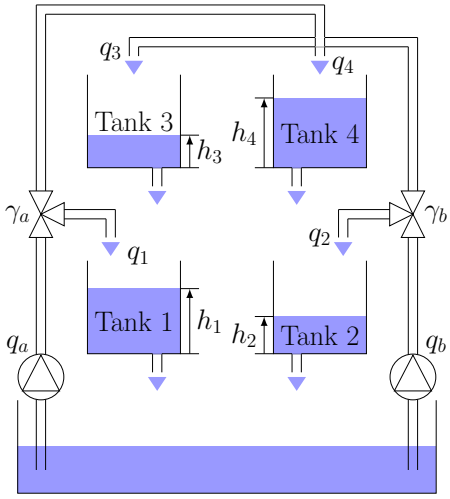


Figure 4.3: Quadruple-tank process scheme.

4.5.1 Quadruple tank process

The first system considered is the quadruple-tank process described in [9, 68]. The process consists of four tanks, where the two on top discharge on the inferior ones. The tanks are fed with two pumps, which send two flows represented by q_a and q_b. These flows enter two three-ways valves, which divide each flow into two branches, determined by the fractions γ_a and γ_b. Thus, $q_a\gamma_a$ goes into the tank number one, $q_a(1-\gamma_a)$ into tank four, $q_b\gamma_b$ into the tank number two and $q_b(1-\gamma_b)$ into tank number three. Tank number three discharges into tank number one, and the fourth one into the second one, as represented in Figure 4.3.

There are two control inputs, the flows q_a and q_b ($\mathrm{m}^3\,\mathrm{h}^{-1}$). The heights of the tanks are denoted $h_i(\mathrm{m}), i \in \mathbb{I}_1^4$. The outputs of the system are the heights of the two lower tanks, i.e., h_1 and h_2. The dynamics of the plant are nonlinear, and are well-modelled by the following set of differential equations:

$$A_1\frac{dh_1(t)}{dt} = -a_1\sqrt{2gh_1(t)} + a_3\sqrt{2gh_3(t)} + \gamma_a\frac{q_a(t)}{3600}, \tag{4.22a}$$

$$A_2\frac{dh_2(t)}{dt} = -a_2\sqrt{2gh_2(t)} + a_4\sqrt{2gh_4(t)} + \gamma_b\frac{q_b(t)}{3600}, \tag{4.22b}$$

$$A_3\frac{dh_3(t)}{dt} = -a_3\sqrt{2gh_3(t)} + (1-\gamma_b)\frac{q_b(t)}{3600}, \tag{4.22c}$$

Param.	Definition	Value	Units
A	Area of the four tanks	0.03	m^2
a_1	Eq. area of the hole of tank 1	1.31×10^{-4}	m^2
a_2	Eq. area of the hole of tank 2	1.51×10^{-4}	m^2
a_3	Eq. area of the hole of tank 3	9.27×10^{-5}	m^2
a_4	Eq. area of the hole of tank 4	8.82×10^{-5}	m^2
γ_a	Fraction of three-ways valve a	0.3	-
γ_b	Fraction of three-ways valve b	0.4	-
g	Gravity acceleration	9.8	$\mathrm{m\,s}^{-2}$

Table 4.1: Parameters of the quadruple tank process.

$$A_4 \frac{dh_4(t)}{dt} = -a_4\sqrt{2gh_4(t)} + (1-\gamma_a)\frac{q_a(t)}{3600}, \tag{4.22d}$$

where A_i (m^2) denotes the area of tank i and a_i (m^2) the equivalent area of the hole of tank i.

The parameters of the model are given in Table 4.1. It is also assumed that the height sensors have a 2% measuring error. The error is generated randomly for each measurement using an uniform distribution.

The constraints in the inputs are $0 \leq q_{a,b} \leq 2.6\,\mathrm{m}^3\,\mathrm{h}^{-1}$, and the constraints in the heights are given by $0 \leq h_1 \leq 1.25\,\mathrm{m}$ and $0 \leq h_2 \leq 1.42\,\mathrm{m}$. Note that this is a really interesting problem from the academical control point of view, since it is a MIMO system with high coupling, which induces non-minimal phase zeros.

First, the set of equilibrium points is estimated using a set of steps in the inputs from $q^{\min}$ to $q^{\max}$ with increments of $0.1\,\mathrm{m}^3\,\mathrm{h}^{-1}$, where each step is long enough to reach a steady sate. The result is shown in Figure 4.4. In addition to obtaining the equilibrium points of the system, this test is used to adjust the sampling time, which is set to $\tau_s/5 = 5\,\mathrm{s}$, where τ_s stands for the mean settling time of the sequence of steps applied. An equilibrium point is chosen to be the reference operating point, i.e. $h^{\mathrm{ref}} = [0.65\ 0.65](\mathrm{m})$, and $q^{\mathrm{ref}} = [1.63\ 1.99](\mathrm{m}^3\,\mathrm{h}^{-1})$.

After defining the set of equilibrium points, a set of experiments is carried out to obtain the data sets. The experiments are designed using the methodologies presented in Chapter 2 to identify the dynamics of a system within a workspace: a sequence of chirp signals covering the workspace are applied to generate the raw data set containing the trajectories of heights and flows, $\mathcal{D}_{\mathrm{raw}}$, with $N_{\mathcal{D}} = 36000$. In addition, several tests with random input

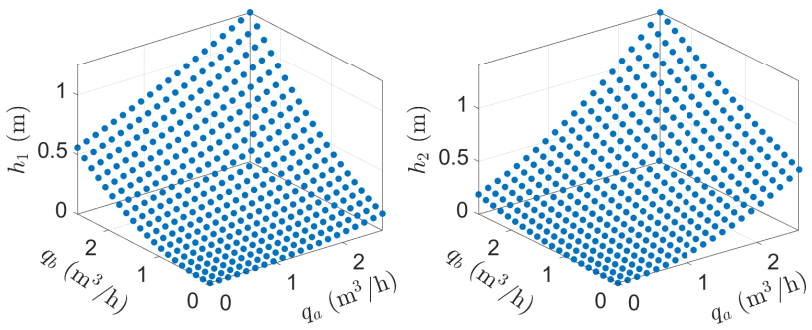


Figure 4.4: Set of equilibrium points of the quadruple tank process.

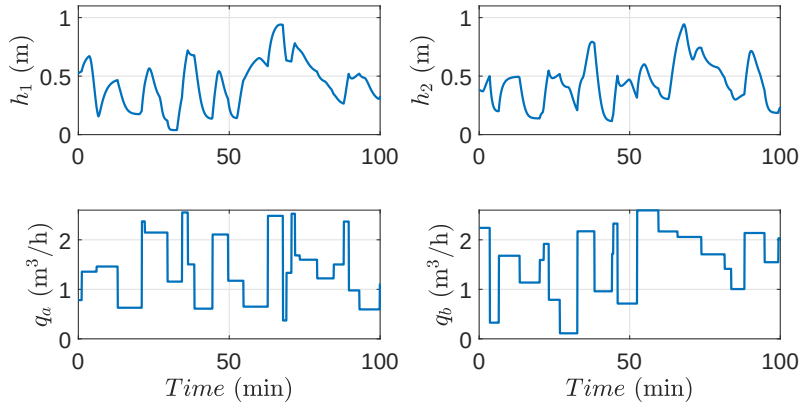


Figure 4.5: Part of the PRS applied to obtain data sets for cross-validation.

signals are carried out in order to obtain data sets for cross-validation, $\mathcal{D}_{\text{test}}$, (see Figure 4.5).

The values of all the signals are scaled between 0 and 1. The regression vector $w(k)$ is constructed for different values of n_a and n_b. The prediction horizon is set to $N = 5$. Cross validation tests are used to estimate the parameter $\mathcal{L}$, such that the sum of the areas of $\mathcal{Y}_j$ is maximized, i.e., setting $g(\cdot,\cdot,\cdot)$ in equation (4.7) (on page 74) to

$$g(\Theta, \mathcal{D}, \mathcal{D}_{\text{test}}) = -\sum_{j=1}^{N} \|\mathcal{Y}_j\|, \tag{4.23}$$

with $\tau_{\mathcal{L}} = 0$ and $\eta = 0.04\,\mathbb{1}_2$. The minimum is obtained for $n_a = 3$

and $n_b = 1$, yielding $\mu = [0.031\ 0.032]$. [1] The set of tightened constraints is obtained considering the Lipschitz case. The resulting $\mathcal{L}$ is

$$\mathcal{L} = \begin{bmatrix} 2 & 0.8 & 5.3 & 1.3 & 3.7 & 1.2 & 7.9 & 2.4 & 5.8 & 1.2 & 2.5 & 4.3 \\ 0.4 & 1.9 & 1.5 & 6 & 1.3 & 4.4 & 1.9 & 7.6 & 1.9 & 6.2 & 2.9 & 3.5 \end{bmatrix}.$$

The solver chosen for the optimization problem is MATLAB®'s `fmincon`. The stage and terminal costs are defined as

$$\ell(x, u) = \|x - x^{\text{ref}}\|_Q^2 + \|u - u^{\text{ref}}\|_R^2$$
$$V_f(x) = \|x - x^{\text{ref}}\|_P^2,$$

where $Q = 100 I_{10}$, $R = I_2$, and $\lambda = 10$. The terminal matrix P is obtained solving an LQR for a linearisation of the CHoKI model around the reference point.

The initial state is set to $h_i = 0.45\,\text{m}, i \in \mathbb{I}_1^4$, and the proposed CHoKI-based controller is applied to the system. In order to compare the results, the same setup is applied to other two controllers, whose models are (i) the ideal state-feedback set of ODEs and (ii) the KI used in Chapter 3.

The results are shown in Figure 4.6, for 100 simulations subject to random noise. Besides, the performance of these simulations is measured according to

$$\Phi = \sum_{i=1}^{t_{\text{sim}}} \ell(x(i), u(i)), \tag{4.24}$$

which is compared in Figure 4.7. Note that the data-based control problem is able to perform in a similar way to the ideal MPC, whereas the standard KI exhibits a worse performance, which illustrates the main properties of CHoKI.

Note that the main advantage of CHoKI with respect to standard KI is not only the improvement of the prediction, which in general leads to better closed-loop performance results, but also the enlargement of the tightened constraints presented in Section 4.3, which implies that the multivariate bound is less conservative. If we used the obtained $\mathcal{D}$ in the standard KI approach presented in Section 3.3, the prediction error would be increased up to $\mu^{\text{KI}} = [0.087\ 0.088]$ [1] (recall that $\mu^{\text{CHoKI}} = [0.031\ 0.032]$). Then, the maximum prediction horizon such that Assumption 3.2 holds, i.e., that $\mathcal{Y}_N$ is not empty, is $N^{\text{KI}} = 2$. The sets $\mathcal{Y}_j$ and $\mathcal{R}_j$ are represented in Figure 4.8, [1] both for the CHoKI and the KI approaches. Note that $\|\mathcal{R}_5^{\text{CHoKI}}\| = 0.014$ while $\|\mathcal{R}_5^{\text{KI}}\| = 54.96$. [1]

[1] Recall that the signals are scaled between 0 and 1.

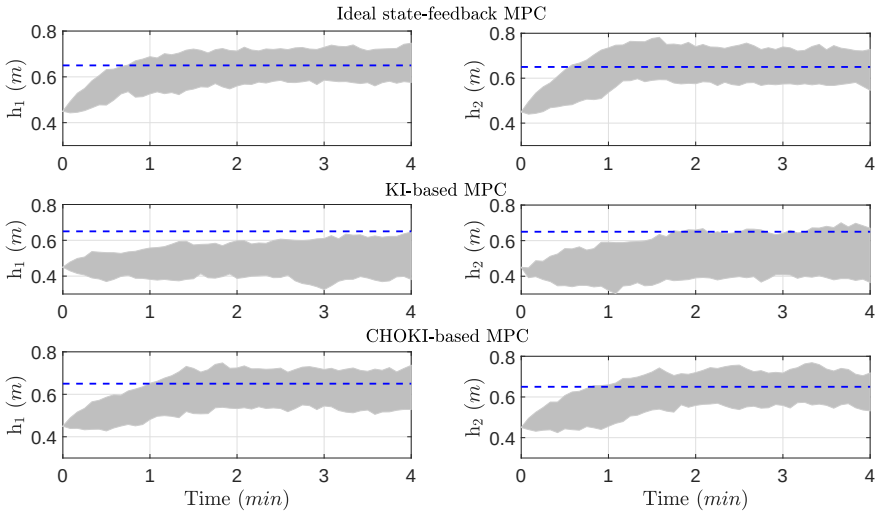


Figure 4.6: A hundred closed-loop simulations of the quadruple-tank process, with three MPCs whose models are (top row) the ideal state-feedback set of ODEs, (middle row) the standard KI and (bottom row) the proposed CHoKI-based MPC.

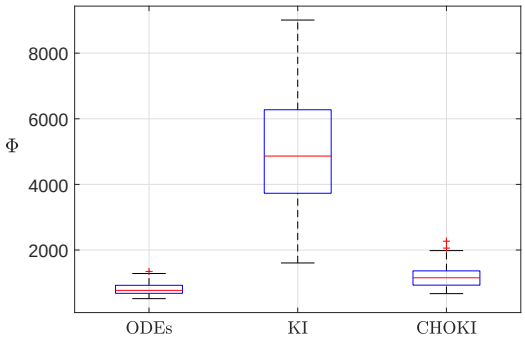
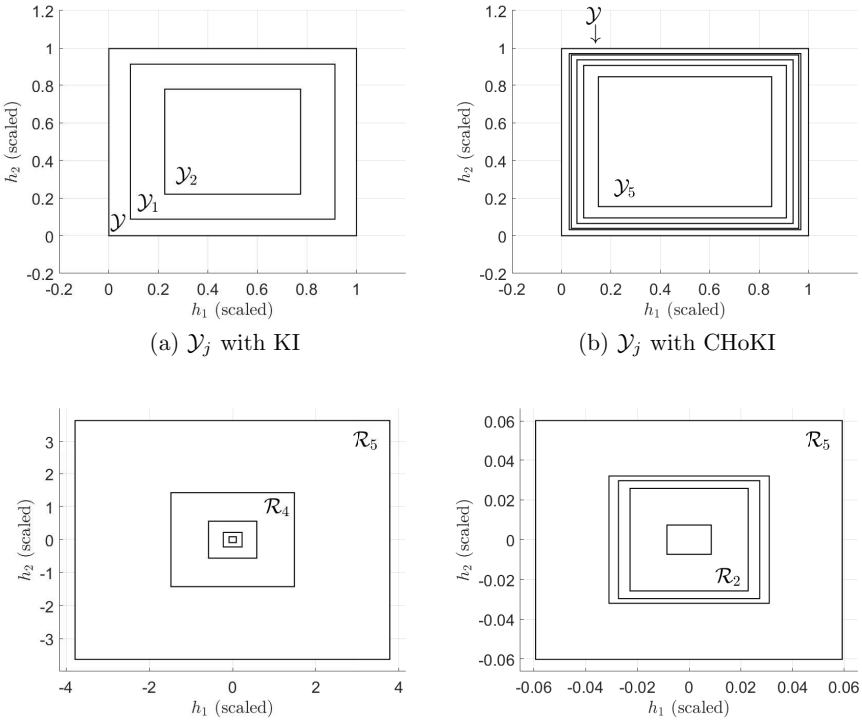


Figure 4.7: Performance index comparison of the three controllers of Fig. 4.6.



(a) $\mathcal{Y}_j$ with KI

(b) $\mathcal{Y}_j$ with CHoKI

(c) $\mathcal{R}_j$ with KI

(d) $\mathcal{R}_j$ with CHoKI

Figure 4.8: Sets $\mathcal{Y}_j$ and $\mathcal{R}_j$, both for the KI and the CHoKI approaches [1] in the quadruple-tank case. Note that $\mathcal{Y}_j$ is empty for $j \geq 3$ in the KI setup, and notice the different scales between the sets $\mathcal{R}_j$ in figures (c) and (d).

4.5.2 Isothermal chemical reactor

An isothermal chemical reactor (ICR) is considered next. It was studied in [132]. The following consecutive-competitive reactions take place:

$$P_0 + B \rightarrow P_1$$

$$P_1 + B \rightarrow P_2.$$

The inputs considered are the flows of P_0 and B, denoted q_a and q_b, respectively; and the outputs are the measured concentrations of P_1 and P_2. The dimensionless equations that model this system are

$$\frac{dP_0(t)}{dt} = q_a - P_0 - k_1 P_0 B \tag{4.25a}$$

$$\frac{dB(t)}{dt} = q_b - B - k_1 P_0 B - k_2 B P_1 \tag{4.25b}$$

$$\frac{dP_1(t)}{dt} = -P_1 + k_1 P_0 B - k_2 B P_1 \tag{4.25c}$$

$$\frac{dP_2(t)}{dt} = -P_2 + k_2 B P_1, \tag{4.25d}$$

in which the parameters k_1 and k_2 are set to 1 and 0.4, respectively. The constraints in the inputs are given by $5 \leq q_a, q_b \leq 10$, and in the outputs by $1.59 \leq P_1 \leq 3.82$ and $0.71 \leq P_2 \leq 2.50$. The reference point is given by $y^{\text{ref}} = [3.09\ 1.66]$ and $u^{\text{ref}} = [8.28\ 7.76]$. We proceed the same way as in the previous case study, obtaining the data sets, the regressors, and the prediction error, which is minimized for $n_a = 2$ and $n_b = 0$. The costs are also defined the same way, with $Q = 100 I_6$, $R = 10 I_2$ and $\lambda = 100$.

The resulting MPC, applied to 100 simulations, is shown in Figure 4.9(a). Noticing the overshoot in the concentration of P_1, a second set of experiments is carried out, in order to demonstrate the ability of the proposed controller to maintain robust satisfaction of the hard constraints. To this end, the upper constraint of P_1 is lowered to from 3.82 to 3.3, and the same controller is applied. Note that the MPC is still able to satisfy the constraint.

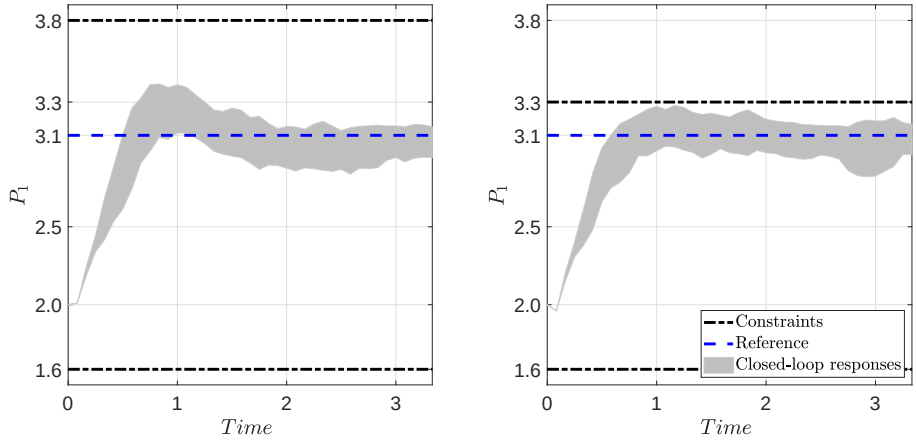


Figure 4.9: Illustration of the ability of the CHoKI-based MPC to satisfy hard constraints in the ICR case study.

Chapter 5

Online learning constrained MPC based on double prediction

5.1 Introduction

The approaches presented so far are based on using a fixed data set to design the model, usually collected offline from ad hoc identification experiments, and do not consider the possibility of improving the predictions online. The terms *online* and *learning* are particularly suitable when considering data-based control, since during the operation of the closed-loop plant, access to new observations of the system becomes available trough feedback. It seems intuitive to add these data points to the database, in order to improve future predictions. According to T. Mitchell's definition [114], an estimation algorithm is learning if it improves its performance -relative to some metric- with increasing exposure to data. These new data points could be used to improve the estimations in real time. The improvement of the performance has been validated in multiple works within the last decade [43, 122]. For the KI methods, the conditions under which it can be guaranteed that the prediction error vanishes (up to a factor of the level of observational error) are presented in Appendix B (see page 140).

However, when considering online learning-based predictive control settings, especially with unknown Lipschitz constants, there are few results that explicitly consider robustness issues, particularly when hard constraints are taken into account. Indeed, the design of data-based predictive controllers

that include flexible online learning capabilities and guarantees of robust stability and constraint satisfaction is still an open problem [63].

As a step towards addressing this challenge, this chapter extends the results presented in chapters 2 and 3, which were based on offline model identification, to an online learning framework. In particular, an online learning MPC based on a double prediction model is presented (similar in spirit to the linear controllers proposed by Aswani et al. [13]). The proposed MPC guarantees robust constraint satisfaction and stability, by means of a set of tightened constraints, an appropriate terminal cost and an ad hoc designed data update policy. The controller proposed is not based on a terminal region, following a design procedure similar to the controller presented in Chapter 3. This avoids the calculation of invariant sets in the controller design, which is in general a hard task.

The results presented in this chapter were published in [101].

5.2 Double prediction framework

This chapter considers the following online learning setup: in a first offline design stage, an initial data set $\mathcal{D}(0)$ is available, obtained via specific experiments or given historical data. Once the controller is designed and applied to the plant, access to new measurements $\tilde{y}(k) = \tilde{f}(w(k-1)) = \tilde{f}(x(k-1), u(k-1))$ becomes available during its operation, allowing one to update the data set up to the current time step k, yielding $\mathcal{D}(k)$. An update method $\mathcal{D}(k-1) \to \mathcal{D}(k)$ will be presented, heuristically tailored to the proposed learning method and control law. Similarly to Aswani et al. [13], the proposed controller will use two different prediction models, one for safety and one for performance. For simplicity, this chapter will consider the Lipschitz case, that is, making the Hölder exponent $p = 1$.

The safe model [1]

$$\hat{y}_s(k+1) = \hat{\mathfrak{f}}_s(x(k), u(k)) = \hat{\mathfrak{f}}(x(k), u(k); L, \mathcal{D}(0)) \tag{5.1}$$

is obtained applying KI with $\mathcal{D}(0)$, and where $L \in \mathbb{R}^{n_y}$ is obtained using the LACKI method. It is derived offline from the initial data available before closing the loop. A state-space version can be obtained using equation (2.8) (on page 33), denoted $\hat{x}_s(k+1) = \hat{F}_s(x(k), u(k))$.

[1] The safe model $\hat{\mathfrak{f}}_s$ must not be confused with the smoothed kinky inference predictor presented in Chapter 2, denoted $\hat{\mathfrak{f}}_S$.

As in previous chapters, the prediction error bound will be obtained in practice via validation tests, as it is customary in identification. In order to prove robust stability, it is assumed that a guaranteed bound of the estimation error is determined, as stated in the following assumption.

Assumption 5.1. *The prediction error of the safe model $d_s(\cdot) \in \mathbb{R}^{n_y}$, which depends on the data set $\mathcal{D}(0)$ and the estimated Lipschitz constant L, is bounded by some known $\mu \in \mathbb{R}^{n_y}$. That is, for all admissible (x, u),*

$$d_s(k) = |y(k+1) - \hat{\mathfrak{f}}_s(w(k); L, \mathcal{D}(0))| \leq \mu. \tag{5.2}$$

The safe model and its bound d_s are necessary to prove robust constraint satisfaction, and therefore, safety of the controlled system. Note that $d_s(\cdot)$ accounts not only for the error induced by the underestimation of the Lipschitz constant, but also for the lack of information on the data set, as well as for the effect of the noise in the sampled data.

Property 5.1. *According to Assumption 5.1, the real output $f(w)$ lies in a ball centered in $\hat{\mathfrak{f}}_s(w)$ and width μ, which is defined as the set*

$$\mathcal{Y}_s(x, u) = \left\{ \hat{\mathfrak{f}}_s(x, u) \oplus \mathcal{B}(\mu) \right\}. \tag{5.3}$$

The online model

$$\hat{y}_{\text{on}}(k+1) = \hat{\mathfrak{f}}_{\text{on}}(x(k), u(k)) = \text{Proj}_{\mathcal{Y}_s(x(k),u(k))}\big(\hat{\mathfrak{f}}(x(k), u(k); L, \mathcal{D}(k))\big) \tag{5.4}$$

provides the prediction with the updated data set $\mathcal{D}(k)$, as long as it is contained in $\mathcal{Y}_s$. If not, a guaranteed prediction is obtained by projection, according to Property 5.1. The state-space online model is denoted and obtained as $\hat{x}_{\text{on}}(k+1) = \hat{F}_{\text{on}}(x(k), u(k))$.

Since new information is added to the data set, it is sensible to think that it will provide better predictions. In general, the estimation error of the online model, which will be denoted $d_{\text{on}}(k)$, decreases as the density of the data set increases. Thus, this model will be used to enhance the closed-loop performance of the plant. However, the guarantee on the bound of the prediction error μ might not be valid for $\hat{\mathfrak{f}}_{\text{on}}$, and then it would not be suitable for safety. This is the reason why a double model framework is used.

5.3 Online learning-based MPC

In this section, the proposed MPC, the required design ingredients and the stability analysis are presented. Additionally, we prove that the closed-loop

system is input-to-state stable w.r.t. the estimation error of the online prediction model, $d_{\text{on}}(k)$.

Analogously to Chapter 3, in order to ensure the satisfaction of hard constraints in the outputs (i.e., $y(k) \in \mathcal{Y} \forall k$), the proposed controller is based on a set of tightened constraints, defined as

$$\mathcal{Y}_j = \mathcal{Y} \ominus \mathcal{B}(d_j(\mu)), \tag{5.5}$$

where $d_j(c_1)$ is obtained by the recursion

$$d_j(c_1) = \sum_{i=1}^{j} c_i(c_1), \tag{5.6a}$$

$$c_{j+1}(c_1) = L r_j(c_1), \tag{5.6b}$$

$$r_j(c_1) = \sum_{i=\sigma_j}^{j} \|c_i(c_1)\|, \tag{5.6c}$$

with $\sigma_j = \max(1, j - n_a)$, for all $j \in \mathbb{I}_1^N$.

The proposed predictive controller is defined in the following optimization problem, denoted $P_N(x(k); L, \mathcal{D}(0), \mathcal{D}(k))$.[2,3]

$$\text{Online} \begin{cases} \min\limits_{\mathbf{u}} & V_N(k) = \sum\limits_{j=0}^{N-1} \ell(\hat{y}_{\text{on}}(j|k), u(k+j)) + \lambda V_f(\hat{x}_{\text{on}}(N|k)) & \text{(5.7a)} \\ \text{s.t.} & \hat{x}_{\text{on}}(0|k) = x(k) & \text{(5.7b)} \\ & \hat{x}_{\text{on}}(j+1|k) = \hat{F}_{\text{on}}(\hat{x}_{\text{on}}(j|k), u(k+j)),\ j \in \mathbb{I}_0^{N-1} & \text{(5.7c)} \\ & \hat{y}_{\text{on}}(j|k) = M \hat{x}_{\text{on}}(j|k),\ j \in \mathbb{I}_0^{N-1} & \text{(5.7d)} \end{cases}$$

$$\text{Safe} \begin{cases} & \hat{x}_s(0|k) = x(k) & \text{(5.7e)} \\ & \hat{x}_s(j+1|k) = \hat{F}_s(\hat{x}_s(j|k), u(k+j)),\ j \in \mathbb{I}_0^{N-1} & \text{(5.7f)} \\ & \hat{y}_s(j|k) = M \hat{x}_s(j|k),\ j \in \mathbb{I}_1^{N} & \text{(5.7g)} \\ & u(k+j) \in \mathcal{U},\ j \in \mathbb{I}_0^{N-1} & \text{(5.7h)} \\ & \hat{y}_s(j|k) \in \mathcal{Y}_j,\ j \in \mathbb{I}_0^{N-1}, & \text{(5.7i)} \end{cases}$$

where $x(k)$ is the current state, $\mathbf{u} = (u(0), ..., u(N-1))$ is the future input trajectory, $\hat{x}_{\text{on}}(j|k)$, $\hat{y}_{\text{on}}(j|k)$, $\hat{x}_s(j|k)$ and $\hat{y}_s(j|k)$ are the state and input

[2]The abbreviation for $P_N(x(k); L, \mathcal{D}(0), \mathcal{D}(k))$ will be denoted $P_N(k)$.

[3]For the sake of conciseness we omit the dependence of the functions V_N, ℓ and V_f with L and $\mathcal{D}(0), \mathcal{D}(k)$.

trajectories predicted from the current state using the online and safe models, respectively. The solution, applied using a receding horizon technique, yields the control law

$$u(k) = \kappa_{\text{MPC}}(x(k); L, \mathcal{D}(0), \mathcal{D}(k)). \tag{5.8}$$

Note how the optimization problem reflects the purpose of each model: the performance index is calculated using the online model, while the constraints must be satisfied by the predicted trajectory using the safe model. Notice that there is no terminal constraint in the optimization problem. Instead, stability is achieved by means of a weighting factor $\lambda \geq 1$ for the terminal cost, following the method presented in Chapter 3. To ensure robust stability and constraint satisfaction, the assumptions of Chapter 3 must hold, subject to some modifications to account for the double prediction framework. Hence, they are rewritten here.

Assumption 5.2.

1. *The stage cost function $\ell(y, u)$ is a continuous positive definite function for all $y \in \mathcal{Y}$ and $u \in \mathcal{U}$ such that $\ell(y, u) \geq \alpha_y(\|y\|) + \alpha_u(\|u\|)$ and*

$$|\ell(y_1, u) - \ell(y_2, u)| \leq \theta_\ell(\|(y_1, u) - (y_2, u)\|),$$

   for certain $\mathcal{K}$-functions $\theta_\ell, \alpha_y, \alpha_u$.

2. *There exists a local control law $u = \kappa_f(x)$, a terminal cost function V_f and a set $\Omega_\gamma = \{x : V_f(x) \leq \gamma\} \subseteq \mathbb{R}^{n_x}$, for $\gamma > 0$ such that for all $x \in \Omega_\gamma$ the following conditions holds:*

   (a)

$$\begin{aligned} \kappa_f(x) &\in \mathcal{U}, && \text{(5.9a)} \\ Mx \oplus \mathcal{B}(a_N) &\subseteq \mathcal{Y}_N, && \text{(5.9b)} \end{aligned}$$

   where a_j is given by the recursion $a_{j+1} = L\|a_j\| + \mu$, with $a_1 = \mu$.

   (b) V_f is a continuous positive definite function such that for certain $\mathcal{K}$-functions $\alpha_f, \beta_f, \theta_f$, $|V_f(x_1) - V_f(x_2)| \leq \theta_f(\|x_1 - x_2\|)$ and

$$\begin{aligned} \alpha_f(\|x\|) \leq V_f(x) \leq \beta_f(\|x\|), && \text{(5.10a)} \\ V_f(\hat{F}_{\text{on}}(x, \kappa_f(x))) - V_f(x) \leq -\ell(Mx, \kappa_f(x)), && \text{(5.10b)} \end{aligned}$$

Remark 5.1. *Note that (5.9b) implicitly states a condition on the estimation error bound μ. It can be proven that if μ is small enough to ensure that $\mathcal{B}(a_N) \oplus \mathcal{B}(d_N(\mu)) \subseteq \epsilon\mathcal{Y}$, for $0 < \epsilon < 1$, then (5.9b) is solvable.*

Remark 5.2. *Equation (5.10b) in Assumption 5.2 requires that the Lyapunov condition holds for the online prediction model. Since the terminal ingredients must be computed offline, the online model is not available for this design. Taking into account that the safe model can be regarded as a sampled subset of the online model, since $\mathcal{D}(0) \subseteq \mathcal{D}(k)$, the online model is contained in a difference inclusion centered in the safe model and given by the estimation error of the LACKI method, which is explicitly known. Therefore, robust design methods can be used to calculate the terminal ingredients, satisfying Assumption 5.2.*

Definition 5.1. *Making use of the definition of $c_j(c_1)$ and $r_j(c_1)$ in (5.6) and of Assumption 5.2, the following function, which is a $\mathcal{K}$-function for every input, is defined:*

$$\theta_a(c_1) = \sum_{j=0}^{N-1} \theta_\ell(\|c_{j+1}(c_1)\|) + \theta_f(r_{N+1}(c_1)). \tag{5.11}$$

Assumption 5.3. *The set*

$$\Upsilon = \{x : \ell(Mx, 0) \leq \theta_a(2\mu)\} \tag{5.12}$$

is contained in Ω_γ.

Definition 5.2. *ϕ is a positive constant such that*

$$\ell(Mx, 0) > \phi, \ \forall x \notin \Omega_\gamma. \tag{5.13}$$

Similarly to Chapter 3, it can be proven that $\phi \geq \theta_a(2\mu)$.

The region of feasibility of the problem (5.7) is denoted X_N, and a level set of its optimal cost is defined as:

$$\Gamma = \{x \in X_N : V_N^*(x) \leq N\phi + \lambda\gamma\}. \tag{5.14}$$

The control law is given in Algorithm 1, where the updating policy of the online data set guarantees closed-loop robust stability. At each time step k, the online data set may be updated with the current data point, that is, $\mathcal{D}(k) = \mathcal{D}(k-1) \cup (y(k), w(k-1))$, subject to the following triple updating criteria:

1. The current data point must not be *close* to any point already contained in $\mathcal{D}(k-1)$, for a given threshold $\tau_t \geq 0$. This rule prevents the

cardinality of $\mathcal{D}$ from becoming large, yielding the method computationally expensive. The policy is given by the Boolean function:

$$\pi_{\text{exp}}(w, \mathcal{D}) = \left(\min_{w_i \in \mathcal{W}_{\mathcal{D}}} \|w - w_i\| > \tau_t \right).$$

A reasonable estimate for τ_t is the noise level $\bar{\mathfrak{e}}$.

2. Adding the new data point must not increase the cost, calculated with the shifted input sequence which, together with the control policy and update rule, is defined in Algorithm 1.

$$\pi_V(y, w, \mathcal{D}) = \left(\bar{V}_N(x, \bar{\mathbf{u}}; \bar{\mathcal{D}}) \leq \bar{V}_N(x, \bar{\mathbf{u}}; \mathcal{D}) \right).$$

3. The new candidate $(y(k), w(k-1))$ must be consistent with the Lipschitz constant. This rule is optional, included so that the online model maintains the properties of the LACKI method (see Appendix B).

$$\pi_L(y, w, L, \mathcal{D}) = \left(\max_{i \in \mathbb{I}_1^{N_{\mathcal{D}}}} \frac{|y - y_i| - 2\bar{\mathfrak{e}}}{\|w - w_i\|} \leq L \right).$$

Hence, the aggregated update rule is the Boolean function

$$\pi(y, w, L, \mathcal{D}) = \pi_{\text{exp}} \wedge \pi_V \wedge \pi_L. \tag{5.15}$$

In the following theorem it is proven that the proposed control algorithm guarantees that the closed-loop system is input-to-state stable (ISS).

Theorem 5.1. *Consider that Assumptions 5.1-5.3 hold, and let $\kappa_{\text{MPC}}(x)$ be the control law derived from the solution of $P_N(k)$ applied using Algorithm 1. Then, for any feasible state $x(0) \in \Gamma$, the system controlled by the control law $u(k) = \kappa_{\text{MPC}}(x(k))$ is input-to-state stable w.r.t. the estimation error $d_{\text{on}}(k)$, and the constraints are fulfilled along the operation, i.e., $y(k) \in \mathcal{Y}$ and $u(k) \in \mathcal{U}$, $\forall k$.*

Proof.

In this proof, different predicted trajectories will be considered, as shown in Figure 5.1. These trajectories are obtained predicting from $x(k)$ with $\mathbf{u}^*(k)$, or from $x(k+1)$ with $\bar{\mathbf{u}}(k+1)$, and using the safe or the online models.

It is first proven that Γ is an invariant set of the closed-loop system, i.e., if $x(k) \in \Gamma$, then $x(k+1) \in \Gamma$. To this end, as it is standard in MPC proofs,

Algorithm 1 Online update and control law.

while automatic control is on **do**
  Read $y(k)$
  $x(k) \leftarrow (y(k), \ldots, y(k-n_a), u(k-1), \ldots, u(k-n_b))$ ▷ State
  function SHIFTED SEQUENCE($\bar{\mathbf{u}}$) ▷ Shifted Sequence
    $\bar{u}(j) \leftarrow u^*(j),\ j \in \mathbb{I}_0^{N-2}$
    $\bar{u}(N) \leftarrow \kappa_f(\hat{x}_{\text{on}}(N|k-1))$
    return $\bar{\mathbf{u}}$
  end function
  $\bar{\mathcal{D}}(k) \leftarrow \mathcal{D}(k-1) \cup (y(k), w(k-1))$ ▷ Updated data set
  if $\pi(y(k), w(k-1), L, \mathcal{D}(k-1))$ **then**
    $\mathcal{D}(k) \leftarrow \bar{\mathcal{D}}(k)$ ▷ Update
  else
    $\mathcal{D}(k) \leftarrow \mathcal{D}(k-1)$ ▷ Do not update
  end if
  $\mathbf{u}^*(k) \leftarrow \kappa_{\text{MPC}}(x(k); L, \mathcal{D}(0), \mathcal{D}(k))$ ▷ Solution of $P_N(k)$
  $u(k) \leftarrow u^*(0)$ ▷ Receding horizon
  Apply $u(k)$ to the system
end while

it is proven that the shifted trajectory presented in Algorithm 1 is feasible for $x(k+1)$.

Provided that $x(k) \in \Gamma$, due to its definition in equation (5.14), it follows that

$$V_N^*(x(k)) \leq N\phi + \lambda\gamma, \tag{5.16}$$

and therefore it can be proven [83] that

$$\hat{x}_{\text{on}}(N|k) \in \Omega_\gamma. \tag{5.17}$$

Feasibility of the shifted sequence $\bar{\mathbf{u}}(k+1)$ is proven for any $x(k) \in \Gamma$, showing that

(i) $\bar{u}(k+j+1) \in \mathcal{U},\ \forall j \in \mathbb{I}_0^{N-1}$.

(ii) $\hat{y}_s(j|k+1) \in \mathcal{Y}_j,\ \forall j \in \mathbb{I}_0^{N-1}$.

Constraint (i) holds because $\mathbf{u}^*(k) \in \mathcal{U}$, and since $\hat{x}_{\text{on}}(N|k) \in \Omega_\gamma$, because of equation (5.9a), the shifted sequence defined in Algorithm 1 is feasible, i.e., $\bar{u}(k+j+1) \in \mathcal{U},\ \forall j \in \mathbb{I}_0^{N-1}$.

To address (ii), the following lemmas are used:

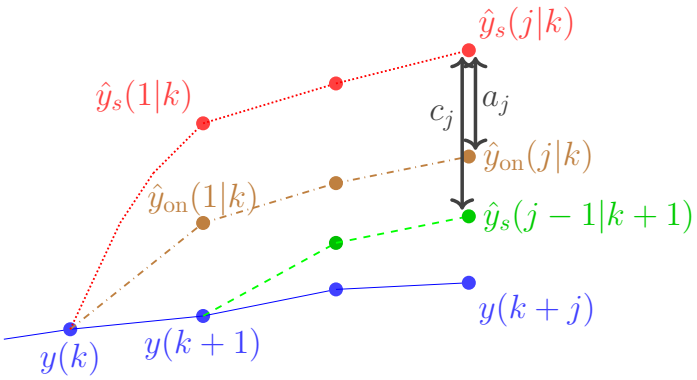


Figure 5.1: Propagation of the predictions.

Lemma 5.1. *The following inequalities hold:*

$$|\hat{y}_{\text{on}}(j-1|k+1) - \hat{y}_{\text{on}}(j|k)| \leq c_j(d_{\text{on}}(k)), \tag{5.18a}$$
$$\|\hat{x}_{\text{on}}(j-1|k+1) - \hat{x}_{\text{on}}(j|k)\| \leq r_j(d_{\text{on}}(k)), \tag{5.18b}$$

where $c_j(c_1)$ and $r_j(c_1)$ are obtained from the recursion (5.6).

The bounds (5.18) hold for the safe model, taking $c_1 = \mu$ instead of $d_{\text{on}}(k)$.

Proof. Given the triangular inequality that applies to the definition of the state vector x,

$$\|x(k+1) - \hat{x}_{\text{on}}(1|k)\| \leq \|y(k+1) - \hat{y}_{\text{on}}(1|k)\| = \|d_{\text{on}}(k)\| = \|c_1\| = r_1. \tag{5.19}$$

Assuming that $r_{j-1}(c_1)$ and $c_{j-1}(c_1)$ are known, we can derive, from the Lipschitz continuity of $\hat{\mathfrak{f}}_{\text{on}}$, that

$$\begin{aligned}
&|\hat{y}_{\text{on}}(j-1|k+1) - \hat{y}_{\text{on}}(j|k)| \\
&\quad = |\hat{\mathfrak{f}}_{\text{on}}(\hat{x}_{\text{on}}(j-2|k+1), u(k+j-1)) - \hat{\mathfrak{f}}_{\text{on}}(\hat{x}_{\text{on}}(j-1|k), u(k+j-1))| \\
&\quad \leq L\|\hat{x}_{\text{on}}(j-2|k+1) - \hat{x}_{\text{on}}(j-1|k)\| \\
&\quad \leq L r_{j-1} = c_j.
\end{aligned} \tag{5.20}$$

The error of the estimated state can be bounded as follows, given that $\hat{x}_{\text{on}}(j|k)$

contains $\min(n_a+1, j)$ estimated outputs (and real measurements otherwise)

$$\begin{aligned}\|\hat{x}_{\text{on}}(j-1|k+1) - \hat{x}_{\text{on}}(j|k)\| \\ &\leq \|\hat{y}_{\text{on}}(j-1|k+1) - \hat{y}_{\text{on}}(j|k)\| + \ldots + \|y(k+1) - \hat{y}_{\text{on}}(j|k)\| \\ &\leq \sum_{i=\sigma_j}^{j} \|c_i\| = r_j. \end{aligned} \tag{5.21}$$

□

Note that the same holds if the safe model is used, with $\hat{x}_s$ and $\hat{y}_s$, being $c_1 = \mu$.

Lemma 5.2. *For all $y \in \mathcal{Y}_j$ and all $\Delta y \in \mathcal{B}(c_j(\mu))$, the sets $\mathcal{Y}_j$ are such that $y + \Delta y \in \mathcal{Y}_{j-1}$.*

Proof. First, it is proven that $\mathcal{B}(\delta_1) \oplus \mathcal{B}(\delta_2) \subseteq \mathcal{B}(\delta_1 + \delta_2)$. Indeed, for all $y = y_1 + y_2$, with $y_1 \in \mathcal{B}(\delta_1)$ and $y_2 \in \mathcal{B}(\delta_2)$, we have that

$$\|y\| \leq \|y_1\| + \|y_2\| \leq \delta_1 + \delta_2, \tag{5.22}$$

and thus $y \in \mathcal{B}(\delta_1 + \delta_2)$.

Since (for any c_1) $d_j = c_j + d_{j-1}$, it is inferred that $\mathcal{B}(d_{j-1}) \oplus \mathcal{B}(c_j) \subseteq \mathcal{B}(d_j)$, and then,

$$\mathcal{Y}_j = \mathcal{Y} \ominus \mathcal{B}(d_j(\mu)) \subseteq \mathcal{Y} \ominus \mathcal{B}(d_{j-1}(\mu)) \ominus \mathcal{B}(c_j(\mu)) = \mathcal{Y}_{j-1} \ominus \mathcal{B}(c_j(\mu)). \tag{5.23}$$

Given that $\Delta y \in \mathcal{B}(c_j(\mu))$ and $y \in \mathcal{Y}_j$, it is derived that

$$y + \Delta y \in \mathcal{Y}_j \oplus \mathcal{B}(c_j(\mu)) \subseteq \mathcal{Y}_{j-1} \ominus \mathcal{B}(c_j(\mu)) \oplus \mathcal{B}(c_j(\mu)) \subseteq \mathcal{Y}_{j-1}. \tag{5.24}$$

□

It is known that $\hat{y}_s(j|k) \in \mathcal{Y}_j$ up to $N-1$, and that

$$|\hat{y}_s(j|k) - \hat{y}_s(j-1|k+1)| \leq c_j(\mu). \tag{5.25}$$

Hence, because of Lemma 5.2,

$$\hat{y}_s(j|k+1) \in \hat{y}_s(j+1|k) \oplus \mathcal{B}(c_{j+1}(\mu)) \subseteq \mathcal{Y}_{j+1} \oplus \mathcal{B}(c_{j+1}(\mu)) \subseteq \mathcal{Y}_j.$$

For the last prediction, the fact that $\hat{x}_{\text{on}}(N|k) \in \Omega_\gamma$ implies that the last predicted output $\hat{y}_{\text{on}}(N|k) \in \mathcal{Y}_N$ (see equation (5.9b)).

Another lemma is introduced:

Lemma 5.3. *The following inequalities hold:*

$$|\hat{y}_{\text{on}}(j|k) - \hat{y}_s(j|k)| \leq a_j, \tag{5.26}$$

where a_j is given by the recursion

$$a_{j+1} = L\|a_j\| + \mu, \tag{5.27}$$

with $a_1 = \mu$.

Proof. This lemma is proven by recursion. For $j = 1$ the condition holds, given the definition of $\hat{\mathfrak{f}}_{\text{on}}$ in (5.4):

$$|\hat{y}_{\text{on}}(1|k) - \hat{y}_s(1|k)| \leq a_1 = \mu. \tag{5.28}$$

Then, assuming that a_{j-1} is known, we can obtain.

$$\begin{aligned}
&|\hat{y}_s(j|k) - \hat{y}_{\text{on}}(j|k)| \\
&\quad = |\hat{\mathfrak{f}}_s(\hat{x}_s(j-1|k), u(k+j-1)) - \hat{\mathfrak{f}}_{\text{on}}(\hat{x}_{\text{on}}(j-1|k), u(k+j-1))| && \text{(5.29a)}\\
&\quad \leq |\hat{\mathfrak{f}}_s(\hat{x}_s(j-1|k), u(k+j-1)) - \hat{\mathfrak{f}}_s(\hat{x}_{\text{on}}(j-1|k), u(k+j-1))| \\
&\qquad + |\hat{\mathfrak{f}}_s(\hat{x}_{\text{on}}(j-1|k), u(k+j-1)) - \hat{\mathfrak{f}}_{\text{on}}(\hat{x}_{\text{on}}(j-1|k), u(k+j-1))| && \text{(5.29b)}\\
&\quad \leq L\|\hat{y}_s(j-1|k) - \hat{y}_{\text{on}}(j-1|k)\| + \mu && \text{(5.29c)}\\
&\quad = L\|a_{j-1}\| + \mu = a_j. && \text{(5.29d)}
\end{aligned}$$

□

Given the definition of $c_j(\mu)$ in Lemma 5.1 and a_j in Lemma 5.3, the following bound is obtained:

$$\begin{aligned}
|\hat{y}_{\text{on}}(j|k) - &\hat{y}_s(j-1|k+1)| \\
&\leq |\hat{y}_{\text{on}}(j|k) - \hat{y}_s(j|k)| + |\hat{y}_s(j|k) - \hat{y}_s(j-1|k+1)| \\
&\leq a_j + c_j(\mu). && \text{(5.30)}
\end{aligned}$$

Using Lemma 5.3 and equation (5.30) it follows that the error between the predictions of the safe and the online models is bounded, so

$$|\hat{y}_{\text{on}}(N|k) - \hat{y}_s(N-1|k+1)| \leq a_N + c_N(\mu).$$

Then, because of equation (5.9b),

$$\hat{y}_s(N-1|k+1) \in M\hat{x}_{\text{on}}(N|k) \oplus \mathcal{B}(a_N) \oplus \mathcal{B}(c_N(\mu)) \subseteq \mathcal{Y}_N \oplus \mathcal{B}(c_N(\mu)) \subseteq \mathcal{Y}_{N-1}, \tag{5.31}$$

which completes the proof of (ii).

Given the definition of the cost in (5.7a), the measurements of $y(k)$ and $y(k+1)$, and the optimal and shifted sequences $\mathbf{u}^*(k)$ and $\bar{\mathbf{u}}(k+1)$ (see Algorithm 1), the following equality holds:

$$\begin{aligned} &V_N(\hat{x}_{\text{on}}(1|k), \bar{\mathbf{u}}(k+1); \mathcal{D}(k)) - V_N^*(x(k), \mathbf{u}^*(k); \mathcal{D}(k)) \\ &\quad = \ell(\hat{y}_{\text{on}}(N|k), \kappa_f(\hat{x}_{\text{on}}(N|k))) + V_f\left(\hat{F}_{\text{on}}\left(\hat{x}_{\text{on}}(N|k), \kappa_f\left(\hat{x}_{\text{on}}(N|k)\right)\right)\right) \\ &\qquad - \ell(y(k), u^*(k)) - V_f(\hat{x}_{\text{on}}(N|k)). \end{aligned} \tag{5.32}$$

Since $\hat{x}_{\text{on}}(N|k) \in \Omega_\gamma$ and taking into account (5.10b) the following inequality is obtained:

$$\begin{aligned} V_f\left(\hat{F}_{\text{on}}\left(\hat{x}_{\text{on}}(N|k), \kappa_f\left(\hat{x}_{\text{on}}(N|k)\right)\right)\right) - V_f(\hat{x}_{\text{on}}(N|k)) \\ +\ell(\hat{y}_{\text{on}}(N|k), \kappa_f(\hat{x}_{\text{on}}(N|k))) \ \leq \ 0, \end{aligned} \tag{5.33}$$

which implies that

$$V_N(\hat{x}_{\text{on}}(1|k), \bar{\mathbf{u}}(k+1); \mathcal{D}(k)) - V_N^*(x(k), \mathbf{u}^*(k); \mathcal{D}(k)) \leq -\ell(y(k), u^*(k)). \tag{5.34}$$

In addition,

$$\begin{aligned} &V_N(x(k+1), \bar{\mathbf{u}}(k+1); \mathcal{D}(k)) - V_N(\hat{x}_{\text{on}}(1|k), \bar{\mathbf{u}}(k+1); \mathcal{D}(k)) \\ &\quad = \sum_{j=0}^{N-1} \ell(\hat{y}_{\text{on}}(j|k+1), \bar{u}(k+j+1)) - \sum_{j=0}^{N-1} \ell(\hat{y}_{\text{on}}(j+1|k), \bar{u}(k+j+1)) \\ &\qquad + V_f(\hat{x}_{\text{on}}(N|k)) - V_f\left(\hat{F}_{\text{on}}\left(\hat{x}_{\text{on}}(N|k), \kappa_f\left(\hat{x}_{\text{on}}(N|k)\right)\right)\right). \end{aligned} \tag{5.35}$$

Using Lemma 5.1 and taking into account continuity of ℓ and V_f, there exists certain $\mathcal{K}$-functions θ_ℓ and θ_f and a function θ_a such that

$$\begin{aligned} &V_N(x(k+1), \bar{\mathbf{u}}(k+1); \mathcal{D}(k)) - V_N(\hat{x}_{\text{on}}(1|k), \bar{\mathbf{u}}(k+1); \mathcal{D}(k)) \\ &\qquad \leq \sum_{j=0}^{N-1} \theta_\ell(\|c_{j+1}(d_{\text{on}}(k))\|) + \theta_f(r_{N+1}(d_{\text{on}}(k))) = \theta_a(d_{\text{on}}(k)), \end{aligned} \tag{5.36}$$

Now, merging (5.34) and (5.36) yields

$$V_N(x(k+1), \bar{\mathbf{u}}(k+1); \mathcal{D}(k)) - V_N^*(x(k), \mathbf{u}^*(k); \mathcal{D}(k)) \\ \leq -\ell(y(k), u^*(k)) + \theta_a(d_{\text{on}}(k)). \quad (5.37)$$

Moreover, because of the updating policy of Algorithm 1,

$$V_N\left(x(k+1), \bar{\mathbf{u}}(k+1); D(k+1)\right) \leq V_N\left(x(k+1), \bar{\mathbf{u}}(k+1); D(k)\right), \quad (5.38)$$

and by optimality,

$$V_N^*(x(k+1); D(k+1)) \leq \bar{V}_N(x(k+1), \bar{\mathbf{u}}(k+1); D(k+1)). \quad (5.39)$$

Summing up,

$$V_N^*(x(k+1), \mathbf{u}^*(k+1); D(k+1)) - V_N^*(x(k), \mathbf{u}^*(k); \mathcal{D}(k)) \\ \leq -\ell(y(k), u^*(k)) + \theta_a(d_{\text{on}}(k)). \quad (5.40)$$

Next, recursive feasibility is proven ensuring that $x(k+1) \in \Gamma$, for which it is necessary that $V_N^*(x(k+1); D(k+1)) \leq N\phi + \lambda\gamma$. Consider that $x(k) \notin \Upsilon$. Then, $\ell(y(k), u^*(k)) > \theta_a(2\mu)$, provided that the maximum online prediction error satisfies $d_{\text{on}} \leq 2\mu$, given the definition of $\hat{\mathbf{f}}_{\text{on}}$ in (5.4). Therefore, using (5.16) and (5.40) it is derived that $x(k+1) \in \Gamma$.

If $x(k) \in \Upsilon$, then $x(k) \in \Omega_\gamma$ (Assumption 5.3). From standard arguments of MPC it follows that

$$V_N^*(x(k)) \leq \lambda V_f(x(k)) \leq \lambda\gamma, \quad (5.41)$$

which taking into account $\theta_a(d_{\text{on}}(k)) \leq \phi$ implies that

$$V_N^*(x(k+1), \mathbf{u}^*(k+1); D(k+1)) \leq \phi + \lambda\gamma - \ell(y(k), u(k)) \leq N\phi + \lambda\gamma, \quad (5.42)$$

and hence $x(k+1) \in \Gamma$.

To prove input-to-state stability, (5.40) and the continuity of the stage cost is used to derive that

$$V_N^*(x(k+1); D(k+1)) - V_N^*(x(k); \mathcal{D}(k)) \\ \leq -\alpha_y(\|y(k)\|) - \alpha_u(\|u(k)\|) + \theta_a(d_{\text{on}}(k)). \quad (5.43)$$

Defining

$$W(x(k); \mathcal{D}(k)) = \sum_{j=0}^{n} V_N^*(x(k-j); \mathcal{D}(k-j)), \quad (5.44)$$

with $n = \max(n_a, n_b + 1)$, it follows that

$$\alpha_x(\|x(k)\|) \leq \sum_{j=0}^{n} \alpha_y(\|y(k-j)\|) + \alpha_u(\|u(k-j-1)\|),$$

as in Chapter 2, and hence

$$W(x(k+1);\mathcal{D}(k+1)) - W(x(k);\mathcal{D}(k)) \leq -\alpha_x(\|x(k)\|) + (n+1)\theta_a \left(\max_{j=0,\dots,n} d_{\text{on}}(k-j) \right), \tag{5.45a}$$

$$\alpha_1(\|x\|) \leq W(x(k)) \leq \alpha_2(\|x\|). \tag{5.45b}$$

Hence, $W(x(k))$ is an ISS Lyapunov function. □

The proposed MPC has been proven to be stable and to robustly satisfy the output constraints under certain assumptions. Note that it lacks a terminal constraint, avoiding the calculation of a robust invariant set, unlike what is common in the design of robust MPCs [133]. The performance of the controller is enhanced by the inclusion of fresh data from the operation of the plant, as it will be illustrated in the case study of the following section.

Not only does the behaviour improve, but also the convergence rates are decreased as the updated estimation error decreases, because the closed-loop system is proven to be input-to-state stable with respect to the estimation error of the online model. This error decreases in average with time, for increasing data sets.

If the terminal cost is designed offline with the safe model, the following corollary proves that input-to-state practical stability can still be derived:

Corollary 5.1. *Consider that Assumptions 5.1 and 5.3 hold, and that Assumption 5.2 holds replacing equation (5.10b) by*

$$V_f(\hat{F}_s(x, \kappa_f(x))) - V_f(x) \leq -\ell(Mx, \kappa_f(x)), \tag{5.46}$$

that is, satisfying the Lyapunov condition for the safe model. Let $\kappa_{\text{MPC}}(x)$ be the control law derived from the solution of $P_N(k)$ applied using Algorithm 1. Then, for any feasible state $x(0) \in \Gamma$, the system controlled by the control law $u(k) = \kappa_{\text{MPC}}(x(k))$ is input-to-state practically stable w.r.t. $d_{\text{on}}(k)$ and μ. Besides the constraints are fulfilled along the operation, i.e., $y(k) \in \mathcal{Y}$.

Proof.

The proof is similar to the one of Theorem 5.1, but replacing (5.33) as follows. First, see that from the uniform continuity of V_f and Lemma 5.3 (equation (5.28)), we have that

$$V_f\left(\hat{F}_{\text{on}}\left(\hat{x}_{\text{on}}(N|k), \kappa_f\left(\hat{x}_{\text{on}}(N|k)\right)\right)\right) - V_f\left(\hat{F}_s\left(\hat{x}_{\text{on}}(N|k), \kappa_f\left(\hat{x}_{\text{on}}(N|k)\right)\right)\right)$$

is upper bounded by

$$\theta_f(\|\hat{F}_{\text{on}}\left(\hat{x}_{\text{on}}(N|k), \kappa_f\left(\hat{x}_{\text{on}}(N|k)\right)\right) - \hat{F}_s\left(\hat{x}_{\text{on}}(N|k), \kappa_f\left(\hat{x}_{\text{on}}(N|k)\right)\|\right),$$

which is equal to

$$\theta_f(\|\hat{f}_{\text{on}}\left(\hat{x}_{\text{on}}(N|k), \kappa_f\left(\hat{x}_{\text{on}}(N|k)\right)\right) - \hat{f}_s\left(\hat{x}_{\text{on}}(N|k), \kappa_f\left(\hat{x}_{\text{on}}(N|k)\right)\|\right) \leq \theta_f(\|\mu\|).$$

Taking this into account, (5.33) can be rewritten in this case as

$$V_f\left(\hat{F}_{\text{on}}\left(\hat{x}_{\text{on}}(N|k), \kappa_f\left(\hat{x}_{\text{on}}(N|k)\right)\right)\right) - V_f(\hat{x}_{\text{on}}(N|k)) + \ell(\hat{y}_{\text{on}}(N|k), \kappa_f(\hat{x}_{\text{on}}(N|k))) \leq \theta_f(\|\mu\|), \quad (5.47)$$

Following the subsequent steps, (5.45a) would be rewritten as follows, given $n = \max(n_a, n_b + 1)$:

$$W(x(k+1); D(k+1)) - W(x(k); \mathcal{D}(k)) \leq -\alpha_x(\|x(k)\|) + (n+1)\theta_a\left(\max_{j=0,\ldots,n} d_{\text{on}}(k-j)\right) + \theta_f(\|\mu\|). \quad (5.48)$$

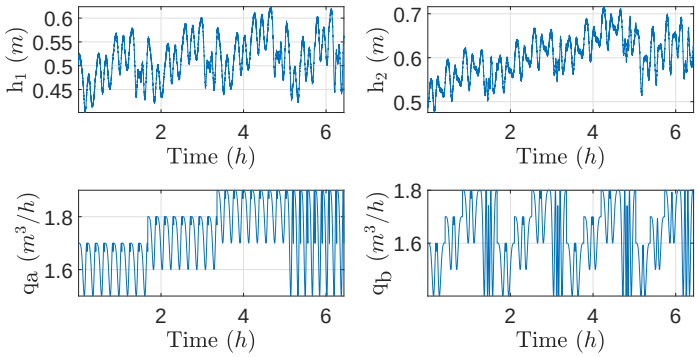
Consequently, the controlled system would be ISS w.r.t. d_{on} and μ. □

Remark 5.3. *The domain of attraction of the controller is defined by the feasibility region Γ in (5.14). This set increases as the weighting factor λ increases, as proven in Chapter 2. Hence, λ can be chosen arbitrarily big in order to enlarge the domain of attraction of the MPC.*

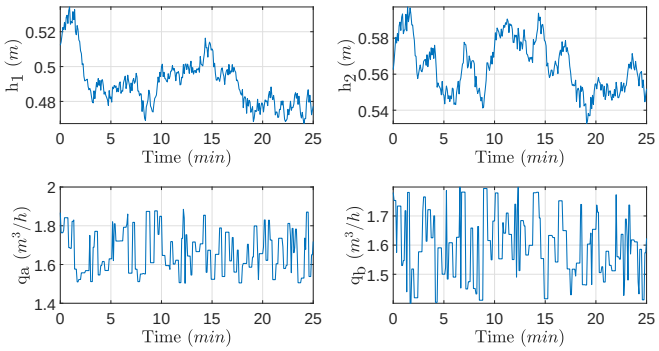
Remark 5.4. *Soft constraints in the outputs, $y(k) \in \mathcal{Y}_{soft}$, may also be considered, so that the performance will be penalized if the outputs go beyond $\mathcal{Y}_{soft}$. To account for the soft constraint, we will make use of barrier functions, adding a penalizing term to the stage cost. Hence, we can describe the stage cost $\ell(y, u)$ as the standard cost to track the reference, ℓ_t, plus the barrier function ℓ_b, as in Chapter 2:*

$$\ell(y, u) = \ell_t(y, u) + \ell_b(y), \quad (5.49)$$

where $\ell_b(y) = \Lambda\mathfrak{d}(y, \mathcal{Y}_{soft})$. Λ is a large constant, and $\mathfrak{d}(y, \mathcal{Y}_{soft})$ is a measurement of the distance of y to the set $\mathcal{Y}_{soft}$.



(a) Chirp signals



(b) Random signals

Figure 5.2: Experiments to obtain $\mathcal{D}(0)$.

5.4 Case study

The proposed controller is applied to the quadruple-tank process presented in Chapter 4 (on page 85). A different set of constraints is considered, given by $1.5 \leq q_a \leq 1.9$, $1.4 \leq q_b \leq 1.8\,\mathrm{m^3\,h^{-1}}$ in the inputs and $0.38 \leq h_1 \leq 0.62$, and $0.45 \leq h_2 \leq 0.73\,\mathrm{m}$ in the outputs.

A sequence of chirp signals covering the workspace is applied to generate the original data set, containing trajectories of inputs and outputs, with $N_0 = 5030$ data points. This data set $\mathcal{D}(0)$ is used for predicting. The simulation is represented in Figure 5.2a. The output sensors introduce noise that follows an uniform distribution with $\bar{\mathfrak{e}}$ equal to 1% of the measurement.

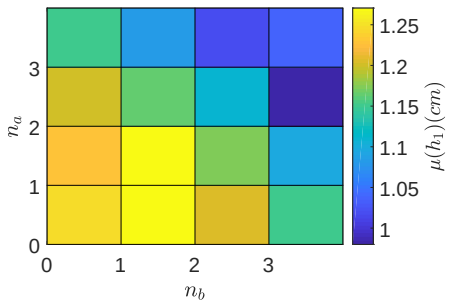


Figure 5.3: Maximum prediction error of h_1 (cm) for different values of n_a, n_b.

In addition, several tests with random input signals (square signals with random mean, amplitude and frequency) are carried out in order to obtain data sets for cross-validation, as represented partially in Figure 5.2b.

The signals are scaled between 0 and 1 w.r.t. the span of the constraints. The regressor $w(k)$ is constructed for different values of n_a and n_b. Cross validation tests are used to estimate the prediction error, which as shown in Figure 5.3, is minimized for $n_a = 2$ and $n_b = 3$. This model results in $\mu = [0.98,\ 1.24]$(cm), for which $L = [1.231,\ 0.836]$.

Then, the MPC presented in (5.7) is applied with $N = 4$, $\lambda = 10$ and $\tau_t = 0.01$. The resulting back-off for the safe model is $d_4 = [9.95,\ 6.43]$(cm), and $a_4 = [5.53,\ 3.86]$(cm). The stage and terminal cost are defined as follows:

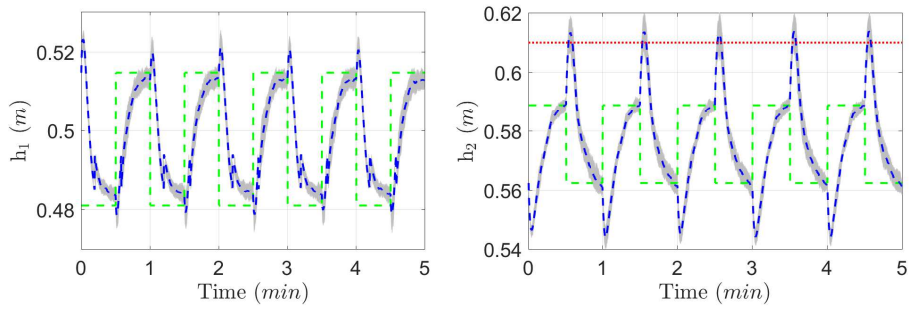
$$\ell(y,u) = \|y - y^{\text{ref}}\|_Q + \|u - u^{\text{ref}}\|_R + \Lambda\left(1 - \exp\left(\frac{-\max(y - y^s, 0)}{\delta}\right)\right) \quad (5.50)$$

$$V_f(x) = \|x - x^{\text{ref}}\|_P, \quad (5.51)$$

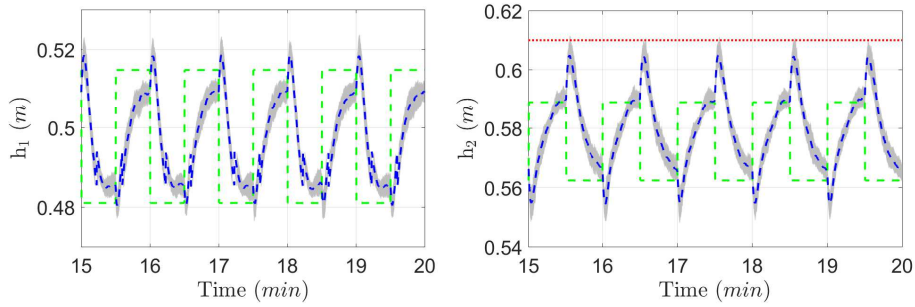
where the height of the second tank is penalized if it goes beyond $y^s = 0.61\,\text{m}$, with $\Lambda = 999$ and $\delta = 3 \times 10^{-3}$. Q is set to $100I_2$, $R = I_2$, and P is obtained by a LQR linearising the NARX model around the reference. The back-off of the tightened constraints results in a terminal constraint given by $\mathcal{Y}_N = \{y : [0.48,\ 0.514] \leq y \leq [0.52,\ 0.666]\text{(m)}\}$.

In the first set of simulations, two references alternate every 5 s between $y_1^{\text{ref}} = [0.481,\ 0.589]$ and $y_2^{\text{ref}} = [0.515,\ 0.562]$. The parameters γ and ϕ are obtained as in Rawlings & Mayne [133], resulting in $\gamma = 11698$ and $\phi = 5.8359 \times 10^7$, satisfying the assumptions of the controller.

The results are shown in Figure 5.4 for 100 simulations subject to the sensors' random noise. The red dotted line represents the soft constraint,



(a) First iterations



(b) Online performance

Figure 5.4: Online learning MPC applied to a quadruple tank process.

the green dashed line the reference, the grey band the set of signals and the blue line its mean.

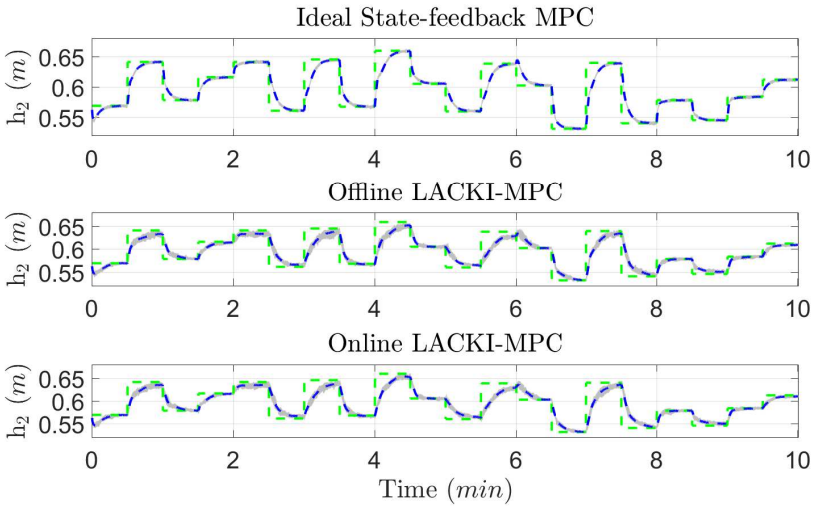
Figure 5.4 illustrates the triple contribution of this section. The set of tightened constraints, applied to the safe model, prevents the closed-loop system from violating the hard constraints (out of scope in the figure). The system is ISS and converges to the reference. Besides, the performance is enhanced by adding the online data to the prediction model used to minimize the cost. For instance, note that after 15 min, the model is able to learn the dynamics well enough to avoid the penalty of the soft constraint (red dotted line).

In the simulation presented, the reference alternated between two values. However, the proposed controller can be applied to any admissible reference. In a second set of simulations, represented in Figure 5.5, the reference changes randomly between different reachable values. In addition, in order to compare the proposed controller with other strategies, we apply the same

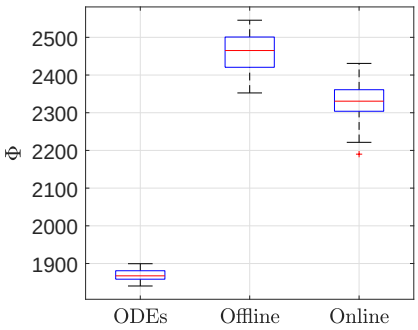
setting to (i) an MPC based on the ideal state-feedback model (identical to the plant), given by the set of ODEs (4.22), and (ii) the LACKI model without updating the data set, i.e., the offline version presented in Chapter 3. We compare the three controllers in Figure 5.5, representing the overall cost for 100 simulations, by means of the closed-loop performance index, which is defined as

$$\Phi = \sum_{i=1}^{t_{\text{sim}}} \ell(y(i), u(i)). \tag{5.52}$$

In the results it can be seen that the performance improves as more data are obtained for the online prediction model, and that the constraints are satisfied along the operation.



(a) Height of the second tank



(b) Boxplot comparison of Φ

Figure 5.5: Same setting applied to 100 simulations of three controllers, subject to random noise, in which the models are the set of ODEs, the offline version of the LACKI, and the online one

Chapter 6

Online learning robust MPC: an exploration-exploitation approach

6.1 Introduction

The controller proposed in Chapter 5 guaranteed robust stability for a given bound of the uncertainty. The worst-case error bound depends on the density of the data set, increasing as the system moves farther from it. This may lead to conservative results, in terms of admissible error bounds. The idea behind the controller proposed in this chapter is to break this dependence, bounding the region in which the system is allowed to be. This region is designed based on a confidence level that renders the controller feasible. The proposed approach is possible thanks to the design of an online learning predictive controller, which is based on an exploration-exploitation criteria.

The new controller follows an MPC for tracking [85] design, which is able to robustly steer the system to the best reachable reference, even in the case of low-dense data sets. Hence, the system is confined to a safe region, while the exploration policy allows such safe region to be time-dependant, increasing it as new measurements are included into the model.

When designing online learning methods, a trade-off between *exploration* and *exploitation* comes up. We use the term 'exploration' to measure how far from the known work-space the system is allowed to move. By 'exploitation' we address the fact that obtaining a large data set may not be the best strategy to follow, computationally speaking.

The results presented in this chapter have been published in [95].

6.2 Exploration-exploitation approach

6.2.1 Exploration

The objective is the design of predictive controllers able to control the system in regions with low data density. In such areas, the prediction error increases rapidly, probably exceeding the robustness bound that the controller can afford.

This problem can be tamed considering an exploration technique in which the control strategy forces the system to stay *close* to observed data, in a safe region where we have enough information to guarantee a worst case upper bound of the prediction error. This *safe region* is defined as

$$\mathcal{W}_r = \left\{ w : \min(\|w - w_i\|) \leq \tau_r, \, \forall i \in \mathbb{I}_1^{N_\mathcal{D}} \right\}, \tag{6.1}$$

for certain threshold $\tau_r \geq 0$.

Using KI with the Lipschitz case, the following property allows the method to relate the exploration distance τ_r with the estimation error bound.

Property 6.1. *The prediction error μ is bounded by*

$$\mu = L_f \tau_r + 2\bar{\epsilon}. \tag{6.2}$$

Note that this bound is based on the true Lipschitz constant L_f, which in general is unknown. In this chapter we assume that this constant is known, equal to the estimated constant L.

Assumption 6.1. $L = L_f$.

The veracity of Assumption 6.1 depends on the density of the data set, and it conditions the validity of the results presented in this chapter. Previous works on the estimation of the Lipschitz constant [35] provided a Pareto probability distribution on the Bayesian estimation of L_f. Besides, if the probability of an underestimation is bounded by ρ, i.e., $\mathbb{P}(L_f > L) \leq \rho$, the applicability of this chapter is extended to a confidence level $(1 - \rho)$, similar in spirit to Corollary 3.1 (see page 63).

6.2.2 Exploitation

It has been proven that the prediction error $d(k)$ vanishes when the density of the data set becomes infinite, up to $2\bar{\mathfrak{e}}$. In practice, adding data points to the model increases computation times, and hence, it may not be the ideal procedure to include every new data point observed. Instead, in this work we propose to implement an exploitation policy, adding only *informative* data points, that is, those that are not close to data points already seen.

We characterize the term *close* by another threshold of the distance, such that a new data point q is not informative if it belongs to the *well-known* region, defined as

$$\mathcal{W}_t = \left\{ w : \min(\|w - w_i\|) \leq \tau_t,\, \forall i \in \mathbb{I}_1^{N_\mathcal{D}} \right\}, \tag{6.3}$$

for a threshold $0 \leq \tau_t \leq \mathfrak{r}_r$.

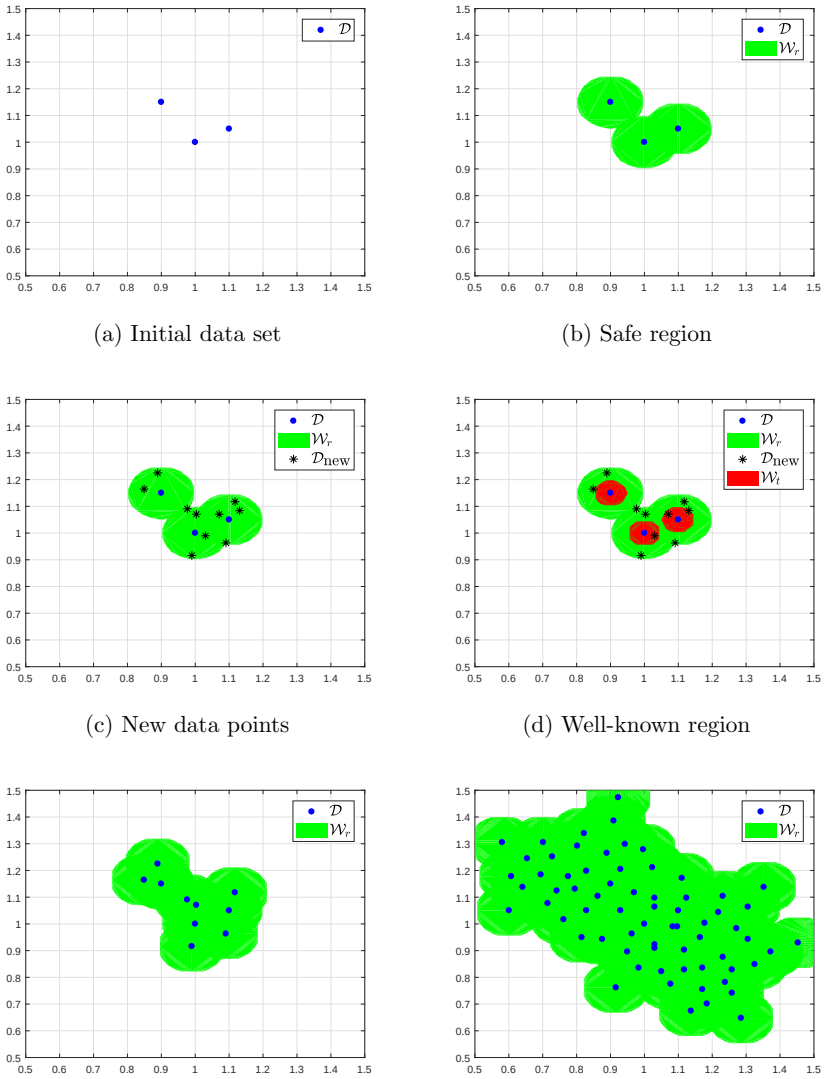
This threshold hyper-parameter has to be appropriately chosen, according to the information added by the inclusion of q in $\mathcal{D}$. The dependence of new data points w.r.t. the data stored in $\mathcal{D}$ was studied in [72], in the context of Gaussian processes for learning-based control. A procedure to prune uninformative sample points given a Lipschitz constant estimation is given in [34].

An example of the exploration-exploitation algorithm for a two-dimensional input space is shown in Figure 6.1. In this figure, an initial data set is considered with $N_\mathcal{D} = 3$. New data points are drown randomly within the safe region $\mathcal{W}_r$, but only added if they do not belong to $\mathcal{W}_t$, with $\mathfrak{r}_r = 0.1$ and $\tau_t = 0.05$. After 25 iterations $N_\mathcal{D} = 68$.

The data set is updated every time step, denoting $\mathcal{D}(k)$ the data set at time instant k, and $\mathcal{D}(0)$ the initial one. Note that the safe and the well-known regions are also time-dependent (i.e., $\mathcal{W}_r(k), \mathcal{W}_t(k)$). This update policy is such that

$$\mathcal{D}(k+1) = \begin{cases} \mathcal{D}(k) & \text{if } w(k) \in \mathcal{W}_t(k) \\ \mathcal{D}(k) \cup (y(k+1), w(k)) & \text{if } w(k) \notin \mathcal{W}_t(k). \end{cases} \tag{6.4}$$

The same occurs with the estimation of the Lipschitz constant. The recalculation of L is done recursively [36]. It can be proven that this estimation tends to the real L_f when the data set becomes infinitely dense. Computationally, this recursion is linear w.r.t. the cardinality of the data set, $\mathcal{O}(N_\mathcal{D})$, in contrast to other existing methods (such as Gaussian processes, which are quadratic, $\mathcal{O}(N_\mathcal{D}^2)$).



(a) Initial data set

(b) Safe region

(c) New data points

(d) Well-known region

(e) Updated data set

(f) $\mathcal{D}$ after 25 iterations

Figure 6.1: Exploration-exploitation algorithm.

Notice that from the computation point of view, the exploration-exploitation algorithms barely increase the calculation times. It is not necessary to calculate a closed-form of the sets $\mathcal{W}_r(k)$ and $\mathcal{W}_t(k)$. Instead, it is only necessary to check whether a given query point q belongs to them. This is carried out evaluating the minimum distance to the data set $\mathcal{W}_\mathcal{D}$; that is,

$$q \in \begin{cases} \mathcal{W}_t \\ \mathcal{W}_r \end{cases} \text{ iff } \min(\|q - w_i\|) \leq \begin{cases} \tau_t \\ \tau_r \end{cases}, \ \forall i \in \mathbb{I}_1^{N_\mathcal{D}}. \tag{6.5}$$

It is also important to remark that these distances $\|q - w_i\|$ are used to make predictions for that regressor, so they were already computed in the prediction step.

6.3 Online learning controller

In this section, a predictive controller that makes use of the data-based prediction model and the exploration-exploitation approach of Section 6.2 is presented. In order to be able to follow references (possibly outside of the initial data set), we propose to use a robust MPC for tracking [85] that explicitly takes into account at each time step the current safe region, in order to guarantee a given uncertainty bound on the predictions. MPC for tracking is designed to guarantee stability in presence of sudden reference changes, even if they are not reachable, which may be the case in the exploration scenario.

To this end, the MPC optimization problem considers an artificial reference (u_s, y_s) as additional decision variables. The deviation of the system to this reference is penalised along the prediction horizon, by means of a stage cost of the form $\ell(y - y_s, u - u_s)$. A term $V_O(y_s - y_t)$ is added to the cost function, in order to penalise the deviation of the artificial reference to the true reference, denoted (u_t, y_t).

Therefore, we propose the following controller, capable of exploring unseen regions, by forcing the system to stay in the explored area $\mathcal{W}_r$. Combined with the exploitation algorithm of adding data points only if $q \notin \mathcal{W}_t$, one can step by step move onto unexplored areas, while maintaining a prediction error bound suitable to ensure robust stability. The resulting optimization

problem P_N is:

$$\begin{aligned}
\min_{\mathbf{u},y_s,u_s} \quad & V_N(x(k),\mathbf{u},u_s,y_s;y_t) \\
& = \sum_{i=0}^{N-1} \ell(\hat{y}(i|k),u(k+i);y_s,u_s) + V_O(y_s-y_t) && \text{(6.6a)}\\
\text{s.t.} \quad & \hat{x}(0|k) = x(k) && \text{(6.6b)}\\
& \hat{x}(j+1|k) = \hat{F}(\hat{x}(j|k),u(k+j)), \quad j \in \mathbb{I}_0^{N-1} && \text{(6.6c)}\\
& \hat{y}(j|k) = M\hat{x}(j|k), \quad j \in \mathbb{I}_1^{N} && \text{(6.6d)}\\
& u(k+j) \in \mathcal{U}, \quad j \in \mathbb{I}_0^{N-1} && \text{(6.6e)}\\
& \hat{y}(j|k) \in \mathcal{Y}_j, \quad j \in \mathbb{I}_1^{N} && \text{(6.6f)}\\
& u_s \in \epsilon\mathcal{U} && \text{(6.6g)}\\
& y_s = \hat{\mathrm{f}}(x_s,u_s) && \text{(6.6h)}\\
& y_s \in \mathcal{Y}_N && \text{(6.6i)}\\
& \hat{y}(N|k) = y_s && \text{(6.6j)}\\
& \hat{w}(j|k) \in \mathcal{W}_r(k), \quad j \in \mathbb{I}_1^{N} && \text{(6.6k)}\\
& w_s \in \mathcal{W}_r(k), && \text{(6.6l)}
\end{aligned}$$

where $x_s = (y_s,\dots,y_s,u_s,\dots,u_s)$, $w_s = (x_s,u_s)$, and ϵ is a design parameter lower than (and close to) 1. For simplicity we may group the constraints $u \in \mathcal{U}$ and $y \in \mathcal{Y}$ under the notation $(y,u) \in \mathcal{Z}$.

Note that a terminal equality constraint is included. Besides, a set of tightened constraints $\mathcal{Y}_j$ is considered, as in Chapter 3. They are defined as

$$\mathcal{Y}_j = \mathcal{Y} \ominus \mathcal{B}(d_j(\mu)), \tag{6.7}$$

where d_j is a function of μ that can be calculated as per equation (3.7) (on page 58).

Notice that the updating policy guarantees recursive feasibility of the controller: since the regions $\mathcal{W}_r$ are dynamical, they evolve with time, and hence if $(y(k),u(k)) \in \mathcal{W}_r(k)$ then $(y(k+1),u(k+1)) \in \mathcal{W}_r(k+1)$.

The set of tightened constraints counteracts the effect of the error between the real plant and the data-based model (which is bounded by μ) on the constraints. Hence, this set must not be empty for any $j \in \mathbb{I}_1^N$. This is stated as an assumption in chapters 3, 4 and 5, conditioning the feasibility of the controllers.

In the proposed approach, the bound on the prediction error within the safe regions is a design parameter (see Property 6.1). This implies that in

general, τ_r is chosen so that $\mathcal{Y}_N$ is not empty, which is an important property from the implementation point of view. If the error bound for the whole state space were considered, it could be too large to obtain non-empty tightened constraints. Given the procedure to calculate such sets in Chapter 3, and the definition of μ in (6.2), the maximum admissible value of the exploration radius $\tau_r^{\max}$ can be explicitly obtained via reverse engineering, provided that Assumption 6.1 holds. The closed-loop properties of the proposed controller are presented next.

6.3.1 Stability analysis

The ingredients of the optimization problem are required to satisfy the following assumption:

Assumption 6.2.

1. *The stage cost function $\ell(y, u; y_s, u_s)$ is a positive definite function, and $\ell(y, u) \leq \alpha_y(\|y - y_s\|) + \alpha_u(\|u - u_s\|)$, for two $\mathcal{K}$-functions α_y, α_u.*
2. *The offset cost function $V_O(y_s - y_t)$ is a subdifferentiable convex positive definite function such that the best reachable reference*

$$y_s^0 = \arg\min_{y_s \in \mathcal{Y}_N} V_O(y_s - y_t)$$

is unique; and

$$V_O(y_s - y_t) - V_O(y_s^0 - y_t) \geq \alpha_O(|y_s - y_s^0|),$$

for a given $\mathcal{K}_\infty$-function α_O.

Define $\mathcal{Y}_e$ as a convex set of reachable equilibrium points,

$$\mathcal{Y}_e \subseteq \{y : \exists u_s \in \epsilon\mathcal{U} : \hat{\mathfrak{f}}(x_s, u_s) = y_s\}. \tag{6.8}$$

Then, the following assumptions are needed to derive the recursive feasibility of the optimization problem.

Assumption 6.3. $\mathcal{Y}_{N-1} \subseteq \mathcal{Y}_e$.

Assumption 6.4. *For all x such that $y \in \mathcal{Y}_{N-1}$ there exists a continuous control law $u_F = \kappa_F(y) \in \mathcal{U}$ such that $\hat{\mathfrak{f}}(x, u_F) \in \mathcal{Y}_N$.*

Theorem 6.1. *Suppose that Assumptions 6.1-6.4 hold for the optimization problem P_N. Let $\kappa_N(x)$ be the control law derived from the solution of P_N applied using a receding horizon policy. Then, for any $x(0) \in \mathcal{Z}$, the system controlled by the control law $u(k) = \kappa_N(x(k))$ is recursively feasible, stable, and the constraints are always satisfied, i.e. $u(k) \in \mathcal{U}$, $y(k) \in \mathcal{Y}$, $\forall k$.*

Proof.

This proof is sketched in two parts. First, recursive feasibility is proven.

Denote the optimal solution of the optimization problem P_N at time step k and state $x(k)$ as $\mathbf{u}^*(k)$. Let define the shifted sequence candidate for the optimization problem for $x(k+1)$ as $\bar{\mathbf{u}}$:

$$\bar{\mathbf{u}}(k+1) = \begin{cases} \bar{u}(j|k+1) & = & u^*(j-1|k), \quad j \in \mathbb{I}_0^{N-2} \\ \bar{u}(N-1|k+1) & = & u_F \\ \bar{u}_s(k+1) & = & \bar{u}_s, \end{cases} \tag{6.9}$$

where $\bar{u}_s$ is such that $y(N|k+1) = \hat{\mathfrak{f}}(x(N|k+1), \bar{u}_s)$.

From (6.6e) it is known that $\bar{u}(j|k+1) \in \mathcal{U}$ for $j \in \mathbb{I}_0^{N-2}$; and from (5.5) and Chapter 3 it is derived that

$$\bar{y}(j|k+1) \in y^*(j-1|k) \oplus \mathcal{B}(d_{j-1}) \in \mathcal{Y}_j.$$

Then, for $j = N-1$, given Assumption 6.4 it is known that $u_F \in \mathcal{U}$ and that

$$\bar{y}(N-1|k+1) \in y^*(N|k) \oplus \mathcal{B}(d_N) \in \mathcal{Y}_N \oplus \mathcal{B}(d_N) \subseteq \mathcal{Y}_{N-1}.$$

Moreover, taking Assumption 6.4, and equations (6.6h) and (6.6i) it is derived that for

$$\bar{y}_s = \hat{\mathfrak{f}}(\bar{x}(N-1|k+1), u_F) \in \mathcal{Y}_N \subseteq \mathcal{Y}_e$$

there exists a

$$\bar{u}_s \in \mathcal{dU} : \hat{\mathfrak{f}}(\bar{x}_s, \bar{u}_s) = y_s.$$

Hence given Assumption 6.3, and equation (6.6j), $y(N|k+1) \in \mathcal{Y}_N$, which proves that the problem is recursively feasible. In second place, stability is proven.

Let $\bar{\mathbf{u}}$ in (6.9) denote the candidate shifted sequence, and define and intermediate sequence (solution of the nominal problem) as

$$\tilde{\mathbf{u}}(k+1) = \begin{cases} \tilde{u}(j|k+1) & = & u^*(j-1|k) \quad j \in \mathbb{I}_0^{N-2} \\ \tilde{u}(N-1|k+1) & = & u_s^*(k) \\ \tilde{u}_s(k+1) & = & u_s^*(k). \end{cases} \tag{6.10}$$

Given the definition of V_N in (2.9a)[1] and its continuity ensured by Assumption 6.2, we know that the difference

$$\begin{aligned} V_N(x(k+1), \bar{\mathbf{u}}(k+1), &\bar{y}_s(k+1), \bar{u}_s(k+1)) \\ &- V_N(x(k+1|k), \bar{\mathbf{u}}(k+1), \bar{y}_s(k+1), \bar{u}_s(k+1)) \end{aligned}$$

is upper bounded by some $\mathcal{K}$-function

$$\theta_1(\|x(k+1) - x(k+1|k)\|) \leq \theta_1(\mu).$$

Analogously,

$$\begin{aligned} V_N(x(k+1|k), &\bar{\mathbf{u}}(k+1), \bar{y}_s(k+1), \bar{u}_s(k+1)) \\ &- V_N(x(k+1|k), \tilde{\mathbf{u}}(k+1), \tilde{y}_s(k+1), \tilde{u}_s(k+1)) \end{aligned}$$

is equal to

$$\begin{aligned} &\left[\sum_{j=0}^{N-2} \Big(\ell(y(j|k) - \bar{y}_s, \bar{u}(k+j) - \bar{u}_s) - \ell(y(j|k) - \tilde{y}_s, \tilde{u}(k+j) - \tilde{u}_s)\Big)\right] \\ &\quad + \Big[\ell(y(N|k) - \bar{y}_s, u_F - \bar{u}_s) - \ell(y(N|k) - \tilde{y}_s, u_s^* - \tilde{u}_s)\Big] \\ &\quad\quad + \Big[V_O(\bar{y}_s - y_t) - V_O(\tilde{y}_s - y_t)\Big]. \end{aligned} \tag{6.11}$$

Provided that $\bar{u}(k+j) = \tilde{u}(k+j)\, \forall j \in \mathbb{I}_0^{N-2}$, and given the definition of the stage cost in Assumption 6.2, then the first bracket is upper bounded by

$$(N-1)\big(\sigma_1(\|\bar{y}_s - \tilde{y}_s\|) + \sigma_2(\|\bar{u}_s - \tilde{u}_s\|)\big),$$

for some $\mathcal{K}$-functions σ_1, σ_2.

Let $g_s(y_s)$ denote the function that yields equilibrium inputs of the model, given its equilibrium output. Since the model is Lipschitz continuous and $\mathcal{Z}$ is compact, then g_s is uniformly continuous, and then

$$\sigma_2(\|\bar{u}_s - \tilde{u}_s\|) = \sigma_2(\|g_s(\bar{y}_s) - g_s(\tilde{y}_s)\|) \leq \sigma_3(\|\bar{y}_s - \tilde{y}_s\|).$$

Recall that for a given $\mathcal{K}$-function σ,

$$\begin{aligned} \sigma(\|\bar{y}_s - \tilde{y}_s\|) &= \sigma(\|y(N|k) - y(N|k+1)\|) \\ &= \sigma(\|\hat{\mathfrak{f}}(x(N|k), \tilde{u}_s) - \hat{\mathfrak{f}}(x(N-1|k+1), u_F)\|) \\ &= \sigma_a(\mu) + \sigma_b(\|\tilde{u}_s - u_F\|), \end{aligned}$$

[1]The dependence of V_N with y_t is omitted, for the sake of conciseness.

where given the continuity of the model, the derivation into σ_a is explained in Chapter 3, and taking $\kappa_F(y) = g_s(y_s)$ if $y \in \mathcal{Y}_N$,

$$\begin{aligned}\sigma_b(\|u_F - \tilde{u}_s\|) &= \sigma_b(\|\kappa_F(y(N-1|k+1)) - \kappa_F(\tilde{y}_s)\|) \\ &\leq \sigma_c(\|y(N-1|k+1) - \tilde{y}_s\|) \leq \sigma_d(\mu).\end{aligned}$$

The second bracket of (6.11) is upper bounded by

$$\sigma_4(\|\bar{y}_s - \tilde{y}_s\|) + \sigma_5(\|u_F - \bar{u}_s\|),$$

for some $\mathcal{K}$-functions σ_4, σ_5, provided that $u_s^* = \tilde{u}_s$.

Notice that, for the $\mathcal{K}$-functions $\sigma_{e,f,g,h}$,

$$\sigma_e(\|u_F - \bar{u}_s\|) \leq \sigma_f(\|u_F - \tilde{u}_s\|) + \sigma_g(\|\tilde{u}_s - \bar{u}_s\|) \leq \sigma_h(\mu).$$

The third bracket is also upper bounded by some $\mathcal{K}$-function $\sigma_6(\|\bar{y}_s - \tilde{y}_s\|)$, provided that Assumption 6.2 holds.

Apart from that, the nominal tracking of changing piece-wise constant reference signals [85] upper bounds

$$V_N(x(k+1|k), \bar{\mathbf{u}}(k+1), \bar{y}_s(k+1), \bar{u}_s(k+1)) - V_N^*(x(k))$$

by some $\mathcal{K}$-function $-\alpha(\|y(k) - \tilde{y}_s(k)\|)$.

Hence,

$$\begin{aligned}V_N(x(k+1), \bar{u}(k+1), \bar{x}_s(k+1), &\bar{u}_s(k+1)) - V_N^*(x(k)) \\ &\leq \theta(\mu) - \alpha(\|y(k) - y_s^*(k)\|).\end{aligned}$$

Thus, by optimality

$$\begin{aligned}V_N^*(x(k+1), u^*(k+1), &x_s^*(k+1), u_s^*(k+1)) \\ &\leq V_N(x(k+1), \bar{u}(k+1), \bar{x}_s(k+1), \bar{u}_s(k+1)),\end{aligned}$$

which yields a bound on the increment of V_N^*.

□

From this theorem, the following corollary can also be proven.

Corollary 6.1 (Convergence)**.** *In case that the prediction error of the KI model $\mu(k)$ tends to $\mathbb{0}$, the system converges to the best reachable reference y_s^0.*

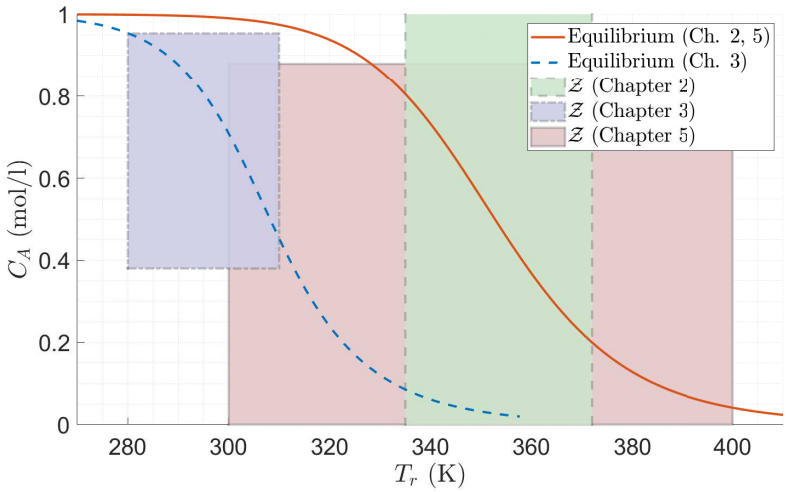


Figure 6.2: Set of equilibrium points of the CSTR, depending on the parameters; and set of constraints considered in each chapter.

Remark 6.1. *Note that the online algorithm presented here decreases the prediction error while operating the system. However, even in the absence of the exploitation policy (i.e. $\tau_t = 0$) and infinitely dense data sets, the maximum prediction error μ vanishes up to a factor of the noise level [36].*

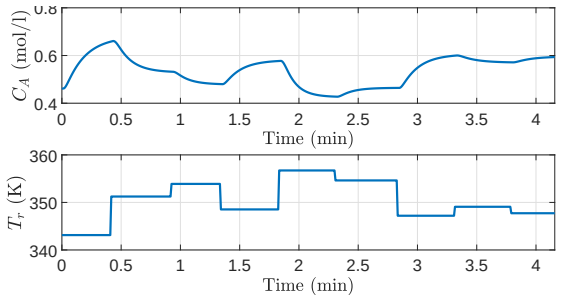
Illustrations of the operation of the proposed controller are presented in the following section.

6.4 Case study

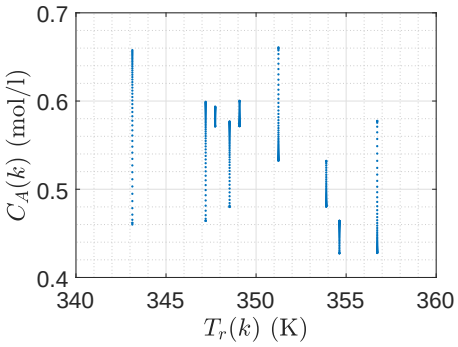
The continuously stirred tank reactor (CSTR) presented in Chapter 2 (on page 42) is considered. The constraints in the input are $300\,\mathrm{K} \leq T_r \leq 400\,\mathrm{K}$, and in the output $0 \leq C_A \leq 0.88\,\mathrm{mol\,l^{-1}}$. The different configurations of the CSTR used in this thesis are represented in Figure 6.2.

Online learning

We first consider a case in which, at the beginning of the simulation, very few data points are known, just 300 corresponding to some equilibrium points of the system, obtained from the experiment shown in Figure 6.3.



(a) Simulation obtained applying piece-wise constant control actions



(b) Data points in the input-output space

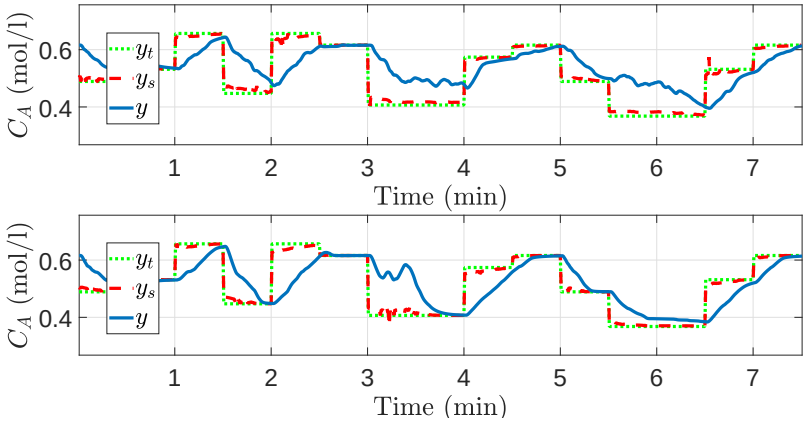
Figure 6.3: Initial data set, consisting of $N_{\mathcal{D}} = 300$ points.

The regressors are constructed for such data set, with $n_a = 2$ and $n_b = 0$. The estimation of $L(0)$ is 1.62.

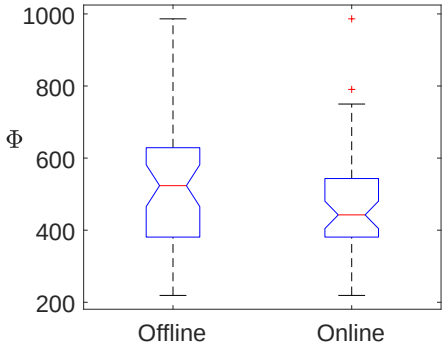
To motivate the online inclusion of data points while operating the system, we apply the proposed MPC (6.6), with 15 references varying randomly among $340\,\mathrm{K} \leq T_r \leq 360\,\mathrm{K}$, each of them maintained for 20 s. The prediction horizon is set $N = 3$, and the costs are quadratic, with $Q = 10$, $R = 1$ and $O = 100$. The exploration-exploitation is not considered in this example, i.e., $\tau_r = \infty$ and $\tau_t \gtrapprox 0$.

We compare two controllers subject to the same random noise, with and without the online updating policy, for 100 simulations. The results are represented in Figure 6.4. The performance is measured by

$$\Phi = \sum_{i=1}^{t_{\mathrm{sim}}} \ell(y(i), u(i), y_s(i), u_s(i)) + V_O(y_s(i) - y_t(i)). \tag{6.12}$$



(a) Output for the offline MPC (top) and online MPC (bottom)



(b) Performance indexes for 100 simulations

Figure 6.4: Comparison between the offline and online MPCs.

The results show that the proposed controller is able to follow the reference better than a controller that does not update the data set, incurring into a smaller cost.

Exploring

Consider that this same CSTR has historically been operated within the region comprised by $335 \leq T_r \leq 370\,\mathrm{K}$, as in Chapter 2. Therefore, a large data set within this region is available, as shown in Figure 6.5. Imagine that the owners consider operating the tank in other temperatures, where nothing is known of how the system behaves.

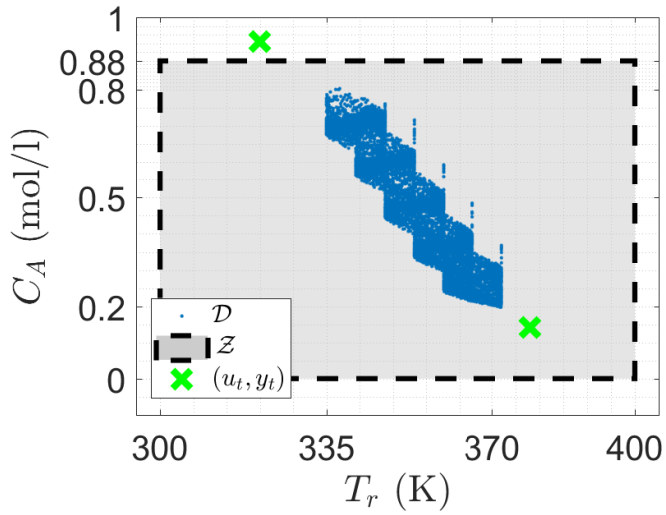


Figure 6.5: Input-output space showing the constraints $\mathcal{Z}$, the initial data set $\mathcal{D}(0)$, and the references (u_t, y_t).

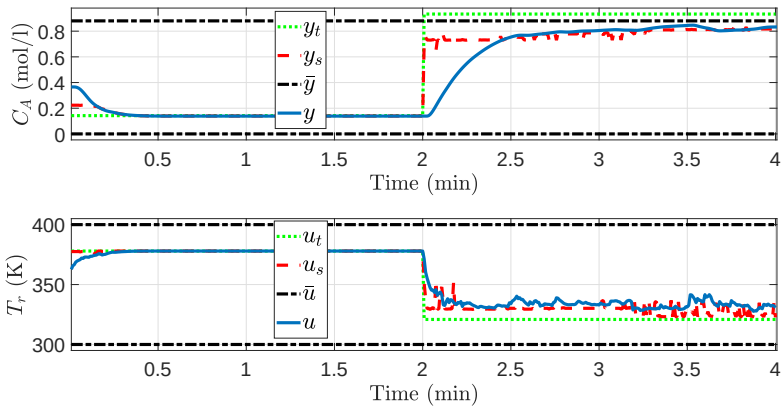
The initial data set yields $L(0) = 1.62$. Assuming this as the true Lipschitz constant, and setting $N = 2$, the maximum exploration radius such that $\mathcal{Y}_N$ is not empty is $\tau_r^{\max} = 0.10$ [2], provided that $\bar{\epsilon} = 0.02\,\mathrm{mol\,l^{-1}}$. The proposed controller (6.6) is applied, with two piece-wise constant references: $y_t = 0.14\,\mathrm{mol\,l^{-1}}$ and $0.93\,\mathrm{mol\,l^{-1}}$, each of them lasting 2 min. Note that both references are in the unexplored area (Fig. 6.5). Besides, notice the second one is not even admissible. The radius for exploitation is set to $\tau_t = 0.002$, and for exploration $\tau_r = 0.6\tau_r^{\max}$, to mitigate the possible effect of the underestimation of L_f.

The result of the simulation is shown in Figure 6.6. Note how the data set is increased with the points visited throughout the operation. Observe also the trajectory of the optimal artificial reference and its convergence to the best reachable steady state. The closed-loop system reaches the real reference, even if it was not reachable in the beginning. In the second part, the robust MPC prevents the closed-loop system from violating the constraints, by means of the set of tightened constraints, while steering the system to the closest reachable state.

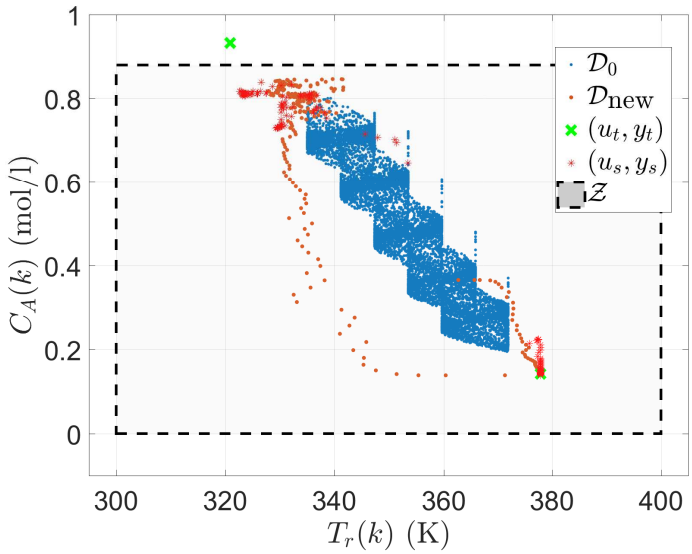
Without the exploration-exploitation algorithm presented in this chapter,

[2] Recall that every signal is scaled to range $0 - 1$.

the closed-loop system would fail to converge to the given reference. On the other hand, if no restriction is added on how far from known data points the system can go, the prediction error increases immensely, being unable to properly forecast the evolution of the plant, and therefore to fulfil the constraints.



(a) Closed-loop system



(b) Data points in the input-output space

Figure 6.6: Online learning MPC.

Chapter 7

Conclusions and future work

This chapter briefly sums up the contributions presented and states some future lines that arise from the resulting research.

7.1 Summary of contributions

This thesis has contributed to two different fields, machine learning and model predictive control, as well as developing a common framework to merge both.

Following the objective of controlling unknown nonlinear systems based on inputs and outputs observations, in an efficient and safe fashion, model predictive control was chosen as control strategy. Seeking a deterministic approach to derive the stability analysis of the resulting closed-loop, prediction techniques that guaranteed a bound on the uncertainty, i.e., on the worst-case prediction error had to be chosen. To this end, the kinky inference class of learning methods resulted particularly handy, since it is able to effectively model nonlinear output-feedback systems. Besides, the resulting prediction model is Hölder continuous, which is a relevant property in the derivation of the stability analysis of the proposed controllers.

As a non-parametric method, the computational burden of the inference algorithm grows with the cardinality of the data set, and its simplicity (which is proven to be an advantage in most cases) may lead to an over conservative method, yielding too large uncertainty bounds. Both facts may forbid the application of the method to model predictive control schemes, given the real-time feature of these, and the necessity for robust constraint satisfaction.

To overcome such problems, several methods to enhance the standard kinky inference learning class have been proposed. Specifically, the projected KI presented in Chapter 2 aims to reduce the computational times of the inference, which allowed for the derivation of output-feedback model predictive controllers. Second, the componentwise Hölder predictor presented in Chapter 4 is proven to improve the standard KI method, yielding smaller worst-case prediction errors. Besides, it was proven that this method enlarges the feasibility region of controllers based on it.

With respect to the controllers proposed, the objective has been to develop stabilizing and robust designs of MPC without terminal constraint, which is itself an important contribution to the field of predictive control. This was motivated by the avoidance of the calculation of invariant robust sets, which is undoubtedly the most difficult ingredient in the design of MPCs, even more in the case of prediction models based on data. In this context, new controllers have been proposed.

First, an output-feedback MPC based on the smoothed projected version of the KI algorithm was derived in Chapter 2. It was proven able to robustly satisfy soft constraints in the outputs and hard constraints in the inputs, while converging to the reference.

Next, an MPC able to satisfy hard constraints in the output, which is a much more complex case, was proposed in Chapter 3. Instead of adding a terminal constraint based on invariant sets, the terminal cost was weighted by a factor λ. Based on this approach, another controller with larger domains of attraction was derived using the componentwise Hölder predictor of Chapter 4.

Chapter 5 opened the door to the online approach, in which the controllers proposed sought to improve their performance by the inclusion of new measurements that became available during the operation of the plant. An online learning MPC was proposed, based on a double prediction model: a safe model was derived offline with the initial data set, in charge of ensuring robust stability and constraints satisfaction. Then, an updated model was used to improve the predictions, yielding smaller costs in the optimization problem, and hence more efficient performance of the closed-loop.

Finally, an online learning MPC for tracking, based on an exploration-exploitation criteria, was presented in Chapter 6. The design proposed in this chapter is able to uncouple the uncertainty and the domain of attraction of the controller, by means of combining the exploration approach with the tracking formulation. Hence, the controller was proven able to maintain robust stability of the closed-loop even in the case of low-dense data sets.

Besides, the exploitation criteria prevented the model to become computationally intractable. These facts allowed the system to be steered to the best reachable reference, even if it was in unexplored areas of the workspace, while robustly satisfying the constraints.

7.2 Future work

The previous paragraphs have briefly summarized the contributions made to the field of data and learning-based predictive control, which has gained increasing attention during the years of elaboration of this thesis. As a consequence of the research presented, further works will certainly arise motivated by those already mentioned.

Among other lines, we will be interested in the validation of the proposed methodologies in experimental plants, which will include the design of the experiments and processing of the data sets, as well as the case-dependant design of the models and of the control law, in order to illustrate the proven behaviour of the proposed strategies.

In addition, note that several methods were proposed to enhance the predictions. Combining these methods is a short or medium term future objective, since it may result in a broader and more general inference algorithm suitable for many machine learning applications, as well as to model predictive control. Hence, novel and more efficient implementations of the KI class will be proposed in the future.

In line with the previous point, the parallelization of the kinky inference algorithm is a project that we have already begun [119], and that will undoubtedly allow for efficient and fast implementations of the prediction models. This will consequently increase the list of case studies where the methodologies presented in this thesis may find application, like embedded and real-time systems, which could be combined with the controllers proposed.

On other matters, the stochastic study of the controllers derived must be addressed. For example, the scenario approach, order statistics or probabilistic optimization [4] may open useful frameworks for the data-based approaches considered in this thesis.

The combination of other inference methods with the ones proposed in these chapters may lead to enhancing the overall performance, so deriving the stability analysis for models based on, e.g., Gaussian processes [94] will

be an interesting research topic.

In addition, the application of these methods to changing operation conditions or to systems varying in time (either their structure, parameters, etc.) can be studied, using the online configurability of the techniques proposed in this thesis. To this end, dynamical data sets, able to forget past measurements in detriment to new ones can be considered.

Finally, another current line of research [105] is the extension of the proposed control laws to take into account economic criteria in the optimization step, so further investigation on how the learning-based approach could be adapted to such controllers is a promising field.

Appendix A

Working with data

In general, when identifying a model based on input-output data, these data must be *informative* and *rich* enough to correctly capture the dynamics of the system within the considered workspace. To this end, this appendix presents some aspects that must be taken into account when working with data-based algorithms.

A.1 Input design

The quality of the available data is a key aspect in system identification, even more in the nonlinear case (as this thesis considers), since the nonlinear behaviour forbids the model to extrapolate well enough when estimating over unseen areas of the workspace of the plant.

The input-output data must capture the dynamics and modes of the system correctly. To this aim, *input design* is a crucial step when obtaining valid data sets for input-output identification.

Consequently, this thesis in general followed the procedures explained by Åström & Hägglund [12], Billings & Fadzil [23] and Rivera & Jun [134] for the design of input signals. Therefore, to obtain data trajectories in order to learn the case studies considered in this thesis, we avoided the use of grids in the inputs and/or in the initial states. Although grids in the considered spaces may result a quick and easily implementable approach in simulated case studies, they are unlikely to represent applicable methods in real plants, and they would probably fail in capturing most of dynamics in the nonlinear system. Instead, there are two other types of input trajectories that are considered:

Pseudorandom binary sequences (PRBS)

PRBS are widely used in the identification and telecommunications fields. The signals have a square shape, alternating between two values at apparently random times. It is said to be *pseudorandom* since it is generated via deterministic computer algorithms.

Following this spirit, this thesis uses pseudorandom sequences (not binary), such that both amplitude and length of the steps of the designed input signal are (pseudo-)random.

Chirp signals

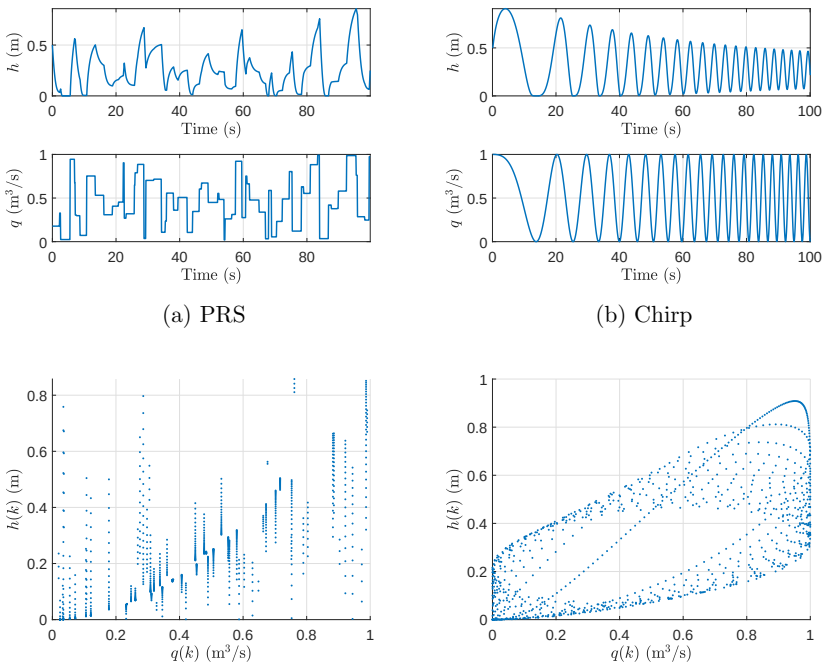
Chirp signals are sinusoidal sequences whose frequency f varies along time. Apart from identification, they are commonly used in sonar and radar applications. This work only focuses in chirp signals whose frequency increase linearly with time, such that

$$u(t) = \sin\left(2\pi\left(\frac{\mathrm{f}_f - \mathrm{f}_0}{T}t + \mathrm{f}_0\right)t\right), \tag{A.1}$$

where $\mathrm{f}_0 \leq \mathrm{f}_f$ are the initial and final frequencies, respectively, and T is the length of the signal. The initial frequency can be taken approximately equal to the crossover frequency of the system, obtained closing the loop with a relay at a given set-point. The final frequency can be taken two or three orders of magnitude higher than the initial frequency.

An example of both signals is shown in Figure A.1, as well as their application to the SISO tank presented in Section 1.3.2. The PRS's amplitude is chosen from a random uniform distribution from 0 to 1, and the same for the length of each step, between 0.2 and 30 s. The initial frequency of the chirp signal is calculated with a relay test over $(y_s, u_s) = (0.25\,\mathrm{m}, 0.5\,\mathrm{m^3\,s^{-1}})$, yielding $\mathrm{f}_0 = 4.3\,\mathrm{mHz}$. The final frequency is set to $\mathrm{f}_f = 0.43\,\mathrm{Hz}$. Both signals last $T = 100\,\mathrm{s}$.

Note that, in general, pseudorandom sequences (Fig. A.1c) generate data points which tend to the set of equilibrium points of the system (cf. Figure 1.2b), while the chirp signals are more likely to generate *rounded* trajectories in the input-output space (see Figure A.1d).



(a) PRS

(b) Chirp

(c) Data set generated with PRS

(d) Data set generated with chirps

Figure A.1: PRS and chirp signals applied to a water tank, and the data set that they generate.

A.2 Dimensionality reduction

High dimensional input spaces may seriously harm the applicability of data-based algorithms when learning the ground truth function. Dimensionality reduction is a sub-field of machine learning that aims to reduce the number of independent variables in consideration, that is, to decrease the number of input dimensions n_w, while worsening the performance at least as possible.

In this thesis, we consider a method that applies a linear transformation to the input space, named *principal component analysis*; but there exists many others, like discriminant analysis [14] or non-negative matrix factorization [75].

A.2.1 Principal component analysis

Principal component analysis (PCA) was invented by K. Pearson in 1901 [123]. In linear algebra it is called *single value decomposition* [45]. The principal component of an n-dimensional data set is the $(n-1)$-dimensional line that minimizes the average squared distance from the points to the line (similar in spirit to the least squares). Then, the second principal component is the line, orthogonal to the first, that minimizes the same value; and continuing until the n-th line. The resulting set conforms an orthogonal basis called the principal components [69].

Given a data set $\mathcal{W}_\mathcal{D} = \{w_i \in \mathbb{R}^{n_w}, i \in \mathbb{I}_1^{N_\mathcal{D}}\} \in \mathcal{W}$, the coefficients that transform $\mathcal{W}$ into the space of principal components of $\mathcal{W}_\mathcal{D}$ are gathered in the matrix T. The transformed space is denoted $\mathcal{V} \subseteq \mathbb{R}^{n_w}$. Analogously, the transformed input data set is denoted $\mathcal{V}_\mathcal{D} \in \mathbb{R}^{n_w \times N_\mathcal{D}}$.

They are obtained applying the linear transformation

$$\mathcal{V}_{(\mathcal{D})} = T\mathcal{W}_{(\mathcal{D})} = \{v = Tw,\, w \in \mathcal{W}_{(\mathcal{D})}\}. \tag{A.2}$$

Example A.1. *Figure A.2 shows an input data set $\mathcal{W}_\mathcal{D}$ of $N_\mathcal{D} = 30000$ data points with $n_w = 4$, from the continuously stirred tank reactor (CSTR) plant described in Chapter 2, obtained applying a sequence of chirp signals. It is a SISO system with $n_a = 2$ and $n_b = 0$, so*

$$w(k) = (y(k), y(k-1), y(k-2), u(k)).$$

Both inputs and outputs are scaled to range from 0 to 1. Figure A.3 shows the transformed input data set $\mathcal{V}_\mathcal{D}$.

Notice that, so far, the principal component analysis does not reduce the dimensionality of the input space. It only applies a linear transformation. In order to decrease dimensionality, an appropriate number $n_h \leq n_w$ of components (or dimensions) is selected. To this end, the standard deviation of $\mathcal{V}_\mathcal{D}$ is obtained as

$$S_i(\mathcal{V}_\mathcal{D}) = \sqrt{\frac{1}{N_D - 1}\sum_{j=1}^{N_\mathcal{D}} |v_{i,j} - m_i|^2}, \tag{A.3}$$

where S_i is the standard deviation of the i-th component of $\mathcal{V}_\mathcal{D}$, $v_{i,j}$ its i,j-th element and m_i its mean, for $i = 1, \ldots, n_w$:

$$m_i(\mathcal{V}_\mathcal{D}) = \frac{1}{N_\mathcal{D}}\sum_{j=1}^{N_\mathcal{D}} v_{i,j}. \tag{A.4}$$

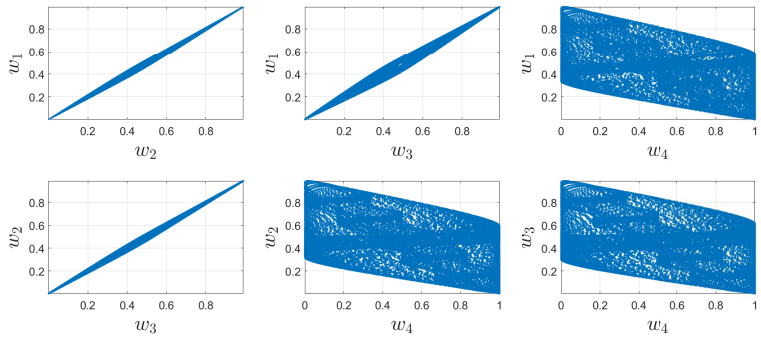


Figure A.2: Input data set $\mathcal{W}_\mathcal{D}$ obtained applying chirp signals to the CSTR. Note the high correlation among $y(k), y(k-1)$ and $y(k-2)$.

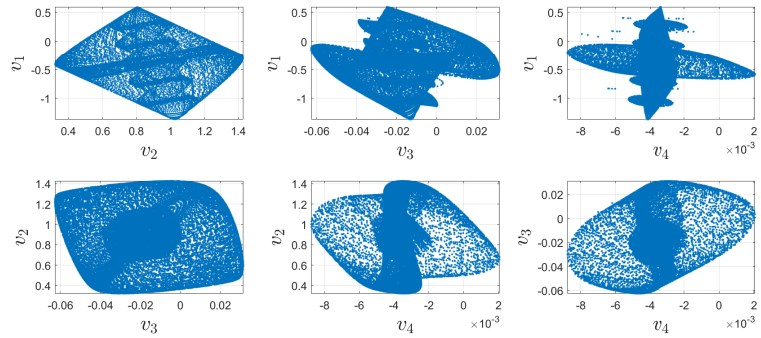


Figure A.3: Transformed data set $\mathcal{V}_\mathcal{D}$ applying PCA to the data set represented in Figure A.2. Note the smaller ranges in v_i as i increases.

Then, the n_h components whose standard deviation is lower than certain threshold are discarded, yielding a lower-dimensional space. The threshold may be selected given by some percentage σ of the maximum standard deviation (the one of the first component of $\mathcal{V}_\mathcal{D}$), that is,

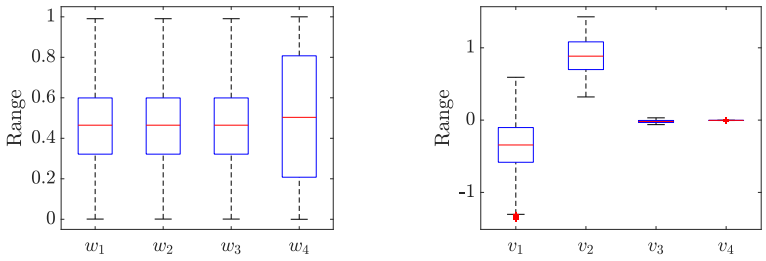
$$n_h = \max\{i : S_i(\mathcal{V}_\mathcal{D}) \geq \sigma S_1(\mathcal{V}_\mathcal{D})\}, \tag{A.5}$$

for some $0 \leq \sigma \leq 1$ and for all $i \in \mathbb{I}_1^{n_w}$.

Table A.1 represents the standard deviations of the example shown in Figure A.3. The ranges in $\mathcal{V}_\mathcal{D}$ are represented in Figure A.4. In this case, it is sensible to choose $\sigma = 10\%$, and thus, a lower dimensional input data set is obtained with the first two components of the transformed variables.

i	1	2	3	4
S_i	0.3966	0.2829	0.0219	0.0012
$\frac{100S_i}{S_1}$	100	71.339	5.5298	0.3132

Table A.1: Standard deviation of the case example shown in Figure A.3.



(a) $\mathcal{W}_\mathcal{D}$ (b) $\mathcal{V}_\mathcal{D}$

Figure A.4: Data set ranges by input dimension, before and after the PCA transformation.

Note that this process of dimensionality reduction is fully accessible and reversible by the algorithm, provided that the transformation matrix T is invertible. Hence, the reduction of input dimensions may be applied only in some parts of the control process. For example, in the learning method proposed in Chapter 2, queries are labelled as *close* to the data set according to the lower-dimensional space, but then the prediction is computed using the whole original input space.

A.3 Data processing

In general, the process of obtaining data and learning with it involves a large amount of suboperations, constituting a whole field of machine learning, from sampling [46] to data mining [57], which is sometimes referred to as the *knowledge discovery in databases* [49]. This thesis is not focused in this field, and only two further operations are performed with the raw observed data set, apart from constructing the regressor presented in (1.11), given past input-output trajectories and applying PCA. These operations are **scaling** and **data pruning**.

When working with databases, it is desirable (if not compulsory) to work with scaled data, such that every value ranges along the same order of magnitude, typically between 0 and 1. This is how the algorithms of this thesis are going to work. Recall that the regressor (1.11) groups output and input terms. The ranges and units of the inputs usually differ from the outputs, or even among them, so the process of computing metrics among data points does not make sense without a scaling. This is done such that for a set of N measurements of the vector $v \in \mathbb{R}^n$,

$$v_{s,i} = \frac{v_i - \min_i v_i}{\max_i v_i - \min_i v_i}, \forall i \in \mathbb{I}_1^N. \tag{A.6}$$

Note also that after a principal component analysis, another scaling is required (see Figure A.4).

A.3.1 Data pruning

The size of the data set of observations, i.e., $N_{\mathcal{D}} \times n_w$, plays a crucial role in the applicability of any machine learning method, specifically in those considered in this thesis. Section A.2 addressed the problem of dimensionality reduction, giving methods to decrease n_w. This section considers reducing $N_{\mathcal{D}}$, that is, the number of points in the data set.

To this end, if the data set is too large, it may be decimated, pruning points that are of less interest to the algorithm. A trade-off between prediction accuracy and computational complexity comes up, and it is the engineer's duty to set the threshold on how large the cardinality of the data set is allowed to be.

There exists many techniques for data pruning. T. Alpcan [8] and H. Kingravi [72] studied the dependence of a data point w.r.t. the data stored in $\mathcal{D}$, in the context of Gaussian processes for learning-based control. A procedure to prune uninformative sample points given a Lipschitz constant estimation is given in [34, Appendix C3].

A simple procedure that is applied in this thesis consists in decimating points that are close to each other. The term *close* is defined with a threshold $\tau_p > 0$ of the distance between data points, such that the j-th sample is considered close to the data set (and therefore uninformative, so it can be removed) if

$$\min(\|w_j - w_i\|) \leq \tau_p, \forall w_i \neq w_j \in \mathcal{W}_{\mathcal{D}}. \tag{A.7}$$

Appendix B

Further introduction to kinky inference

B.1 Computing the Hölder parameters

As it was stated in Chapter 1, the true Hölder constant L_f is unknown. Two methods are analysed here, used to obtain the Hölder parameter L, to make predictions using (1.16).

B.1.1 Lazily adapted constant kinky inference

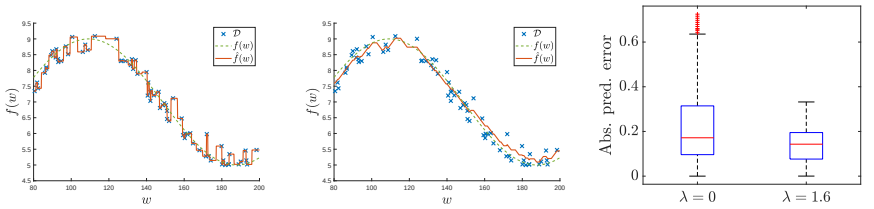
This method (LACKI) was proposed in [36], and used, e.g., in Chapter 3. It yields the minimum L that is consistent with the observed data set. Many interesting properties of the KI predictor can de derived if this method is applied, as it will be analysed in the following section.

The constant is computed as

$$L = \max\left(\frac{\|\tilde{y}_i - \tilde{y}_j\| - \eta}{\|w_i - w_j\|}, \forall i \neq j \in \mathbb{I}_1^{N_\mathcal{D}}\right), \tag{B.1}$$

where η is a regularization parameter that has proven useful to smooth out the noise. [1] Some properties of the resulting LACKI predictor that will be commented in the following section require to set $\eta \geq 2\bar{\mathfrak{e}}$.

[1] The regularization parameter is usually denoted λ in the KI literature. However, we will denote it η, to prevent confusion with the λ used to weight the terminal cost.



(a) $\eta = 0$ (b) $\eta = 1.6$ (c) Abs. prediction errors

Figure B.1: Effect of the regularization term in LACKI.

Example B.1. *Consider the ground truth* $f(w) = 2\sin((w-70)/25) + 7$. *Drawn* $N_{\mathcal{D}} = 70$ *data points from an uniform distribution with Gaussian noise of* 0 *mean and standard deviation of* 0.25, *Figure B.1 shows the performance of the LACKI predictor with* $p = 1$, *both for* $\eta = 0$ *and* $\eta = 1.6$. *The resulting constants are* $L = 53.39$ *for the former and* $L = 0.08$ *for the latter.*

B.1.2 Parameter optimized kinky inference

This method (POKI) was proposed in [37], and consists in obtaining θ as those parameters that minimize the prediction error. To this aim, two data sets are needed: a training data set $\mathcal{D}$ to compute predictions, and a validation data set $\mathcal{D}_{\text{test}}$, used to evaluate the prediction error.

Example B.2. *Consider the case of a tank with two holes, one at the bottom and the other one at* $H = 1\,\text{m}$. *If the water height* h *were to be controlled in a compact set such that* $0.8 \leq h \leq 1.2\,\text{m}$, *the model of the problem is:*

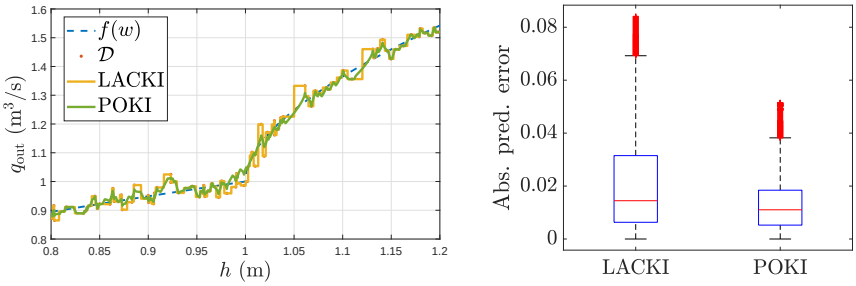
$$A\frac{dh(t)}{dt} = q_{\text{in}}(t) - q_{\text{out}}(h(t)), \tag{B.2}$$

where A *is the area of the tank,* q_{in} *the input flow and the output flow is*

$$q_{\text{out}}(t) = k_1\sqrt{h(t)} + k_2\sqrt{\max(h(t) - H, 0)}.$$

Figure B.2 shows the equilibrium points of the output flow q_{out}, *to be identified for* $k_1 = k_2 = 1$, *with respect to the height. The true Lipschitz constant* $L_f(p = 1)$ *becomes infinite for* h=1 m, *while the Hölder continuity still holds for* $p \leq 0.5$.

Given $N_{\mathcal{D}} = 100$ *random samples, with Gaussian noise of 0 mean and standard deviation of* 0.025, *Figure B.2 also shows the prediction for the*



(a) Set of equilibrium

(b) Abs. prediction errors

Figure B.2: Performance of the LACKI and the POKI predictors.

LACKI (without regularization and $p = 1$) and the POKI predictors, obtaining $L = 1105.8$ for the former and $L = 1.57$, $p = 0.6$ for the latter.

The properties, advantages and drawbacks of the kinky inference class are analysed in the following section.

B.2 Properties

In comparison to other machine learning methods, kinky inference enjoys several properties that make it suitable for the derivation of robust predictive controllers. These properties are the reason why this method has been chosen as the cornerstone technique under research in this thesis.

B.2.1 Complexity

Before analysing the computational complexity, it is important to remark the learning simplicity of KI. The *tuning* process of KI only requires the computation of two hyper-parameters, L and p. Obtaining these constants given a noisy data set is also easily done, as shown in the previous section. This learning step is often computed offline, prior to operating the system or making predictions, so computational complexity is typically not crucial at this step. The LACKI problem is $\mathcal{O}(N_{\mathcal{D}}^2)$.

The prediction step is also simple, as opposed to other non-parametric techniques. Note, for example, that Gaussian process regression requires the inversion of the covariance matrices. Instead, KI just computes the ceiling

and floor functions, and the mean between them. The method is therefore numerically stable, and it can be embedded in micro-controllers or platforms with fewer computing capacity. On the other hand, its simplicity often requires large data sets to perform well enough, and the method is not typically suitable for extrapolation.

Given the non-parametric nature of the method, it explicitly uses the training data set $\mathcal{D}$ for prediction. This is one of the main drawbacks of non-parametric methods, although the computational effort of KI is linear with respect to the cardinality of the data set $\mathcal{O}(N_{\mathcal{D}} n_w)$, in contrast to the cubic dependence of Gaussian processes $\mathcal{O}(N_{\mathcal{D}}^3)$ (assuming that $N_{\mathcal{D}} > n_w$).

B.2.2 Consistency

One of the most important properties of the method is that the predictor presented in (1.16) is Hölder continuous, with Hölder parameters L and p. This is proven in [34] and it has been an important property in this work, because it implies that additional functions derived from $\hat{\mathfrak{f}}$ are also Hölder continuous. In particular, the cost to be minimized by the model predictive controllers proposed inherited Hölder continuity with known parameters, which is a requirement for most of the robust MPC designs.

Besides, if the LACKI approach is applied, several properties that will be analysed in what follows hold. First, the predictor is sample consistent up to $\eta/2 + \bar{\mathfrak{e}}$, as proven in [36, Lemma 2.7].

B.2.3 Bounded uncertainty

The bound on the uncertainty (μ) is a key value to design safe and robust predictive controllers. First, it is remarked that μ exists since (i) the ground truth is bounded and (ii) the KI predictor is bounded, irrespective of the boundedness of input space $\mathcal{W}$, as proven in [36, Lemma 2.5].

Next, note that Hölder continuity of the ground truth function implies that the enclosure defined by the ceiling and floor functions, obtained with the real Hölder parameters (L_f, p), defines the real bound on the uncertainty, up to twice the noise level. Hence, for all $q \in \mathcal{W}$,

$$\|\hat{\mathfrak{f}}(q; L_f, p, \mathcal{D}) - f(q)\| \leq 2L_f \mathcal{R}_{\mathcal{D}}^p + 2\bar{\mathfrak{e}}, \tag{B.3}$$

where $\mathcal{R}_{\mathcal{D}}$ denotes distance between the furthest possible query and the data

set:

$$\mathcal{R}_{\mathcal{D}} = \max_{q \in \mathcal{W}} \min_{w \in \mathcal{W}_{\mathcal{D}}} (\|q - w\|). \tag{B.4}$$

Note that if the kinky inference predictor is used, computing the prediction as the mean value between the ceiling and floor functions with the real L_f, the bound in (B.3) reduces to

$$\|\hat{\mathfrak{f}}(q; L_f, p, \mathcal{D}) - f(q)\| \leq L_f \mathcal{R}_{\mathcal{D}}^p + 2\bar{\mathfrak{e}}. \tag{B.5}$$

If the real Hölder constant L_f is unknown, the predictor $\hat{\mathfrak{f}}$ can be computed with an L estimated from the data set. If the LACKI method is applied with $\eta \geq 2\bar{\mathfrak{e}}$, note that $L \leq L_f$, and then the predictor error is bounded, as proven in [36], by

$$\|\hat{\mathfrak{f}}(q; \theta, \mathcal{D}) - f(q)\| \leq (L_f + L)\mathcal{R}_{\mathcal{D}}^p + \frac{\eta}{2} + \bar{\mathfrak{e}}. \tag{B.6}$$

The previous equations provide a bound on μ depending only on the real L_f (B.5) or also on the estimated one L (B.6). In any case, the real Hölder constant L_f is unknown. Hence, when knowledge of the bound on the uncertainty is required, one must make certain assumptions, e.g. either

- Assume the real Hölder constant L_f is known, as done in NSM approaches [112].
- Derive stochastic estimations of the real Hölder constant L_f, given the LACKI estimation L, as proposed in [35]. Then, one could derive probabilistic estimations of the bound of the uncertainty, μ.
- Assume that the real μ is computed via validation. This is customary in identification frameworks. The validation process must be broad enough to find as much prediction errors as possible, in order to truly describe the uncertainty of the model.

B.2.4 Learning

Notice that the previous property, (B.6), proves that the LACKI estimator is a learning method, according to T. Mitchell's definition [113]. Indeed, it is proven that as the data set becomes infinitely dense, making $\mathcal{R}_{\mathcal{D}} \to 0$, the prediction error vanishes, up to twice the noise level.

The learning feature of an algorithm is an important fact, since it allows the online configuration of the method, as it was discussed in Chapter 5. Note that for the rigorous proof it is necessary to demonstrate two facts:

1. That the predictor is sample consistent. This requires that if new data points are not consistent with the estimated L, this L will have to be updated. Other option is to discard those data points, but that forbids the density to become infinite, $\mathcal{R}_{\mathcal{D}} \not\to 0$.

2. That the recursive estimation of the L does not go to infinity.

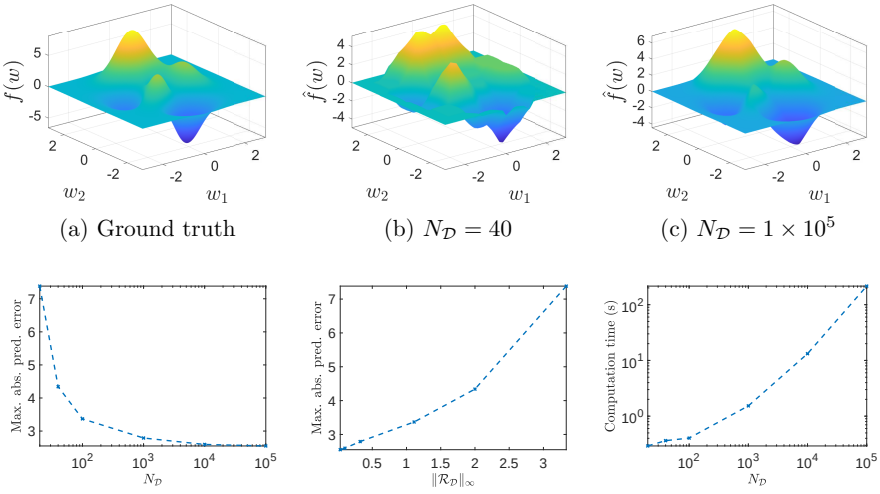
Both facts prevent the POKI method to be proven as a learning method. However, the constraint added for the hybrid method proposed in Chapter 4 allowed the derivation of the learning proof.

In any case, given the non-parametric interpolation nature of the KI algorithm, it seems intuitive to assume that the prediction error eventually decreases as more data are considered to make predictions. This is illustrated in the following example:

Example B.3. *Consider MATLAB® 's* `peaks` *function, represented in Figure B.3a and defined as*

$$\begin{aligned} f(w) &= 3(1-w_1)^2 \exp(-(w_1^2) - (w_2+1)^2) \\ &\quad -10\left(\frac{w_1}{5} - w_1^3 - w_2^5\right)\exp(-w_1^2 - w_2^2) - \frac{1}{3}\exp\left(-(w_1+1)^2 - w_2^2\right). \end{aligned} \tag{B.7}$$

Figure B.3 shows the learning feature of the KI algorithms, predicting with different grids in $\mathcal{D}$ and a separate grid of $N_{\text{test}} = 122500$ for testing. The Hölder parameters are set to $L = 5$ and $p = 1$.



(a) Ground truth (b) $N_\mathcal{D} = 40$ (c) $N_\mathcal{D} = 1 \times 10^5$

(d) μ w.r.t. $N_\mathcal{D}$ (e) μ w.r.t. $\mathcal{R}_\mathcal{D}$ (f) $\mathcal{O}(N_\mathcal{D})$

Figure B.3: Illustration of the learning ability of KI. (a) Shows the ground truth function; (b) and (c) represent the KI prediction given data sets of $N_\mathcal{D} = 40$ and 1×10^5 points, respectively; (d) and (e) show the maximum prediction error μ w.r.t. the cardinality of the data set $N_\mathcal{D}$ and radius of the density $\mathcal{R}_\mathcal{D}$. (f) illustrates the computation times w.r.t. $N_\mathcal{D}$. Notice the logarithmic scale in figures (d) and (f).

Appendix C

Stability of MPC

This appendix presents the stability analysis of nonlinear discrete-time systems, controlled by standard MPC laws. An initial analysis considers the standard nominal case, that is, in the absence of noise and uncertainty, and with a state-feedback configuration. Then, the extension to robust scenarios (with uncertainty) will be presented.

C.1 Stability of nonlinear systems

Consider the system described by equation (1.17), with an equilibrium point at the origin. If the system is controlled by an automatic control law such that $u(k) = \kappa(x(k))$, then it can be posed in an autonomous formulation, such that $x(k+1) = F(x(k), \kappa(x(k))) = F(x(k))$. [1]

Definition C.1 (Stability)**.** *A system $x(k+1) = F(x(k))$ with $0 = F(0)$ is stable in the origin if for all $\delta > 0$ there exists an $\epsilon = \epsilon(\delta, k)$ such that*

$$\forall x(0) : \|x(0)\| \leq \epsilon \Rightarrow \|x(k)\| \leq \delta, \forall k \geq 0. \tag{C.1}$$

If ϵ is constant, i.e., $\epsilon \neq \epsilon(k)$, the condition is referred to as *uniform stability*. This condition can be read as follows: for a given initial state $x(0)$ in a region around the origin (bounded by ϵ), the system's trajectory is bounded by δ. Besides, if the trajectory converges asymptotically to the origin while being stable, then the system is said to be *asymptotically stable*. The convergence property is also referred to as *attractivity*.

[1] For simplicity, $x(k+1)$ may be denoted x^+.

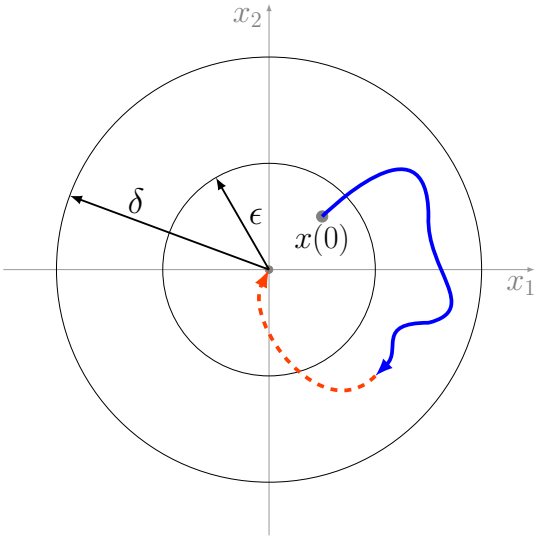


Figure C.1: Representation of a stable system, for a two-dimensional state-space. Uniform stability is represented by the trajectory in blue, and asymptotic stability by the dashed orange line.

Definition C.2 (Asymptotic stability)**.** *A system* $x(k+1) = F(x(k))$ *is asymptotically stable in the origin if it is stable and if there exist a constant* $\epsilon > 0$ *such that*

$$\forall x(0) : \|x(0)\| \leq \epsilon \Rightarrow \lim_{k\to\infty} x(k) = 0. \tag{C.2}$$

A representation of both definitions is sketched in Figure C.1. Note that the opposite relation does not hold in general, that is, if a system converges to the origin it does not imply that it is stable [155]. Several mathematical tools are useful for the stability analysis, so they are defined next.

Definition C.3 ($\mathcal{K}$-function)**.** *A function* $\alpha(\cdot) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ *is a* $\mathcal{K}$*-function if it is continuous, strictly increasing and if* $\alpha(0) = 0$*.*

Definition C.4 ($\mathcal{K}_\infty$-function)**.** *A function* $\alpha(\cdot) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ *is a* $\mathcal{K}_\infty$*-function if it is a* $\mathcal{K}$*-function and if* $\lim_{a\to\infty} \alpha(a) = \infty$*.*

Definition C.5 ($\mathcal{KL}$-function)**.** *A function* $\beta(\cdot,\cdot) : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ *is a* $\mathcal{KL}$*-function if* $\beta(a,k)$ *is a* $\mathcal{K}$*-function in* a *for all fixed* $k \geq 0$ *and if it is decreasing in* k *for all fixed* $a \geq 0$*, such that* $\lim_{k\to\infty} \beta(a,k) = 0$*.*

Definition C.6 (Positive definiteness)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is said to be positive definite if there exists a $\mathcal{K}$-function α such that*

$$\alpha(\|x\|) \leq f(x), \forall x. \tag{C.3}$$

Some interesting properties of $\mathcal{K}$-functions can be found in [155]. Notice that the asymptotic stability condition in Def. C.2 can be reformulated stating that there exits a $\mathcal{KL}$-function β such that

$$\|x(k)\| \leq \beta(\|x(0)\|, k) \tag{C.4}$$

for all $k \geq 0$ and all $x(0)$ such that $\|x(0)\| \leq \epsilon$.

In what is to follow, an understanding of the theory of invariant sets is required [25, 71].

Definition C.7 (Positive invariant set)**.** *A set $\Omega \subseteq \mathbb{R}^{n_x}$ is a positive invariant set of the system (1.17) if*

$$\forall x \in \Omega \Rightarrow F(x) \in \Omega. \tag{C.5}$$

This implies that if $x(0) \in \Omega$, then $x(k) \in \Omega, \forall k$. If the control law is taken into account, instead of the autonomous version of the system, Ω is said to be a *control invariant set*, and it suffices for one $u(k)$ to exist such that $F(x(k), u(k)) \in \Omega$. Note that the set of equilibrium points of the system is an example of invariant control sets.

If the constraints to which the system is subject are taken into account, recall $u \in \mathcal{U}$ and $x \in \mathcal{X}$, then the invariant set must be such that $\Omega \subseteq \mathcal{X}$. Then, the maximum invariant control set is denoted $\mathcal{C}_\infty(\mathcal{X})$, and it is such that contains all others control invariant sets of the system.

Note that if $x(0) \in \mathcal{C}_\infty(\mathcal{X})$, there exists a control sequence able to satisfy the constraints of the system. It is also useful to define the following sets

Definition C.8 (Controllable set in i steps)**.** *The controllable set $\mathcal{K}_i(\mathcal{X}, \Omega)$ is the set of states for which an admissible control sequence drives the system to $\Omega \subseteq \mathcal{X}$ in i steps, following an admissible trajectory, i.e.,*

$$\mathcal{K}_i(\mathcal{X}, \Omega) = \{x(0) \in \mathcal{X} : \forall k \in \mathbb{I}_0^{i-1} \exists u(k) \in \mathcal{U} | x(k) \in \mathcal{X} \wedge x(i) \in \Omega\}. \tag{C.6}$$

Definition C.9 (Stabilizing set in i steps)**.** *The stabilizing set in i steps to the invariant set $\Omega \in \mathcal{X}$ is the set of states $\mathcal{S}_i(\mathcal{X}, \Omega)$ for which a feasible control sequence exists such that the system is steered to Ω in i steps, following an admissible trajectory:*

$$S_i(\mathcal{X}, \Omega) = \{x(0) \in \mathcal{X} : \forall k \in \mathbb{I}_0^{i-1} \exists u(k) \in \mathcal{U} | x(k) \in \mathcal{X} \wedge x(i) \in \Omega\}. \tag{C.7}$$

The only difference between $\mathcal{K}_i(\mathcal{X}, \Omega)$ and $S_i(\mathcal{X}, \Omega)$ is that the latter requires Ω to be an invariant (control) set.

C.2 Lyapunov stability

A different and efficient method to analyse the stability of nonlinear systems was developed by A.M. Lyapunov at the end of the nineteenth century [91]. The method makes use of a function, often referred to as Lyapunov function, and denoted $V(x)$, which resembles the energy function of dynamical systems, but can be generally applied to any dynamical system. The idea behind Lyapunov stability is proving that the "energy" of the system is somehow decreasing with time.

Definition C.10 (Lyapunov function)**.** *A function $V(x) : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$, associated to a system (1.17) is a Lyapunov function (LF) if it is positive definite and the following conditions are satisfied:*

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|) \tag{C.8}$$
$$V(F(x)) - V(x) = \Delta V(x) \leq 0, \tag{C.9}$$

for two $\mathcal{K}$-functions α_1, α_2 and for all $x \in \mathcal{B}(\epsilon)$, for some $\epsilon > 0$.

Analogously, a *control Lyapunov function* (CLF) includes the control action in (C.9). Then, Lyapunov stability states the following:

Theorem C.1. *Given a dynamical system $x^+ = F(x)$ with $0 = F(0)$, if there exists an associated Lyapunov function, then the origin is an stable equilibrium point of the system.*

It can be proven that every level set of $V(x)$ defined as

$$\Omega = \{x : V(x) \leq \delta\}, \tag{C.10}$$

with $\delta \leq \alpha_1(\epsilon)$, is a positive invariant set of the system, since $\Omega \in \mathcal{B}(\epsilon)$.

Theorem C.2. *If the Lyapunov function is such that*

$$\Delta V(x) \leq -\alpha_3(\|x\|), \tag{C.11}$$

for certain $\mathcal{K}$-function α_3, then the system is asymptotically stable.

The proof of this theorems can be found in [79, 133].

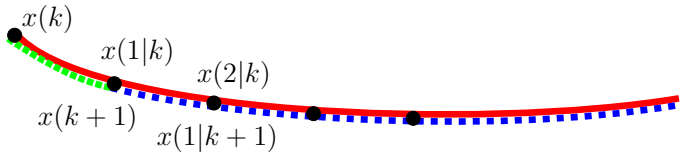


Figure C.2: Representation of the optimal costs. The red solid line represents $V_\infty^*(x(k))$, the green dotted line the stage cost $\ell(x(k), u^*(k))$ from $x(k)$ to $F(x(k), u^*(k))$, and the blue dashed line represents $V_\infty^*(x(k+1))$. Note that the prediction $x(1|k)$ is equal to the real trajectory, that yields $x(k+1)$.

C.3 Stability of optimal controllers

In general, finding a Lyapunov function for nonlinear systems is not an easy task. However, the Lyapunov stability analysis is particularly relevant in the field of predictive controllers, since under some conditions, the cost that is to be minimized in the optimization problem can serve as a Lyapunov function.

Recall the optimal control problem given by equations (1.20). From Bellman's optimality principle and provided that the exact prediction is obtained, it can be proven that the optimal solution of the problem $P_\infty(x(k))$, denoted $\mathbf{u}^*(k)$, is the same as the solution in the following step $\mathbf{u}^*(k+1)$, without the first element $u^*(k)$, which was applied to the system.

Hence, both costs (recall the definition of the optimal cost in equation (1.19)), differ only in the stage cost of the first iteration, as illustrated by Figure C.2. Thus,

$$V_\infty^*(x(k)) = \ell(x(k), u^*(k)) + V_\infty^*(x(k+1)). \tag{C.12}$$

Then, if $x(0)$ belongs to the invariant control set $\mathcal{C}_\infty$ and the optimal cost $V_\infty^*(x)$ is bounded, then $x(k)$ converges to the origin. To show stability, it is proven that $V_\infty^*(x)$ is a control Lyapunov function of the system.

To do so, the stage cost must be defined as a positive definite function and such that $\ell(x,u) \geq \alpha_1(\|x\|)$, for a $\mathcal{K}$-function α. A typical choice is a quadratic function in both states and inputs:

$$\ell(x,u) = \|x\|_Q^2 + \|u\|_R^2.$$

The lower bound on V (see equation (C.8)) is obtained provided that

$$V_\infty^*(x) \geq \ell(x,u) \geq \alpha_1(\|x\|), \tag{C.13}$$

and the lower bound given that

$$V_\infty^*(x) \leq \alpha_2(\|x\|), \forall x \in \Omega. \tag{C.14}$$

Finally, the decrease in ΔV (C.9) is obtained from Bellman's optimality principle in equation (C.12), yielding

$$\Delta V_\infty^*(x) = -\ell(x, \kappa_\infty(x)) \leq -\alpha_1(\|x\|). \tag{C.15}$$

Hence, $V_\infty^*(x)$ is a CLF, so the system is asymptotically stable.

C.4 Stability of MPC

The practical application of optimal control problem is model predictive controllers. However, given the fixed and receding horizon policy in MPC, Bellman's optimality principle does not hold in general, which may cause stability issues. A general stability formulation for model predictive controllers was proposed by Mayne et al. in [107].

The *feasibility region* of the problem $P_N(x)$ (1.22), denoted $\mathcal{X}_N$, is the set of initial states from which a control action exists that steers the system to the terminal region $\mathcal{X}_f$ in N steps. Hence $\mathcal{X}_N = \mathcal{K}_N(\mathcal{X}, \mathcal{X}_f)$ (see Definition C.8). However, in general it does not hold that the controllable set in $i-1$ steps is contained in the controllable set in i steps (see [79]). Thus,

$$x(k) \in \mathcal{K}_N(\mathcal{X}_f) \Rightarrow x(k+1) \in \mathcal{K}_{N-1}(\mathcal{X}_f) \not\subseteq K_N(\mathcal{X}_f), \tag{C.16}$$

so recursive feasibility of the problem is lost. [2]

The solution would consist in designing the terminal region $\mathcal{X}_f$ such that it is a control invariant set of the system, and hence making $\mathcal{X}_N = \mathcal{S}_N(\mathcal{X}_f)$ (instead of $\mathcal{K}_N(\mathcal{X}_f)$). Hence, since $x(N|k) \in \mathcal{X}_f$, all following subsequent steps would also be contained in $\mathcal{X}_f$, since it is an invariant. This implies that an extra control action $u_f = u(k+N)$ can be found to maintain $x(N|k) \in \mathcal{X}_f$, so a feasible solution for $P_N(k+1)$, as illustrated in Figure C.3 is the so-called *shifted* sequence:

$$\bar{\mathbf{u}} = (u^*(k+1), \ldots, u^*(k+N-1), u_f)\,. \tag{C.17}$$

[2] The dependence of $K_N(\mathcal{X}_f)$ with $\mathcal{X}$ is omitted, for sake of conciseness.

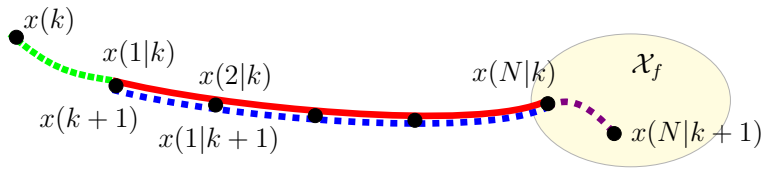


Figure C.3: Representation of the MPC costs. The green dotted line represents the stage cost $\ell(x(k), u^*(k))$, which together with the red solid line yield $V_N^*(x(k))$, and the blue dashed line is $\bar{V}_N(x(k+1))$ (not taking into account $V_f(x(N))$). If $x(N|k)$ belongs to the terminal region $\mathcal{X}_f$ and this is an invariant set, then $x(N|k+1)$ also belongs to $\mathcal{X}_f$.

From Figure C.3 it can also be intuited the calculation of both costs, denoted $\tilde{\Delta}V = \bar{V}_N(x(k+1)) - V_N^*(x(k))$, given the definition of $V_N(x)$ in (1.23) (on page 19). It is derived as follows:

$$\begin{aligned}\tilde{\Delta}V = &\Bigg[\underbrace{\sum_{j=0}^{N-2} \ell(x(j|k+1), u(k+j+1))}_{A} \\ &+ \underbrace{\ell(x(N-1|k+1), u_f)} + V_f(x(N|k+1))\Bigg] \\ &- \Bigg[\underbrace{\ell(x(k), u(k))} + \underbrace{\sum_{j=1}^{N-1} \ell(x(j|k), u(k+j))}_{A} + V_f(x(N|k))\Bigg], \end{aligned} \tag{C.18}$$

where the braces follow the colour code of Figure C.3. Note that the terms labelled with A cancel out, yielding

$$\begin{aligned}\tilde{\Delta}V = &- \ell(x(k), u(k)) \\ &+ [\ell(x(N-1|k+1), u_f) + V_f(x(N|k+1)) - V_f(x(N|k))]. \end{aligned} \tag{C.19}$$

Then, a terminal cost is required, designed as a CLF such for all $x \in \mathcal{X}_f$,

$$\min_{u \in \mathcal{U}} \{V_f(F(x,u)) - V_f(x) : F(x,u) \in \mathcal{X}_f\} \leq -\ell(x,u) \leq 0. \tag{C.20}$$

This accounts for the term in brackets in (C.19), resulting in

$$\tilde{\Delta}V = \bar{V}_N(x(k+1)) - V_N^*(x(k)) \leq -\ell(x(k), u^*(k)). \tag{C.21}$$

By optimality, $V_N^*(x(k+1)) \leq \bar{V}_N(x(k+1))$, and hence

$$\Delta V = V_N^*(x(k+1)) - V_N^*(x(k)) \leq \tilde{\Delta} V \leq -\ell(x(k), u^*(k)), \tag{C.22}$$

which proofs (C.9).

Another important property is the following:

Property C.1. *For every state $x \in \mathcal{X}_f$, its optimal cost $V_N^*(x)$ is less or equal to the terminal cost $V_f(x)$.*

Proof.

Let $x(0) \in \mathcal{X}_f$ and define $u_f = \min_{u \in \mathcal{U}} \{V_f(F(x,u)) - V_f(x) : F(x,u) \in \mathcal{X}_f\}$, denoting the state $x(k+1) = F(x(k), u_f(x(k)))$. Given the definition of V_f,

$$V_f(x(k+1)) - V_f(x(k)) \leq -\ell(x(k), u_f(k)).$$

Summing for the prediction sequence,

$$V_f(x(N)) - V_f(x(0)) \leq -\sum_{j=0}^{N-1} \ell(x(j), u_f(j)).$$

Since $x(k) \in \mathcal{X}_f$, the sequence of $u_f(k)$ is a feasible solution of $P_N(x(0))$, and hence,

$$V_N^*(x(0)) \leq \sum_{j=0}^{N-1} \ell(x(j), u_f(j)) + V_f(x(N)) \leq V_f(x(0)). \tag{C.23}$$

□

In summary,

- $V_N^*(x) \geq \ell(x,u) \geq \alpha_1(\|x\|), \forall x \in \mathcal{X}_N$,
- $V_N^*(x) \leq V_f(x) \leq \alpha_2(\|x\|), \forall x \in \mathcal{X}_f$,
- $V_N^*(x^+) - V_N^*(x) \leq -\ell(x, \kappa_N(x)) \leq -\alpha_1(\|x\|), \forall x \in \mathcal{X}_N$,

and hence, $V_N^*(x)$ is a Lyapunov function of the system, guaranteeing stability in the feasible region $\mathcal{X}_N$, which is an admissible invariant set.

Note that the design ingredients of the standard MPC problem are the control horizon N, the stage ℓ and terminal V_f costs, and the terminal region $\mathcal{X}_f$. An easy design of stabilizing MPC transforms the inequality terminal constraint $x(N|k) \in \mathcal{X}_f$ into a equality terminal constraint, forcing that $x(N|k) = 0$.

In [83] the conditions under which the terminal constraint can be removed were studied. This is done by a factor $\lambda \geq 1$ that weights the terminal cost. It is a great advance in MPC stability theory, since the terminal constraint is the most difficult ingredient of $P_N(x)$ to calculate. The controllers proposed in this thesis have also avoided the use of a terminal constraint. In addition, it is proven that this weight λ enlarges the domain of attraction of the controller.

Another method to enlarge the feasibility region is by considering a prediction horizon N_p larger than the control horizon N_c. The prediction in time steps among N_c and N_p is carried out by a terminal control law, $\kappa_f(x)$. A general framework for the application of prediction horizons larger than the control horizon for nonlinear systems was developed by D. Limon in [79]. In Chapter 2, a data-based MPC with $N_p \geq N_c$ was proposed.

C.5 Robust stability

Note that both in the optimal control problem and in the nominal predictive control case, the predictions are ideal, yielding the same result as the real transition function, lacking prediction error or uncertainty, i.e., they were such that $x(k+1) = \hat{x}(1|k)$ (see figures C.2 and C.3).

Usually, when relying on a model of the real plant, modelling errors, noise and disturbances are going to be present. This was definitely the case in the data-based scenario considered in this thesis, where the plant was a black-box system, and a model had to be derived based on input-output observations. Hence, the stability analysis presented in the previous section may not hold in general: the real trajectory of the system may not be optimal, since it will not be the predicted one; or it may even violate the constraints of feasible regions or diverge, becoming unstable. Therefore, a broader scope must be considered to account for the robust stability of MPC under uncertainties.

For an introduction on robust stability analysis, we consider a state-feedback system with bounded additive uncertainty:

$$x(k+1) = F(x(k), u(k), d(k)), \tag{C.24}$$

where the uncertainty d is confined in a compact set $\mathcal{B}(\mu)$. The system is

assumed to be nominally and asymptotically stable (i.e., that satisfies the conditions of the previous section for $d = 0$).

A first approach to consider uncertainty in the stability analysis could be to study if a bounded uncertainty renders a bounded evolution of the state. It is interesting to study properties such us bounded-input bounded-output (BIBO) stability, asymptotic gains (AG) or stability margins (SM) of the system, in particular if the uncertainty is bounded by a $\mathcal{K}$-function of the norm of the state [81].

If not, the *input-to-state stability* (ISS) theory offers methodology to account for a robust analysis of systems subject to uncertainty. Input-to-state stability was first described by E. Sontag [145], applied to continuous-time systems, and extended to discrete-time systems by Z. Jiang and Y. Wang in [66]. A general framework of ISS for robust MPC was proposed by Limon et al. in [81].

Definition C.11 (Input-to-state stability)**.** *A system $x(k+1) = F(x(k), d(k))$ is input-to-state stable (ISS) w.r.t. d if there exists a $\mathcal{KL}$-function β and a $\mathcal{K}$-function α such that*

$$\|x(k)\| \leq \beta(\|x(0)\|, k) + \sup_{j\in[0,k]} \alpha(\|d(j)\|). \tag{C.25}$$

An ISS system implies that bounded uncertainties have a bounded effect on the evolution of the system. Further extension is the so-called *input-to-state practical stability* (ISpS) [146], in which the norm of the state is not only bounded by the $\mathcal{KL}$-function β and the $\mathcal{K}$-function α, but also by a constant c (cf. (C.25)).

Following the procedure in previous section, a positive invariant set (cf. Definition C.7) can be extended to a *robust positive invariant* (RPI) set such that Γ is a RPI if $F(x, d) \in \Gamma$ for all $x \in \Gamma$ and $d \in \mathcal{B}(\mu)$.

Analogously, an ISpS-Lyapunov function (cf. Definition C.10) is defined as follows [81].

Definition C.12 (ISpS-Lyapunov function)**.** *A function $V : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$ is called an ISpS-Lyapunov function in for the system, w.r.t. d, in a RPI set Γ containing the origin if there exists a compact set $\{0\} \in \Omega \subseteq \Gamma$ such that*

$$V(x) \geq \alpha_1(\|x\|), \forall x \in \Gamma \tag{C.26a}$$

$$V(x) \leq \alpha_2(\|x\|) + c_1, \forall x \in \Omega \tag{C.26b}$$

$$V(F(x, \kappa(x), d)) - V(x) \leq -\alpha_3(\|x\|) + \gamma(\|d\|) + c_2, \forall x \in \Gamma, d \in \mathcal{B}(\mu), \tag{C.26c}$$

where $\alpha_1, \alpha_2, \alpha_3$ are $\mathcal{K}_\infty$-functions, γ is a $\mathcal{K}$-function and $c_1, c_2 \geq 0$ are constants such that if they are zero, $V(x)$ is called a ISS-Lyapunov function.

Then, if a system admits an IS(p)S-Lyapunov function, it is IS(p)S in Γ, w.r.t. d, as proven in [130].

Once the robust analysis of the stability of uncertain system has been performed, the conditions to design robust stabilizing controllers must be addressed. Standard approaches in the MPC literature [133] usually consider the reachable sets that bound the trajectory of the closed-loop system, by means of *tube-based* MPC [21, 106] or extending the robust case to MPCs based on nominal predictions [80, 111].

In Chapter 3, a set of tightened constraints [80] was designed to counteract the effect of uncertainty in the predictions, tailored to a data-based model predictive control framework.

List of Figures

1.1 Scheme of the overall framework. 3
1.2 Simulation of a tank. 13
1.3 Identification of a tank via least squares 13
1.4 Illustration of the kinky inference algorithm. 16

2.1 Set of equilibrium points of the CSTR. 44
2.2 Relay test response. 45
2.3 Chirp signals used to obtain the data set. 45
2.4 CSTR dataset 46
2.5 Test used for cross-validation. 46
2.6 Prediction error w.r.t. memory horizons 48
2.7 Histogram of the validation error 48
2.8 Partition of the input space $\mathcal{W}$. 49
2.9 Ideal ODEs-MPC. 51
2.10 Primary KI-MPC. 51
2.11 Data-based SPKI predictive control applied to a CSTR. 52
2.12 Boxplot comparison of different prediction models 53

3.1 Propagation of the prediction error. 57
3.2 Simulation obtained applying a sequence of chirp signals. . . . 66
3.3 Learning-based MPC for a nonlinear constrained CSTR 67

4.1 Illustration of the performance of CHoKI 79

4.2 Learning ability of CHoKI 79
4.3 Quadruple-tank process scheme. 85
4.4 Set of equilibrium points of the quadruple tank process. 87
4.5 Application of random input signals 87
4.6 Performance of the CHoKI-based MPC 89
4.7 Comparison of the performance 89
4.8 Set of tightened constraints 90
4.9 CHoKI-based MPC applied to an ICR 92

5.1 Propagation of the predictions. 101
5.2 Experiments to obtain $\mathcal{D}(0)$. 108
5.3 Optimization of the mmory horizons 109
5.4 Online learning MPC applied to a quadruple tank process. . . 110
5.5 Online learning under random references 112

6.1 Exploration-exploitation algorithm. 116
6.2 State-space of the CSTR 123
6.3 Initial data set, consisting of $N_{\mathcal{D}} = 300$ points. 124
6.4 Comparison between the offline and online MPCs. 125
6.5 Initial knowledge of the CSTR 126
6.6 Online learning MPC. 128

A.1 Pseudorandom and chirp signals applied to a water tank . . . 135
A.2 Four dimensional input data set 137
A.3 Application of principal component analysis 137
A.4 Data set range w.r.t. PCA 138

B.1 Effect of the regularization term in LACKI. 141
B.2 Performance of the LACKI and the POKI predictors. 142
B.3 Illustration of the learning ability of KI 146

C.1 Representation of a stable system 148

C.2 Optimal costs 151
C.3 MPC costs 153

List of Tables

2.1 Parameters of the system 43
2.2 Parameters of the chirp signals. 44

3.1 Parameters of the system 65

4.1 Parameters of the quadruple tank process. 86

A.1 Standard deviation of the principal components data set . . . 138

Bibliography

[1] Adetola, V., DeHaan, D., and Guay, M. Adaptive model predictive control for constrained nonlinear systems. *Systems & Control Letters 58*, 5 (2009), 320–326. (Cited on pages 5 and 20).

[2] Akametalu, A. K., Fisac, J. F., Gillula, J. H., Kaynama, S., Zeilinger, M. N., and Tomlin, C. J. Reachability-based safe learning with Gaussian processes. In *53rd IEEE Conference on Decision and Control* (2014), IEEE, pp. 1424–1431. (Cited on pages 4 and 6).

[3] Alamo, T., Bravo, J. M., and Camacho, E. F. Guaranteed state estimation by zonotopes. *Automatica 41*, 6 (2005), 1035–1043. (Cited on page 6).

[4] Alamo, T., Manzano, J. M., and Camacho, E. Robust design through probabilistic maximization. In *Uncertainty in Complex Networked Systems*. Springer, 2018, pp. 247–274. (Cited on page 131).

[5] Alamo, T., Tempo, R., and Camacho, E. F. Randomized strategies for probabilistic solutions of uncertain feasibility and optimization problems. *IEEE Transactions on Automatic Control 54*, 11 (2009), 2545–2559. (Cited on page 5).

[6] Alamo, T., Tempo, R., Luque, A., and Ramirez, D. R. Randomized methods for design of uncertain systems: Sample complexity and sequential algorithms. *Automatica 52* (2015), 160–172. (Cited on page 5).

[7] Allgöwer, F., and Zheng, A. *Nonlinear model predictive control*, vol. 26. Birkhäuser, 2012. (Cited on page 19).

[8] Alpcan, T. Dual control with active learning using Gaussian process regression. *arXiv preprint arXiv:1105.2211* (2011). (Cited on pages 4 and 139).

[9] Alvarado, I., Limon, D., Muñoz de la Peña, D., Maestre, J. M., Ridao, M., Scheu, H., Marquardt, W., Negenborn, R., De Schutter, B., Valencia, F., et al. A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark. *Journal of Process Control 21*, 5 (2011), 800–815. (Cited on page 85).

[10] Amrit, R., Rawlings, J. B., and Angeli, D. Economic optimization using model predictive control with a terminal cost. *Annual Reviews in Control 35*, 2 (2011), 178–186. (Cited on page 20).

[11] Angeli, D., Amrit, R., and Rawlings, J. B. On average performance and stability of economic model predictive control. *IEEE transactions on automatic control 57*, 7 (2011), 1615–1626. (Cited on page 20).

[12] Åström, K.-J., and Hägglund, T. *Advanced PID control*. 2006. (Cited on pages 18, 43, and 133).

[13] Aswani, A., Gonzalez, H., Sastry, S. S., and Tomlin, C. Provably safe and robust learning-based model predictive control. *Automatica 49*, 5 (2013), 1216–1226. (Cited on pages 4, 7, 69, and 94).

[14] Baudat, G., and Anouar, F. Generalized discriminant analysis using a kernel approach. *Neural computation 12*, 10 (2000), 2385–2404. (Cited on page 135).

[15] Beliakov, G. Interpolation of Lipschitz functions. *Journal of computational and applied mathematics 196*, 1 (2006), 20–44. (Cited on pages 14 and 22).

[16] Bellman, R. Dynamic programming. *Science 153*, 3731 (1966), 34–37. (Cited on page 18).

[17] Benosman, M. Model-based vs data-driven adaptive control: An overview. *International Journal of Adaptive Control and Signal Processing 32*, 5 (2018), 753–776. (Cited on pages 4 and 20).

[18] Berkenkamp, F., and Schoellig, A. P. Safe and robust learning control with Gaussian processes. In *2015 European Control Conference (ECC)* (2015), IEEE, pp. 2496–2501. (Cited on page 4).

[19] Berkenkamp, F., Turchetta, M., Schoellig, A., and Krause, A. Safe model-based reinforcement learning with stability guarantees.

In *Advances in neural information processing systems* (2017), pp. 908–918. (Cited on page 20).

[20] Bertsekas, D. P. Nonlinear programming. *Journal of the Operational Research Society 48*, 3 (1997), 334–334. (Cited on page 19).

[21] Bertsekas, D. P., and Rhodes, I. B. On the minimax reachability of target sets and target tubes. *Automatica 7*, 2 (1971), 233–247. (Cited on page 157).

[22] Biegler, L. T. *Nonlinear programming: concepts, algorithms, and applications to chemical processes*, vol. 10. Siam, 2010. (Cited on page 19).

[23] Billings, S., and Fadzil, M. The practical identification of systems with nonlinearities. *IFAC Proceedings Volumes 18*, 5 (1985), 155–160. (Cited on page 133).

[24] Blaas, A., Manzano, J. M., Limon, D., and Calliess, J.-P. Localised kinky inference. In *2019 18th European Control Conference (ECC)* (2019), IEEE, pp. 985–992. (Cited on page 30).

[25] Blanchini, F. Set invariance in control. *Automatica 35*, 11 (1999), 1747–1767. (Cited on page 149).

[26] Bomberger, J. D., and Seborg, D. E. Determination of model order for NARX models directly from input-output data. *Journal of Process Control 8*, 5-6 (1998), 459–468. (Cited on page 10).

[27] Boyd, S., Boyd, S. P., and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004. (Cited on page 19).

[28] Bravo, J. M., Alamo, T., and Camacho, E. F. Robust MPC of constrained discrete-time nonlinear systems based on approximated reachable sets. *Automatica 42*, 10 (2006), 1745–1751. (Cited on page 6).

[29] Bravo, J. M., Alamo, T., Gegúndez, M., and Marín, D. Combined stochastic and deterministic interval predictor for time-varying systems. In *2015 23rd Mediterranean Conference on Control and Automation (MED)* (2015), IEEE, pp. 833–839. (Cited on page 6).

[30] Bristol, E. H. Pattern recognition: An alternative to parameter identification in adaptive control. *Automatica 13*, 2 (1977), 197–202. (Cited on page 5).

[31] Bryson, A. E. *Applied optimal control: optimization, estimation and control*. CRC Press, 1975. (Cited on page 18).

[32] Calafiore, G., and Dabbene, F. *Probabilistic and randomized methods for design under uncertainty*. Springer, 2006. (Cited on page 60).

[33] Calafiore, G. C., and Campi, M. C. The scenario approach to robust control design. *IEEE Transactions on automatic control 51*, 5 (2006), 742–753. (Cited on page 5).

[34] Calliess, J.-P. *Conservative decision-making and inference in uncertain dynamical systems*. PhD thesis, University of Oxford, 2014. (Cited on pages 10, 13, 14, 27, 29, 30, 115, 139, and 143).

[35] Calliess, J.-P. Bayesian Lipschitz constant estimation and quadrature. In *Workshop on Probabilistic Integration, 29th Conference on Neural Information Processing Systems, 2015.* (2015). (Cited on pages 114 and 144).

[36] Calliess, J.-P. Lazily adapted constant kinky inference for nonparametric regression and model-reference adaptive control. *arXiv preprint arXiv:1701.00178* (2016). (Cited on pages 15, 74, 115, 123, 140, 143, and 144).

[37] Calliess, J.-P. Lipschitz optimisation for Lipschitz interpolation. *American Control Conference (ACC), 2017* (2017), 3141–3146. (Cited on pages 15 and 141).

[38] Camacho, E. F., and Bordons, C. *Model predictive control*. Springer Science & Business Media, 2013. (Cited on pages 7 and 19).

[39] Camponogara, E., Jia, D., Krogh, B. H., and Talukdar, S. Distributed model predictive control. *IEEE control systems magazine 22*, 1 (2002), 44–52. (Cited on page 19).

[40] Canale, M., Fagiano, L., and Signorile, M. C. Nonlinear model predictive control from data: a set membership approach. *International Journal of Robust and Nonlinear Control 24*, 1 (2014), 123–139. (Cited on pages 4, 7, 15, 22, 69, and 78).

[41] Chen, S., and Billings, S. A. Representations of non-linear systems: the NARMAX model. *International journal of control 49*, 3 (1989), 1013–1032. (Cited on pages 4 and 9).

[42] Christofides, P. D., Scattolini, R., de la Pena, D. M., and Liu, J. Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering 51* (2013), 21–41. (Cited on page 19).

[43] Di Cairano, S., Bernardini, D., Bemporad, A., and Kolmanovsky, I. V. Stochastic MPC with learning for driver-predictive vehicle control and its application to HEV energy management. *IEEE Transactions on Control Systems Technology 22*, 3 (2013), 1018–1031. (Cited on pages 5, 21, and 93).

[44] Diehl, M., Amrit, R., and Rawlings, J. B. A Lyapunov function for economic optimizing model predictive control. *IEEE Transactions on Automatic Control 56*, 3 (2010), 703–707. (Cited on pages 7 and 20).

[45] Eckart, C., and Young, G. The approximation of one matrix by another of lower rank. *Psychometrika 1*, 3 (1936), 211–218. (Cited on page 136).

[46] Elmasri, R. *Fundamentals of database systems*. Pearson Education India, 2008. (Cited on page 138).

[47] Faulwasser, T., Grüne, L., Müller, M. A., et al. Economic nonlinear model predictive control. *Foundations and Trends® in Systems and Control 5*, 1 (2018), 1–98. (Cited on page 20).

[48] Faulwasser, T., Hagenmeyer, V., and Findeisen, R. Constrained reachability and trajectory generation for flat systems. *Automatica 50*, 4 (2014), 1151–1159. (Cited on page 6).

[49] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. From data mining to knowledge discovery in databases. *AI magazine 17*, 3 (1996), 37–37. (Cited on page 138).

[50] Fernández-Cantí, R. M., Tornil-Sin, S., Blesa, J., and Puig, V. Non-linear set-membership identification approach based on the Bayesian framework. *IET Control Theory & Applications 9*, 9 (2015), 1392–1398. (Cited on page 6).

[51] Fiacchini, M., Alamo, T., and Camacho, E. F. On the computation of convex robust control invariant sets for nonlinear systems. *Automatica 46*, 8 (2010), 1334–1338. (Cited on page 60).

[52] Fisac, J. F., Akametalu, A. K., Zeilinger, M. N., Kaynama, S., Gillula, J., and Tomlin, C. J. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control* (2018). (Cited on pages 4 and 21).

[53] Friedman, J., Hastie, T., and Tibshirani, R. *The elements of statistical learning*, vol. 1. Springer series in statistics New York, 2001. (Cited on page 5).

[54] Gros, S., and Zanon, M. Data-driven economic NMPC using reinforcement learning. *IEEE Transactions on Automatic Control* (2019). (Cited on page 20).

[55] Grüne, L., and Pannek, J. Nonlinear model predictive control. In *Nonlinear Model Predictive Control*. Springer, 2017, pp. 45–69. (Cited on page 19).

[56] Haber, R., and Unbehauen, H. Structure identification of nonlinear dynamic systems - a survey on input/output approaches. *Automatica 26*, 4 (1990), 651–677. (Cited on page 10).

[57] Han, J., Pei, J., and Kamber, M. *Data mining: concepts and techniques*. Elsevier, 2011. (Cited on page 138).

[58] Hastie, T., Tibshirani, R., and Wainwright, M. *Statistical learning with sparsity: the lasso and generalizations*. CRC press, 2015. (Cited on page 5).

[59] He, X., and Asada, H. A new method for identifying orders of input-output models for nonlinear dynamic systems. In *1993 American Control Conference* (1993), IEEE, pp. 2520–2523. (Cited on page 10).

[60] Heidarinejad, M., Liu, J., and Christofides, P. D. Economic model predictive control of nonlinear process systems using Lyapunov techniques. *AIChE Journal 58*, 3 (2012), 855–870. (Cited on page 7).

[61] Hewing, L., Kabzan, J., and Zeilinger, M. N. Cautious model predictive control using Gaussian process regression. *IEEE Transactions on Control Systems Technology* (2019). (Cited on page 84).

[62] Hewing, L., Liniger, A., and Zeilinger, M. N. Cautious NMPC with Gaussian process dynamics for autonomous miniature race cars. In *2018 European Control Conference (ECC)* (2018), IEEE, pp. 1341–1348. (Cited on pages 5 and 21).

[63] Hewing, L., Wabersich, K. P., Menner, M., and Zeilinger, M. N. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems 3* (2019). (Cited on pages 4, 7, 21, and 94).

[64] Houska, B., Ferreau, H. J., and Diehl, M. ACADO toolkit - An open-source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods 32*, 3 (2011), 298–312. (Cited on page 19).

[65] Isermann, R. *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2006. (Cited on page 5).

[66] Jiang, Z.-P., and Wang, Y. Input-to-state stability for discrete-time nonlinear systems. *Automatica 37*, 6 (2001), 857–869. (Cited on page 156).

[67] Johansen, T. A., and FOSS, B. Constructing NARMAX models using ARMAX models. *International journal of control 58*, 5 (1993), 1125–1153. (Cited on page 8).

[68] Johansson, K. H. The quadruple-tank process: A multivariable laboratory process with an adjustable zero. *IEEE Transactions on control systems technology 8*, 3 (2000), 456–465. (Cited on page 85).

[69] Jolliffe, I. T. Principal components in regression analysis. In *Principal component analysis*. Springer, 1986, pp. 129–155. (Cited on page 136).

[70] Kabzan, J., Hewing, L., Liniger, A., and Zeilinger, M. N. Learning-based model predictive control for autonomous racing. *IEEE Robotics and Automation Letters 4*, 4 (2019), 3363–3370. (Cited on page 21).

[71] Kerrigan, E. C., and Maciejowski, J. M. Invariant sets for constrained nonlinear discrete-time systems with application to feasibility in model predictive control. In *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187)* (2000), vol. 5, IEEE, pp. 4951–4956. (Cited on page 149).

[72] Kingravi, H. *Reduced-set models for improving the training and execution speed of kernel methods*. PhD thesis, Georgia Institute of Technology, 2014. (Cited on pages 115 and 139).

[73] Kocijan, J. *Modelling and control of dynamic systems using Gaussian process models*. Springer, 2016. (Cited on page 4).

[74] Koller, T., Berkenkamp, F., Turchetta, M., and Krause, A. Learning-based model predictive control for safe exploration. In *2018 IEEE Conference on Decision and Control (CDC)* (2018), IEEE, pp. 6059–6066. (Cited on page 22).

[75] Lee, D. D., and Seung, H. S. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems* (2001), pp. 556–562. (Cited on page 135).

[76] Leontaritis, I., and Billings, S. A. Input-output parametric models for non-linear systems part I: deterministic non-linear systems. *International journal of control 41*, 2 (1985), 303–328. (Cited on pages 4 and 9).

[77] Levin, A., and Narendra, K. Identification of nonlinear dynamical systems using neural networks. In *Neural systems for control*. Elsevier, 1997, pp. 129–160. (Cited on page 9).

[78] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015). (Cited on page 5).

[79] Limon, D. *Control predictivo de sistemas no lineales con restricciones: estabilidad y robustez*. PhD thesis, Universidad de Sevilla, 2002. (Cited on pages 19, 60, 150, 152, and 155).

[80] Limon, D., Alamo, T., and Camacho, E. Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.* (2002), vol. 4, IEEE, pp. 4619–4624. (Cited on page 157).

[81] Limon, D., Alamo, T., Raimondo, D., Muñoz de la Peña, D., Bravo, J., Ferramosca, A., and Camacho, E. Input-to-state stability: a unifying framework for robust model predictive control. In *Nonlinear model predictive control*. Springer, 2009, pp. 1–26. (Cited on pages 41, 54, 60, 63, 64, and 156).

[82] Limon, D., Alamo, T., Salas, F., and Camacho, E. F. Input to state stability of min–max MPC controllers for nonlinear systems with

bounded uncertainties. *Automatica 42*, 5 (2006), 797–803. (Cited on page 7).

[83] Limon, D., Alamo, T., Salas, F., and Camacho, E. F. On the stability of constrained MPC without terminal constraint. *IEEE Transactions on Automatic Control 51*, 5 (2006), 832–836. (Cited on pages 27, 34, 61, 100, and 155).

[84] Limón, D., Alvarado, I., Alamo, T., and Camacho, E. F. MPC for tracking piecewise constant references for constrained linear systems. *Automatica 44*, 9 (2008), 2382–2387. (Cited on page 20).

[85] Limon, D., Ferramosca, A., Alvarado, I., and Alamo, T. Nonlinear MPC for tracking piece-wise constant reference signals. *IEEE Transactions on Automatic Control 63*, 11 (2018), 3735–3750. (Cited on pages 20, 113, 117, and 122).

[86] Ljung, L. *System identification*. Springer, 1998. (Cited on pages 4 and 64).

[87] Lorenzen, M., Cannon, M., and Allgöwer, F. Robust MPC with recursive model update. *Automatica 103* (2019), 461–471. (Cited on pages 5 and 22).

[88] Lorenzen, M., Dabbene, F., Tempo, R., and Allgöwer, F. Stochastic MPC with offline uncertainty sampling. *Automatica 81* (2017), 176–183. (Cited on page 21).

[89] Lucia, S., Andersson, J. A., Brandt, H., Diehl, M., and Engell, S. Handling uncertainty in economic nonlinear model predictive control: A comparative case study. *Journal of Process Control 24*, 8 (2014), 1247–1259. (Cited on page 20).

[90] Lucia, S., and Karg, B. A deep learning-based approach to robust nonlinear model predictive control. *IFAC-PapersOnLine 51*, 20 (2018), 511–516. (Cited on page 22).

[91] Lyapunov, A. M. The general problem of the stability of motion. *International journal of control 55*, 3 (1992), 531–534. (Cited on page 150).

[92] Maciejowski, J. M. *Predictive control: with constraints*. Pearson education, 2002. (Cited on page 19).

[93] Maiworm, M., Limon, D., and Findeisen, R. Online gaussian process learning-based model predictive control with stability guarantees. *arXiv preprint arXiv:1911.03315* (2019). (Cited on page 22).

[94] Maiworm, M., Limon, D., Manzano, J. M., and Findeisen, R. Stability of Gaussian process learning based output feedback model predictive control. *IFAC-PapersOnLine 51*, 20 (2018), 455–461. (Cited on pages 4 and 131).

[95] Manzano, J. M., Calliess, J.-P., Muñoz de la Peña, D., and Limon, D. Online learning robust MPC: an exploration-exploitation approach. In *Accepted for publication in the 21st IFAC World Congress* (2020), IFAC. (Cited on page 114).

[96] Manzano, J. M., Limon, D., Alamo, T., and Callies, J.-P. Control predictivo basado en datos. *Actas de las XXXVIII Jornadas de Automática* (2017).

[97] Manzano, J. M., Limon, D., and Muñoz de la Peña, D. Técnicas de aprendizaje automatizado para la operación económica basada en datos de sistemas ciberfísicos. *XVII Simposio CEA de Ingeniería de Control y V Seminario de Innovación Docente en Automática* (2019).

[98] Manzano, J. M., Limon, D., Muñoz de la Peña, D., and Calliess, J.-P. Robust data-based model predictive control for nonlinear constrained systems. *IFAC-PapersOnLine 51*, 20 (2018), 505–510. (Cited on page 55).

[99] Manzano, J. M., Limon, D., Muñoz de la Peña, D., and Calliess, J.-P. Data-based robust MPC with componentwise Hölder kinky inference. In *2019 IEEE 58th Conference on Decision and Control (CDC)* (2019), IEEE, pp. 6449–6454. (Cited on page 69).

[100] Manzano, J. M., Limon, D., Muñoz de la Peña, D., and Calliess, J.-P. Output feedback MPC based on smoothed projected kinky inference. *IET Control Theory & Applications 13*, 6 (2019), 795–805. (Cited on pages 17 and 28).

[101] Manzano, J. M., Limon, D., Muñoz de la Peña, D., and Calliess, J.-P. Online learning constrained MPC based on double prediction. *Accepted in the International Journal of Robust and Nonlinear Control* (2020). (Cited on page 94).

[102] Manzano, J. M., Limon, D., Muñoz de la Peña, D., and Calliess, J.-P. Robust learning-based MPC for nonlinear constrained systems. *Automatica 117* (2020), 108948. (Cited on page 55).

[103] Manzano, J. M., Muñoz de la Peña, D., Calliess, J.-P., and Limon, D. Componentwise Hölder inference for robust learning-based MPC. *Submitted to the IEEE Transactions on Automatic Control* (2020). (Cited on page 69).

[104] Manzano, J. M., Muñoz de la Peña, D., and Limon, D. Oracle-based economic predictive control. *Submitted to Computer Applications in Chemical Engineering* (2020).

[105] Manzano, J. M., Nadales, J. M., Limon, D., and Muñoz de la Peña, D. Oracle-based economic predictive control. In *2019 IEEE 58th Conference on Decision and Control (CDC)* (2019), IEEE, pp. 4246–4251. (Cited on pages 9 and 132).

[106] Mayne, D. Q., Kerrigan, E. C., Van Wyk, E., and Falugi, P. Tube-based robust nonlinear model predictive control. *International Journal of Robust and Nonlinear Control 21*, 11 (2011), 1341–1353. (Cited on page 157).

[107] Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. Constrained model predictive control: Stability and optimality. *Automatica 36*, 6 (2000), 789–814. (Cited on pages 19 and 152).

[108] McElroy, T. *A to Z of Mathematicians*. Infobase Publishing, 2014. (Cited on page 14).

[109] Mesbah, A. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine 36*, 6 (2016), 30–44. (Cited on page 22).

[110] Mesbah, A. Stochastic model predictive control with active uncertainty learning: a survey on dual control. *Annual Reviews in Control 45* (2018), 107–117. (Cited on page 20).

[111] Michalska, H., and Mayne, D. Q. Robust receding horizon control of constrained nonlinear systems. *IEEE transactions on automatic control 38*, 11 (1993), 1623–1633. (Cited on page 157).

[112] Milanese, M., and Novara, C. Set membership identification of nonlinear systems. *Automatica 40*, 6 (2004), 957–975. (Cited on pages 14, 15, and 144).

[113] Mitchell, T. Machine learning. *McGraw-Hill Higher Education,New York* (1997). (Cited on pages 10, 75, and 144).

[114] Mitchell, T. M. Machine learning and data mining. *Communications of the ACM 42*, 11 (1999), 30–36. (Cited on page 93).

[115] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature 518*, 7540 (2015), 529. (Cited on page 5).

[116] Montgomery, D. C., Peck, E. A., and Vining, G. G. *Introduction to linear regression analysis*, vol. 821. John Wiley & Sons, 2012. (Cited on page 12).

[117] Morari, M., and Lee, J. H. Model predictive control: past, present and future. *Computers & Chemical Engineering 23*, 4-5 (1999), 667–682. (Cited on page 19).

[118] Müller, M. A., Angeli, D., and Allgöwer, F. On necessity and robustness of dissipativity in economic model predictive control. *IEEE Transactions on Automatic Control 60*, 6 (2014), 1671–1676. (Cited on page 20).

[119] Nadales, J. M., Manzano, J. M., Barriga, A., and Limon, D. FPGA parallel implementation of Lipschitz interpolation techniques. In *Submitted to the 59th IEEE Conference on Decision and Control* (2020), IEEE. (Cited on page 131).

[120] Nevistić, V. *Constrained control of nonlinear systems*. PhD thesis, ETH Zurich, 1997. (Cited on page 17).

[121] Ostafew, C. J., Schoellig, A. P., and Barfoot, T. D. Learning-based nonlinear model predictive control to improve vision-based mobile robot path-tracking in challenging outdoor environments. In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (2014), IEEE, pp. 4029–4036. (Cited on page 5).

[122] Ostafew, C. J., Schoellig, A. P., Barfoot, T. D., and Collier, J. Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking. *Journal of Field Robotics 33*, 1 (2016), 133–152. (Cited on pages 21 and 93).

[123] Pearson, K. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2*, 11 (1901), 559–572. (Cited on page 136).

[124] Piga, D., Forgione, M., Formentin, S., and Bemporad, A. Performance-oriented model learning for data-driven MPC design. *IEEE Control Systems Letters 3*, 3 (2019), 577–582. (Cited on page 21).

[125] Piga, D., Formentin, S., and Bemporad, A. Direct data-driven control of constrained systems. *IEEE Transactions on Control Systems Technology 26*, 4 (2017), 1422–1429. (Cited on page 20).

[126] Pillonetto, G., and De Nicolao, G. A new kernel-based approach for linear system identification. *Automatica 46*, 1 (2010), 81–93. (Cited on page 6).

[127] Pontryagin, L. S. *Mathematical theory of optimal processes*. Routledge, 2018. (Cited on page 18).

[128] Qin, S. J. Survey on data-driven industrial process monitoring and diagnosis. *Annual reviews in control 36*, 2 (2012), 220–234. (Cited on page 5).

[129] Qin, S. J., and Badgwell, T. A. An overview of industrial model predictive control technology. In *AIche symposium series* (1997), vol. 93, New York, NY: American Institute of Chemical Engineers, 1971-c2002., pp. 232–256. (Cited on page 19).

[130] Raimondo, D. M., Limon, D., Lazar, M., Magni, L., and ndez Camacho, E. F. Min-max model predictive control of nonlinear systems: A unifying overview on stability. *European Journal of Control 15*, 1 (2009), 5–21. (Cited on page 157).

[131] Rasmussen, C. E., and Williams, C. K. *Gaussian process for machine learning*. MIT press, 2006. (Cited on pages 13 and 21).

[132] Rawlings, J. B., Angeli, D., and Bates, C. N. Fundamentals of economic model predictive control. In *2012 IEEE 51st IEEE conference on decision and control (CDC)* (2012), IEEE, pp. 3851–3861. (Cited on pages 20 and 91).

[133] Rawlings, J. B., and Mayne, D. Q. *Model predictive control: Theory and design*. Nob Hill Pub., 2009. (Cited on pages 19, 39, 58, 59, 62, 63, 106, 109, 150, and 157).

[134] Rivera, D. E., and Jun, K. S. An integrated identification and control design methodology for multivariable process system applications. *IEEE Control Systems 20*, 3 (2000), 25–37. (Cited on pages 43 and 133).

[135] Rosolia, U., and Borrelli, F. Learning how to autonomously race a car: a predictive control approach. *IEEE Transactions on Control Systems Technology* (2019). (Cited on page 21).

[136] Salvador, J. R., Muñoz de la Peña, D., Ramirez, D., and Alamo, T. Predictive control of a water distribution system based on process historian data. *Optimal Control Applications and Methods 41*, 2 (2020), 571–586. (Cited on page 4).

[137] Salvador, J. R., Ramirez, D. R., Alamo, T., and Muñoz de la Peña, D. Offset free data driven control: application to a process control trainer. *IET Control Theory & Applications 13*, 18 (2019), 3096–3106. (Cited on pages 4 and 20).

[138] Schwenkel, L., Gharbi, M., Trimpe, S., and Ebenbauer, C. Online learning with stability guarantees: A memory-based real-time model predictive controller. *arXiv preprint arXiv:1812.09582* (2018). (Cited on pages 5 and 6).

[139] Seborg, D. E., Mellichamp, D. A., Edgar, T. F., and Doyle III, F. J. *Process dynamics and control*. John Wiley & Sons, 2010. (Cited on page 42).

[140] Shawe-Taylor, J., Cristianini, N., et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004. (Cited on page 5).

[141] Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Deylon, B., Glorennec, P.-Y., Hjalmarsson, H., and Juditsky, A. *Nonlinear black-box modeling in system identification: a unified overview*. Linköping University, 1995. (Cited on page 9).

[142] Sohrab, H. H. *Basic real analysis*, vol. 231. Springer, 2003. (Cited on page 14).

[143] Soloperto, R., Müller, M. A., Trimpe, S., and Allgöwer, F. Learning-based robust model predictive control with state-dependent uncertainty. *IFAC-PapersOnLine 51*, 20 (2018), 442–447. (Cited on page 22).

[144] Sontag, E. D. On the observability of polynomial systems, I: Finite-time problems. *SIAM Journal on Control and Optimization 17*, 1 (1979), 139–151. (Cited on page 9).

[145] Sontag, E. D. Smooth stabilization implies coprime factorization. *IEEE transactions on automatic control 34*, 4 (1989), 435–443. (Cited on page 156).

[146] Sontag, E. D., and Wang, Y. New characterizations of input-to-state stability. *IEEE transactions on automatic control 41*, 9 (1996), 1283–1294. (Cited on page 156).

[147] Sra, S., Nowozin, S., and Wright, S. J. *Optimization for machine learning*. Mit Press, 2012. (Cited on page 5).

[148] Stephant, J., Charara, A., and Meizel, D. Virtual sensor: Application to vehicle sideslip angle and transversal forces. *IEEE Transactions on industrial electronics 51*, 2 (2004), 278–289. (Cited on page 6).

[149] Sukharev, A. Optimal method of constructing best uniform approximations for functions of a certain class. *USSR Computational Mathematics and Mathematical Physics 18*, 2 (1978), 21–31. (Cited on page 14).

[150] Sutton, R. S., and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018. (Cited on page 4).

[151] Suykens, J. A., and Vandewalle, J. Least squares support vector machine classifiers. *Neural processing letters 9*, 3 (1999), 293–300. (Cited on page 5).

[152] Tan, H., Feng, G., Feng, J., Wang, W., Zhang, Y.-J., and Li, F. A tensor-based method for missing traffic data completion. *Transportation Research Part C: Emerging Technologies 28* (2013), 15–27. (Cited on page 6).

[153] Tanaskovic, M., Fagiano, L., and Gligorovski, V. Adaptive model predictive control for linear time varying MIMO systems. *Automatica 105* (2019), 237–245. (Cited on pages 5 and 20).

[154] Terzi, E., Fagiano, L., Farina, M., and Scattolini, R. Learning-based predictive control for linear systems: a unitary approach. *Automatica 108* (2019), 108473. (Cited on page 22).

[155] Vidyasagar, M. *Nonlinear Systems Theory*. Prentice-Hall Englewood Cliffs, 1993. (Cited on pages 148 and 149).

[156] Wabersich, K. P., and Zeilinger, M. N. Safe exploration of nonlinear dynamical systems: A predictive safety filter for reinforcement learning. *arXiv:1812.05506* (2018). (Cited on pages 5 and 21).

[157] Wold, S., Esbensen, K., and Geladi, P. Principal component analysis. *Chemometrics and intelligent laboratory systems 2*, 1-3 (1987), 37–52. (Cited on page 30).

[158] Xiang, C. Existence of global input-output model for nonlinear systems. In *2005 International Conference on Control and Automation* (2005), vol. 1, IEEE, pp. 125–130. (Cited on page 9).

[159] Yu, D., and Gomm, J. Implementation of neural network predictive control to a multivariable chemical reactor. *Control Engineering Practice 11*, 11 (2003), 1315–1323. (Cited on page 4).

[160] Zabinsky, Z. B., Smith, R. L., and Kristinsdottir, B. P. Optimal estimation of univariate black-box Lipschitz functions with upper and lower error bounds. *Computers & Operations Research 30*, 10 (2003), 1539–1553. (Cited on page 15).