Escuela Superior Técnica de Ingeniería
Departamento de Ingeniería de Sistemas y Automática

Doctoral Thesis

Safe and optimal operation under uncertainty and plant-model mismatch

Victor Mirasierra Calleja

Supervised by:
Daniel Limon Marruedo
Teodoro Alamo Cantarero

Seville, 2024

Acknowledgements
Agradecimientos

Me gustaría dedicar este espacio para agradecer a todos aquellos que me han acompañado en este camino durante los últimos años. Lo que siempre parecía tan lejano, es ahora una realidad y eso es en parte gracias a vosotros.

Gracias a mi familia y en especial a mi madre Dory por apoyarme siempre, cuando salían los resultados y cuando nada parecía avanzar durante meses. También quiero agradecer a mis amigos, tengo la suerte de tener muchos y sería difícil nombrarlos a todos, así que mencionaré sólo a aquellos que más me han tenido que aguantar hablar sobre mi tesis. Muchas gracias Jesús, Dani, Ale, Juanjo, Carlos, Ignacio y Guille.

Durante mi paso por el doctorado, he tenido la suerte de coincidir con compañeros que no sólo me han hecho aprender, sino que también me han sacado muchas risas entre simulación y simulación. Muchas gracias Pepe, Pablo, José Ramón, Juan, Joaquín, José Antonio, Richard y Dani. Me gustaría incluir también a Ignacio Alvarado, un profesor al que siempre he visto como un compañero.

Tampoco me puedo olvidar de mis directores Daniel Limón y Teodoro Álamo con los que he compartido tantos años, y de mis supervisores de estancia en Turín, Fabrizio Dabbene y Martina Mammarella. Sin ellos este trabajo no sería posible.

Por último, me gustaría concluir esta sección con un mensaje de especial importancia: *Niños, tened cuidado con los baobabs.*

Victor Mirasierra,
Sevilla, 2024.

Abstract

Designing model-based control systems involves addressing uncertainty and modeling errors to ensure robust control. While this is a fundamental control problem, in recent years, the rapid growth of the field of machine learning has led to a significant rise in the development of probabilistic and robust techniques based on data sampling.

The objective of this thesis is twofold. On the one hand, improve the estimation of safe regions by means of sample-based techniques. Safe regions bound the probability of meeting the constraints and keep it above a certain threshold. On the other hand, in this thesis we propose techniques that modify the real-time optimization problem in order to calculate the optimal operation of the plant despite the precision of the available model.

Resumen

El diseño de sistemas de control basados en modelo requiere la consideración de la incertidumbre y de los errores de modelado para dotar de robustez al controlador. Si bien éste es un problema fundamental del control, en los últimos años, el crecimiento exponencial del campo del aprendizaje automático ha provocado un incremento notable en el desarrollo de técnicas probabilísticas y robustas basadas en el muestreo de datos del sistema.

El objetivo de esta tesis es doble. Por un lado, mejorar la estimación de regiones seguras mediante técnicas basadas en la extracción de muestras. Estas regiones seguras permiten acotar la probabilidad de cumplimiento de las restricciones con el fin de mantenerla siempre por encima de cierto umbral. Por otro lado, en la tesis se proponen técnicas basadas en modificadores que ajustan el problema de optimización en tiempo real para calcular el punto de funcionamiento óptimo de la planta, aun cuando el modelo disponible sea impreciso.

Contents

Notation, conventions and definitions

Vectors and matrices

$v \in \mathbb{R}^n$	—	Column vector v of dimension n.
v^T	—	Transpose of vector v.
$\mathbf{0}_n$	—	Column vector of zeros of dimension n.
$\mathbf{1}_n$	—	Column vector of ones of dimension n.
$A \in \mathbb{R}^{m \times n}$	—	Matrix A of dimension $m \times n$.
A^T	—	Transpose of matrix A.
$\mathbf{I}_n$	—	Identity matrix of dimension n.
$\|v\|_p$	—	ℓ_p-norm of v ($p \in \mathbb{R}$), i.e. $\|v\|_p \doteq \sqrt[p]{\|v_1\|^p + \|v_2\|^p + \ldots + \|v_n\|^p}$.
$\|v\|_{p^*}$	—	Dual norm of ℓ_p-norm ($p \in \mathbb{R}$), i.e. $\|v\|_{p^*} \doteq \sup_{\|z\|_p \leq 1} v^T z$, $\forall v \in \mathbb{R}^s$.
$P \succ 0$	—	P is a positive definite matrix, i.e. $x^T P x > 0$ for all $x \in \mathbb{R}^n \setminus \{\mathbf{0}\}$.
$P \succeq 0$	—	P is a positive definite matrix, i.e. $x^T P x \geq 0$ for all $x \in \mathbb{R}^n$.
$\|v\|_P$	—	Weighted euclidean norm of vector v, i.e. $\|v\|_P \doteq \sqrt{v^T P v}$.
$\begin{bmatrix} A & B \end{bmatrix}$	—	Horizontal composition of matrices A and B.[1]
$\begin{bmatrix} A \\ B \end{bmatrix}$	—	Vertical composition of matrices A and B.[1]
$\mathrm{tr}(A)$	—	Trace of matrix A, i.e. sum of the elements of the main diagonal of A.
$\det(A)$	—	Determinant of matrix A.

Sets

$\mathbb{N}_{\geq 0}$	—	Set of natural numbers including 0.
$\mathbb{N}_{>0}$	—	Set of natural numbers excluding 0.
$\mathbb{R}^n$	—	Set of real vectors of dimension n.
$\mathbb{R}^{m \times n}$	—	Set of real matrices of dimension $m \times n$.
$[N]$	—	Set of integers ranging from 1 to N.

[1] When using $\mathbf{0}, \mathbf{1}$ or $\mathbf{I}$ in a matrix composition, the dimension of these matrices may be omitted and set accordingly to make the composition possible.

$x \in \mathcal{A}$	—	Point x is contained in set $\mathcal{A}$.
$\mathcal{A} \oplus \mathcal{B}$	—	Minkowski sum of $\mathcal{A}$ and $\mathcal{B}$, i.e. $\mathcal{A} \oplus \mathcal{B} \doteq \{a + b \,\vert\, a \in A, b \in \mathcal{B}\}$.
$\mathcal{A} \ominus \mathcal{B}$	—	Pontryagin difference of $\mathcal{A}$ and $\mathcal{B}$, i.e. $\mathcal{A} \ominus B \doteq \{c \,\vert\, c + b \in \mathcal{A}, \forall b \in \mathcal{B}\}$.
$\mathcal{A} \times \mathcal{B}$	—	Cartesian product of $\mathcal{A}$ and $\mathcal{B}$, i.e. $\mathcal{A} \times B \doteq \{(a, b) \,\vert\, a \in \mathcal{A}, b \in \mathcal{B}\}$.
$\gamma\mathcal{A}$	—	Scalar multiplication of γ and $\mathcal{A}$, i.e. $\gamma\mathcal{A} \doteq \{\gamma a \,\vert\, a \in \mathcal{A}\}$.
$\mathcal{A} \subset \mathcal{B}$	—	$\mathcal{A}$ is a subset of $\mathcal{B}$.
$\mathcal{A} \subseteq \mathcal{B}$	—	Set $\mathcal{A}$ is equal or a subset of $\mathcal{B}$.
$\mathcal{A} \cup \mathcal{B}$	—	Union between sets $\mathcal{A}$ and $\mathcal{B}$.
$\mathcal{A} \cap \mathcal{B}$	—	Intersection between sets $\mathcal{A}$ and $\mathcal{B}$.
$x_{i:N}$	—	i-th smallest element of set x, which contains N scalars.
$x_{1:N}$	—	Smallest element of set x, which contains N scalars.
$x_{N:N}$	—	Largest element of set x, which contains N scalars.
$\mathbb{B}_p^s$	—	ℓ_p-norm ball of radius one in $\mathbb{R}^s$, i.e. $\mathbb{B}_p^s \doteq \{z \in \mathbb{R}^s : \|z\|_p \leq 1\}$.
$\{w^{(i)}\}_{i=1}^N$	—	Set of N samples of w, i.e. $\{w^{(1)}, w^{(2)}, \ldots, w^{(N)}\}$.

Probabilities

$\Pr_{\mathcal{W}}$	—	Probability distribution of vector $w \in \mathcal{W}$.
$\Pr_{\mathcal{W}}\{w \leq 0\}$	—	Probability of $w \leq 0$ given the probability distribution $\Pr_{\mathcal{W}}$.
$E_{\mathcal{W}}\{w\}$	—	Expected value of the random variable w subject to probability distribution $\Pr_{\mathcal{W}}$.
$E_{\mathcal{W}}\{w\|x\}$	—	Expected value of the random variable w conditioned to x.

Other

$\lfloor x \rfloor$	—	Greatest integer no larger than x.
$\lceil x \rceil$	—	Smallest integer no smaller than x.
$B(k; N, \varepsilon)$	—	Binomial cumulative distribution, i.e. $B(k; N, \varepsilon) \doteq \sum_{i=0}^{k} \binom{N}{i} \varepsilon^i (1 - \varepsilon)^{N-i}$.
$I^g(x)$	—	Indicator function of constraint $g(x) \leq 0$, i.e. $I^g(x) = 1$ if $g(x) \leq 0$, and $I^g(x) = 0$ otherwise.
$\square$	—	*Quod erat demonstrandum*, symbolizes the end of a proof.
$\mathrm{Cheb}_p(\mathcal{A})$	—	Chebyshev center of a given set $\mathcal{A}$ with respect to the norm $\|\cdot\|_p$,

i.e. the center of the largest ℓ_p-norm ball inscribed in $\mathcal{A}$.

Abbreviations

i.e.	—	*Id est*, meaning 'that is'.
e.g.	—	Example given.
s.t.	—	Subject to.
i.i.d.	—	Independent and identically distributed.
iff	—	If and only if.
CCS	—	Chance Constrained Set.
DRTO	—	Dynamic Real-Time Optimization.
DT	—	Digital Twin.
FPS	—	Feasible Parameter Set.
KKT	—	Karush Kuhn Tucker.
MA	—	Modifier-Adaptation.
MPC	—	Model Predictive Controller.
NCO	—	Necessary Conditions of Optimality
OLMA	—	One-Layer Modifier-Adaptation.
P-MA	—	Periodic Modifier-Adaptation.
PBPS	—	Pack-Based Probabilistic Scaling.
PS	—	Probabilistic Scaling.
ROL	—	Robust One-Layer.
ROLMA	—	Robust One-Layer Modifier-Adaptation.
RTO	—	Real-Time Optimization.
SAS	—	Simple Approximating Set.
SSTO	—	Steady-State Target Optimization.
STTO	—	Steady-Trajectory Target Optimization.
TI	—	Tight Immersion.
VC dimension	—	Vapnik-Chervonenkis dimension.

Chapter 1

Introduction

1.1 Motivation and objectives

The main motivation of this dissertation is to improve the automatic control of systems in which the available model differs from the real behaviour of the system. To this end, this thesis deals with two main problems: the identification of safe regions through a novel methodology referred to as probabilistic scaling, and the proper real-time optimal operation of systems under plant-model mismatch using modifier-adaptation techniques.

Automatic control design requires models which contain not only the physical behaviour of the system, but also the design specifications (constraints) that are meaningful for the user. All this information is then used in the so-called control problem, which ultimate goal is to find a control law, i.e. a sequence of inputs such that, when applied to the system, its constraints are always met and a cost function is minimized. In practical applications, control problems face several challenges, which often results into the impossibility of achieving true optimal control. These challenges include model complexity, uncertainty and plant-model mismatch.

Models are mathematical entities that are used to represent the behaviour and constraints of a system. The identification of models can be classified into three categories. Depending of the source of the information used to build it, we can distinguish into data-based, physics-based and hybrid models.

Data-based models rely only on empirical observations of the system and statistical relationships derived from them. Given past measurements from the system, data-based models are built using techniques such as machine learning and statistical analysis [1, 2, 3]. Due to the empirical nature of the information used to build the model, these models suffer from measurement errors and their accuracy dwindles if the number of samples is too small or when the system exits the sampled zone. One example of a data-based model would be that of stock market models, in which forecasts of a company value are made based on past values (see e.g. [4, 5]).

On the other side, physics-based models do not use information from samples, but from the fundamental physics principles that govern the behaviour of the system. These models are based on the assumption that the real system follows a theoretical behaviour. However, this often implies that the system meets a series of assumptions, which it often does not. Examples of this type of models can be seen e.g. in fluid mechanics [6].

Lastly, hybrid models are a mixture of the previous ones. They use sample information to tune the parameters of physics-based models and therefore they are able to improve their accuracy in the sampled regions. These models can be very intricate and require both deep knowledge of the system and representative samples of the operating region. Despite all of this, they still can not always capture the real behaviour of the system, mostly because of unmodelled variability. Some examples of hybrid models, also known as greybox models can be found in [7, 8].

This thesis tackles some of the issues that arise when dealing with real world systems. The results that stem from this work can be applied to any of the aforementioned models and improve both the robustness and the performance of the automatic control. In order to do it, this dissertation mainly deepens into two methodologies: probabilistic scaling and modifier-adaptation.

In probabilistic scaling, the aim is to calculate probabilistic safe regions of user-defined complexity such that the points inside them satisfy random constraints with some given probability. This is accomplished by scaling some user-defined geometry until it meets the given probabilistic guarantees. Probabilistic scaling is a novel framework presented in this thesis. Along this work, we will study some of its applications and compare it with state-of-the-art alternatives.

Besides, modifier-adaptation techniques [9, 10] focus more explicitly in the optimal operation of systems for which an accurate model is not available or when the complexity of the accurate model makes it ill-suited for control purposes. At its core, modifier-adaptation techniques use gradient information to update the control model with zeroth and first term modifiers so that, upon convergence, the system reaches a steady behaviour which optimizes the given cost function. In this work we propose new control schemes that use modifier-adaptation and discuss their benefits over the state-of-the-art alternatives.

As we will see throughout the document, both probabilistic scaling and modifier-adaptation tackle the uncertainty problem, but they do it at two different levels. In the first part of this thesis, we focus on the calculation of safe regions, which are relevant for many applications dealing with uncertain data. Then, in the second part of this work, the focus will be put on automatic control, where modifier-adaptation techniques will help the controlled systems reach their steady optimal behaviours.

1.2 Preliminaries: Dealing with uncertainty in automatic control

In the field of control systems engineering, the presence of uncertainty in real-world systems has always constituted a fundamental challenge that can hurdle the performance of practical applications. As we strive to design and implement control strategies for complex systems, it is necessary to acknowledge and effectively manage the uncertainty. Neglecting the uncertainty may have catastrophic consequences, ranging from poor performance to stability issues and even major failures that cause the collapse of the system. For example, in the case of autonomous vehicles, uncertainty in the signal detection system may result in an incorrect speed signal detection. This may turn the vehicle slower or faster than intended, increasing the risk of accident. In medical applications, a neglectful management of the uncertainty may result in an overconfident misleading diagnosis.

Uncertainty in control systems manifests in many forms, including deterioration of the systems, external disturbances, sensor noise, parameter variation and communication delays. The dynamic nature of many systems is usually coupled with an unpredictable environment, which further amplifies their uncertain behaviour.

As deterministic models proved to be inadequate to capture the unpredictable nature of real-world systems, robust control arised to shift the paradigm (see [11, 12, 13]). The emergence of robust control dates back to the early 1970s, when due to failure of optimal control theory to tolerate regular differences between the design models and the real system's behaviour, the focus of research shifted from optimality to robustness [14]. Ever since, robustness of the control schemes has became a mainstay of control schemes.

Unlike traditional control methodologies that rely on deterministic system models, the robust frameworks acknowledge the limitations of the models and embrace the spectrum of potential uncertainties. At its core, robust control studies the uncertainty and bounds it to a set of possible values. Then, it adopts a worst-case scenario approach and guarantees that the designed controllers maintain stability and acceptable performance under the most challenging and unfavourable conditions. As a consequence, the success of robust control is linked to the accuracy of the characterization of the uncertainty and the assumption that it remains bounded in a closed and compact region. Moreover, designing the controllers for the worst-case scenario has some drawbacks, namely the control can be very conservative and therefore reduce the performance in cases with little to none uncertainty, and unbounded uncertainties would render the robust design impossible.

Stochastic control [15, 16, 17, 18, 19] represents another shift in the paradigm of control systems engineering. It considers the probabilistic nature of the uncertainty and the fact that worst-case scenario is generally very uncommon, and designs the control scheme in order to keep the probability of failure under a

certain threshold. The statistical framework used by stochastic control enables the design of controllers that are founded on the likelihood of different scenarios and not solely on worst-case assumptions. In real-world systems where the uncertainties exhibit an inherent randomness, stochastic control emerges as a valuable paradigm that complements and extends the capabilities of robust control, enhancing the performance of control strategies.

In the following section, we introduce the notion of chance constraints first proposed in [20], which plays a major role not only in stochastic control, but also in a variety of fields such as transportation or manufacturing engineering [21, 22].

1.2.1 Preliminaries: Chance constraints and safe regions

Consider the uncertainty $w \in \mathbb{R}^{n_w}$, which represents one of the admissible uncertainty realizations of a random vector subject to the probability distribution $\Pr_{\mathcal{W}}$ and (possibly unbounded) support $\mathcal{W}$. Consider also the decision variable $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$, where Θ represents the admissible region, and let the design specifications of a problem be described as a set of n_ℓ uncertain inequalities of the form:

$$\begin{bmatrix} g_1(\theta, w) \\ \vdots \\ g_{n_\ell}(\theta, w) \end{bmatrix} \leq \mathbf{0} \iff g(\theta, w) \leq 0, \tag{1.1}$$

where $g_\ell : \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_w} \to \mathbb{R}$, for $\ell = 1, \ldots, n_\ell$ and $g : \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_w} \to \mathbb{R}$.

In contrast to robust frameworks, probabilistic settings allow the violation of the constraints if the probability of violation is kept below a certain (usually small) threshold. This choice helps to keep the problem feasible and not overly conservative. The relaxed design specifications receive the name of chance constraints or probabilistic constraints and can be expressed as

$$\Pr_{\mathcal{W}}\{g(\theta, w) \leq 0\} \geq 1 - \varepsilon,$$

where $\varepsilon \in (0, 1)$ is the probabilistic design choice. The previous definition let us introduce the notion of chance-constrained set of probability ε.

Chance-constrained set of probability ε (ε-CCS)

Given $\varepsilon \in (0, 1)$, the chance-constrained set of probability ε (also known as ε-CCS) is defined as the set of all possible values of the design parameter θ for which the chance constraint is satisfied, i.e.

$$\mathbb{X}_\varepsilon \doteq \{\theta \in \Theta : \Pr_{\mathcal{W}}\{g(\theta, w) > 0\} \leq \varepsilon\}. \tag{1.2}$$

Note that the ε-CCS constitutes a probabilistic safe region, and it can be seen as an extension of the robust safe region. By choosing $\varepsilon = 0$, we have that $\mathbb{X}_0$ is the set of all possible values of the design paremeter $\theta \in \Theta$ that meet the design

specifications (1.1) for any value of the uncertainty $w \in \mathcal{W}$, i.e. the robust safe region.

Chance constraints can be split into two categories, namely joint and individual chance constraints:

Joint vs. individual chance constraints

The chance constraint $\theta \in \mathbb{X}_\varepsilon$, with $\mathbb{X}_\varepsilon$ defined in (1.2), describes a joint chance constraint. That is, it requires that the probability of satisfying all the inequality constraints $g_\ell(\theta, w) \leq 0$ (with $\ell = 1, \ldots, n_\ell$) is guaranteed to be no smaller than the probabilistic level $1 - \varepsilon$. We remark that the joint chance constraint is notably harder to impose than individual chance constraints, which take the form

$$\theta \in \mathbb{X}_{\varepsilon_\ell}^{(\ell)} \doteq \{\theta \in \Theta : \Pr_{\mathcal{W}} \{g_\ell(\theta, w) > 0\} \leq \varepsilon_\ell\},$$

with $\varepsilon_\ell \in (0, 1)$ being the probabilistic level of the individual chance constraint ℓ. A discussion on the differences and implications of joint and individual chance constraints may be found in several papers, see for instance [23, 24] and references therein. Note that a well-known conservative approximation to the joint chance-constrained set is to use the intersection of multiple individual chance constraints.

Relaxing the constraints and taking probabilities into account make stochastic schemes less conservative than their robust counterpart. Moreover, they make possible to deal with infinite support uncertainties. In return, their design process is much more intricate for two main reasons: First, it is highly difficult to check whether solutions of chance-constrained problems are feasible, and second, chance constraints usually involve non convexity.

Optimization problems that involve chance constraints are known as chance-constrained optimization (CCO). Given the decision variable $\theta \in \Theta$ and the cost function $J : \mathbb{R}^{n_\theta} \to \mathbb{R}$, the CCO can be expressed with the general formulation:

$$\min_{\theta \in \mathbb{X}_\varepsilon} \quad J(\theta). \tag{1.3}$$

According to the accuracy and the source of the approximations, the authors of [23] classify the computation of the ε-CCS and the solution of the subsequent CCO in three different classes:

Exact techniques

In a handful of cases and given that the uncertainty distribution is known, the ε-CCS is convex and hence the CCO problem may be solved in a manageable time. This is the case, for instance, of individual chance constraints with w being gaussian [25]. Other important examples of convexity of the set $\mathbb{X}_\varepsilon$ involve log-concave distribution [26, 27]. General sufficient conditions on the convexity of

chance constraints may be found in [28, 29, 30, 31]. Although all these cases may work well for very specific cases, they require previous knowledge on the distribution of the uncertainty (which in real applications is unknown), and hardly extend to the joint chance constraints considered in this work.

All these previously cited references deal with continuous distributions. A different line of research concentrates instead on discrete distributions, which arise frequently in applications, either directly, or as empirical approximations of the underlying distribution (see, for example, [26, 32]). For this particular case, exact results based on the concept of p-efficiency points [33] or dual methods [34] have been proposed.

As pointed out in [23], the computation of the ε-CCS is usually extremely difficult, since the evaluation of the probability $\Pr_{\mathcal{W}}\{g(\theta, w) \leq \mathbf{0}\}$ amounts to the solution of a multivariate integral, which is an NP-hard problem [35]. These limitations on the exact computation of the ε-CCS motivate the search of good approximations.

Deterministic approximations

A second class of approaches consist in finding deterministic conditions that allow to construct a convex inner approximation of the probabilistic set $\mathbb{X}_\varepsilon$. The classical solution consists in the applications of Chebyshev-like inequalities, see e.g. [36, 37]. More recent techniques, which are proved particularly promising, involve robust optimization [38], as the convex Bernstein-based approximations introduced in [39, 40]. A particular interesting convex relaxation involves the so-called Conditional Value at Risk (CVaR), see [41] and references therein. Finally, we point out some recent techniques based on Genz' code for gaussian probabilities of rectangles [42], or on polynomial moments relaxations [43, 44].

Specific solutions have been proposed for the case of discrete distributions, see the recent survey [45]. In particular, we point out the recent works proposing a boolean reformulation of the feasible set of individual and joint chance constraints (see [46, 47]).

Nonetheless, it should be remarked that these techniques usually suffer from conservatism and computational complexity issues, especially in the case of joint chance constraints.

Sample-based techniques

The third approach to approximate the CCO is based on random sampling of the uncertain parameters. This approach has gained popularity in recent years due to its versatility, see e.g. [48, 49, 50, 51] and references therein. Sampling-based techniques are characterized by the use of a finite number N of i.i.d. samples of the uncertainty $\mathbf{w}_N = \left\{w^{(1)}, w^{(2)}, \ldots, w^{(N)}\right\}$, each of them drawn according to the probability distribution $\Pr_{\mathcal{W}}$. With each sample $w^{(i)}, i \in [N]$, we can associate

the following sample safe set

$$\Phi_0^g(w^{(i)}) = \{\theta \in \Theta : g(\theta, w^{(i)}) \leq 0\}, \tag{1.4}$$

sometimes referred to as scenario, since it represents an observed instance of the uncertain constraint.

The scenario approach [52] considers the CCO problem (1.3) and approximates its solution through the following scenario problem

$$\begin{aligned} \theta_{sc}^* = \arg\min_{\theta} \quad & J(\theta) \\ \text{s.t.} \quad & \theta \in \Phi_0^g(w^{(i)}), \quad \forall i \in [N]. \end{aligned} \tag{1.5}$$

We note that, if the function $J(\theta)$ is convex and constraint g is linear, then problem (3.7) becomes a linearly constrained convex program, for which very efficient solution approaches exist. Under some technical assumptions (feasibility of the problem and non-degeneracy), a fundamental result [52, 53, 54, 55] provides a probabilistic certification of the constraint satisfaction for the solution to the scenario problem. In particular, it is shown that

$$\Pr_{\mathcal{W}^N} \{w \in \mathbf{w}_N : \Pr\{g(\theta_{sc}^*, w) > 0\} > \varepsilon\} \leq B(n_\theta - 1; N, \varepsilon), \tag{1.6}$$

where $B(n_\theta - 1; N, \varepsilon) \doteq \sum_{i=0}^{n_\theta - 1} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i}$ and $\Pr\{g(\theta_{sc}^*, w) > 0\}$ constitutes the probability of violation of the chance constraint. Given a random multisample $\mathbf{w}_N \in \mathcal{W}^N$, equation (1.6) offers an upper bound to the probability that at least one of them does not meet the chance constraint.

A few observations are at hand regarding the scenario approach and its relationship with the approximation of the ε-CCS. First, if we define the multisample safe set as

$$\Phi_0^g(\mathbf{w}_N) \doteq \bigcap_{i=1}^{N} \Phi_0^g(w^{(i)}), \tag{1.7}$$

we see that the scenario approach consists in approximating the constraint $\theta \in \mathbb{X}_\varepsilon$ in (1.3) with its sampled version $\theta \in \Phi_0^g(\mathbf{w}_N)$. On the other hand, it should be remarked that the scenario approach cannot be used to derive any guarantee on the existing relationship between $\Phi_0^g(\mathbf{w}_N)$ and $\mathbb{X}_\varepsilon$.

Indeed, the nice probabilistic property in (1.6) holds only for the optimal value of the scenario problem θ_{sc}^*. This is a fundamental point, since the scenario results build on the so-called support constraints, which are defined only for the optimum θ_{sc}^*.

To conclude, sample-based approximations of the ε-CCS may not be limited by the random distribution of the uncertainty and have therefore been a popular option to solve chance-constrained optimization problems. However, the fixed shape of the approximations combined with the increasing number of samples required for higher dimensional problems makes them unsuitable for many applications. One of the objectives of this dissertation is to propose a sample-based approach

to the problem of approximating probabilistic safe regions, which also mitigates the limitations of the state-of-the-art alternatives.

In the next section, we introduce the second part of this thesis. It encompasses the economic control problem and the hierarchical approach to control systems to their real-time optimal operation. In order to converge to the real optimal operation, we will make use of the so-called modifier-adaptation schemes, which aim to remove the impact of the plant-model mismatch on the steady operation.

1.2.2 Preliminaries: Real-time optimization and modifier-adaptation

In the field of automatic control, the pursuit of optimal operation of a dynamic system is a longstanding challenge, which has received much attention because of its economic impact. In this section, we go through the optimal economic control problem and motivate the use of the two-layer strategy to approximate it in practical applications. Furthermore, modifier-adaptation strategies to deal with the uncertainty on a real-time optimization (RTO) level will be introduced and open lines will be discussed.

Consider the following discrete system:

$$x_{k+1} = f_{p,k}(x_k, u_k),$$

where $x_k \in \mathbb{R}^{n_x}$ and $u_k \in \mathbb{R}^{n_u}$ are respectively the states and inputs of the system at time k, and $f_{p,k} : \mathbb{R}^{n_x \times n_u} \to \mathbb{R}^{n_x}$ represents the dynamics of the real system at time k. Each step in k represents t_T seconds.

At any time k, the states and inputs of system can be subject to (possibly nonlinear) constraints of the form:

$$g_k(x_k, u_k) \leq 0.$$

Let the function $\phi_k : \mathbb{R}^{n_x \times n_u}$ represent the economic cost of operating the system at time k and consider the initial state $x_0 \in \mathbb{R}^{n_x}$. Then, the optimal economic control problem calculates the infinite sequence of inputs that, when applied to the system, minimizes the economic cost over time and can be formulated as follows:

$$\begin{aligned} \min_{\mathbf{u}_\infty} \quad & \sum_{k=0}^{\infty} \phi_k(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f_{p,k}(x_k, u_k), \quad \text{for all } k = 0, 1, \ldots, \infty \\ & g_k(x_k, u_k) \leq 0, \quad \text{for all } k = 0, 1, \ldots, \infty. \end{aligned} \tag{1.8}$$

The previous formulation is rarely implemented in real applications due to the lack of knowledge of the real system's dynamics (i.e. $f_{p,k}$) and the complexity of dealing with an infinite number of constraints and decision variables. Therefore, simplifications needs to be made.

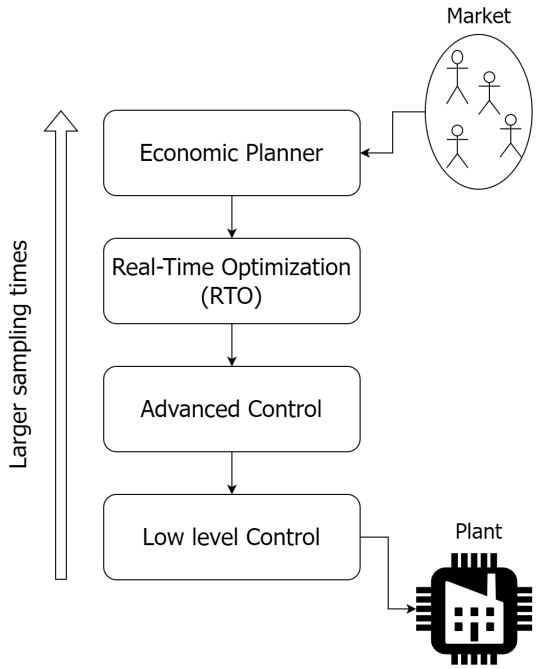


Figure 1.1: Control diagram.

One of the most studied approaches to the optimal economic control problem is the hierarchical scheme [56]. The hierarchical control scheme (Figure 1.1) represents a structured approach to managing complex systems. In this scheme, control responsibilities are distributed across multiple entities represented by the different layers. These layers make use of different time scales and models, leading to some interesting properties. Traditionally, upper layers use more complex models with larger time scales, which makes it possible to solve global intricate poblems. Whereas lower layers use simple models (many times linear models) and require short time scales. This combination is required to react to local disturbances upon operating the plant.

At the topmost level of the hierarchical structure lies the economic planner, which calculates the economic parameters and optimal production according to the state of the market. Then, this information is passed to the real-time optimization, which uses it to calculate the optimal steady operation of the plant. Once this steady operation is calculated, the advanced control is in charge of taking the plant from its current state to the steady reference. Finally, at the lowest level, low level control mechanisms govern individual actuators, implementing precise manipulation to execute the control sequence given by the advanced control.

Two-layer control scheme

The combination of the real-time optimization and the advanced control constitutes what we call the two-layer control scheme.

Modifier-adaptation

Being at the upper part of the hierarchical scheme, real-time optimization plays a significant role in the economic cost associated to the operation of the plant. Therefore, the impact of plant-model mismatch in this layer should not be understated.

In 1979, Roberts presented the integrated system optimization and parameter estimation algorithm (ISOPE) [57], which aimed at optimizing the steady-state under plant-model mismatch. This approach first modified the model parameters and then it solved another optimization problem in which the cost function was modified with gradient-based modifiers. This modification aimed to align the first-order necessary conditions for optimality of the model-based problem with those of the optimal problem. Building upon this foundation, in 2009 Marchetti et al. formally articulated and refined this methodology, coining their method as modifier-adaptation [9]. In this work, the authors discarded the modification of the model parameters and introduced first order modifiers not only to the cost function but also to the constraints, thereby matching the necessary optimality conditions. Like the work from Roberts, the modifier-adaptation methodology also required knowledge about the real plant gradients. To this end, many reformulations of modifier-adaptation have been proposed.

Dual modifier-adaptation [58] added an extra (dual) constraint that guarantees the excitation of the system is enough to calculate empirically the directional gradients required to set the modifiers. The iterative gradient-modification optimization (IGMO) [59] takes a similar approach to the dual modifier-adaptation, but takes into account the noise of the gradient estimation and proposes the use of quadratic approximations to make the estimation smoother. Other example of modifier-adaptation reformulation is the nested modifier-adaptation (NMA) presented in [60]. Unlike the previous approaches, NMA does not rely on the estimation of gradients of the real plant, and presents a nested architecture with a gradient-free optimization algorithm to update the modifiers. This algorithm iterates with the modifiers until a point that meets the necessary conditions for optimality is reached.

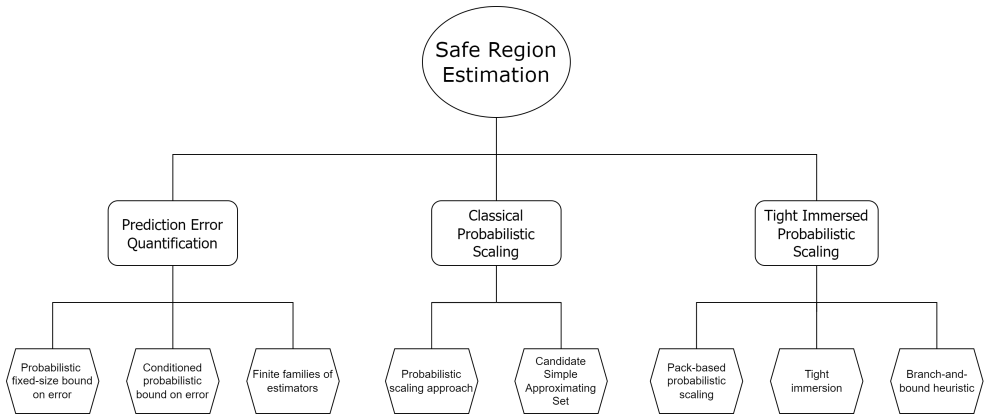
In future chapters of this dissertation, we will focus on the real-time optimization and the advanced control layers and study new modifier-adaptation reformulations that allow us to achieve optimal steady performance.

In the following section, we provide a summary of the contents of this thesis.

1.3 Thesis overview

As mentioned at the begining of this chapter, the contributions of this thesis can be classified into two main categories, which corresponds to the two-part division made for the remainder of this dissertation: safe region estimation and real-time optimization under plant-model mismatch.

Part I of this work focuses on the estimation of safe regions, i.e. sets in which the probability of meeting random constraints is above a given value. This part includes the following chapters:

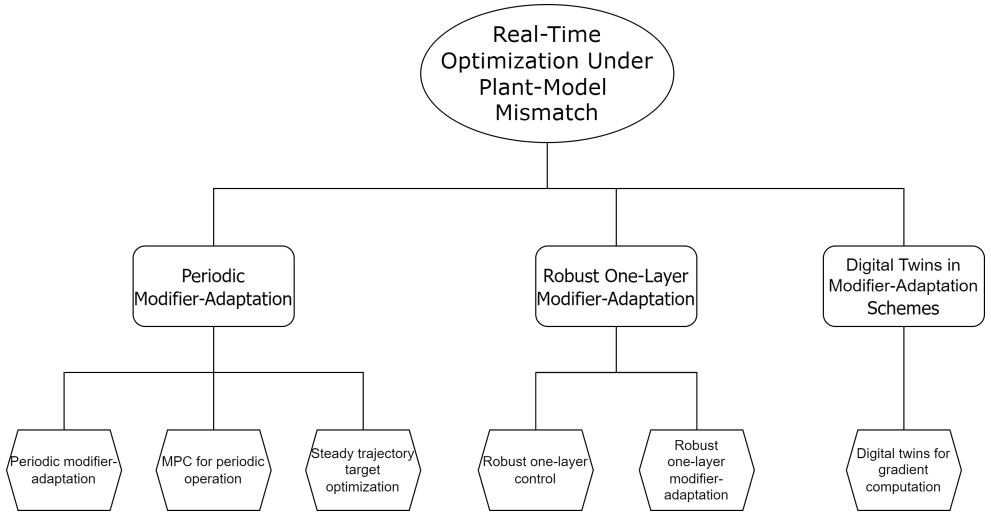


- **Chapter 2** addresses the probabilistic error quantification of a general class of prediction methods. Given any predictor and several (representative) samples, this chapter shows how to obtain a probabilistic upper bound on the absolute value of the prediction error. The required number of samples is independent of the complexity of the prediction model. The performance of the proposed approach is tested in an interval prediction problem.

- **Chapter 3** presents the regular formulation of probabilistic scaling, which is used to approximate safe regions. Probabilistic scaling takes a simple set and scales it around a point so that the scaled set meets some desired probabilistic guarantees. In this chapter, two families of simple sets are presented. Then, the proposed approach is tested against a probabilistic set membership estimation problem.

- **Chapter 4** introduces a measure of tightening of the approximations of the safe region and extends the regular formulation of probabilistic scaling with a pack-based approach that tightens the approximation. This chapter also presents a branch-and-bound heuristic to calculate the approximating set. The performance of this new formulation is tested against an academic example that illustrates the limitations of the regular formulation in some applications.

Part II contains all the contributions related to modifier-adaptation schemes in the real-time optimization layer. This encompasses the following chapters:



- In **Chapter 5**, a periodic modifier-adaptation formulation of the dynamic real-time optimization is proposed. This formulation updates the problem with affine modifiers so that, upon convergence, its optimal solution matches the optimal steady periodic trajectory. The full control scheme is detailed and tested against a periodic version of the quadruple tank benchmark.
- **Chapter 6** presents a robust one-layer control scheme which can converge to the optimal steady-state of the system using affine modifiers. The resulting integrated scheme, called ROLMA, is robust and able to reduce the economic control problem using to a single quadratic programming. Under some assumptions, both the robustness and convergence to the optimal steady operation of the system are proven and the performance of ROLMA in the quadruple tank benchmark is shown.
- **Chapter 7** proposes the use of digital twins in modifier-adaptation formulations. The digital-twin framework can identify in real-time the gradients of a system, solving along the way the main challenges of modifier-adaptation schemes.
- Finally, **Chapter 8** recapitulates the contributions of this thesis and review future research lines that could expand upon the foundations of this work.

1.4 Main results

This section showcases the main contributions of this thesis in the fields of safe region estimation and modifier-adaptation. These contributions will be distributed throughout the thesis following the structure outlined in the previous section.

- Two novel data-based algorithms tailored to bounding the prediction error. These algorithms are predictor agnostic, which means that they can be used upon any given predictor to derive bounds on their prediction errors.

- An extension to one of the algorithms proposed above that minimizes the risk of choosing a bad predictor.
- A measure of tightening of the approximations of the chance-constrained set.
- A simple data-based approximation of the chance-constrained set that is able to scale simple sets until they meet the desired probabilistic guarantees.
- An extension to the aforementioned approximation, which divides the samples in packs in order to improve some weakness of the previous approach.
- A branch-and-bound algorithm that efficiently computes the proposed pack-based approximation of the chance-constrained set.
- A periodic modifier-adaptation formulation that can achieve steady optimal periodic performance as opposed to the optimal steady-state performance achieved by the state-of-the-art modifier-adaptation schemes.
- A robust one-layer control scheme that can converge to the optimal steady-state using affine modifiers. The control problem associated with this controller can boil down to a single quadratic programming.
- A discussion on the benefits that the digital twin framework can bring to modifier-adaptation schemes.

1.5 Publications

Most of the results presented throughout this thesis have been published in several international journals and congresses, some of which are currently under review.

The probabilistic scaling approach to the estimation of chance constrained sets, along with its application to quantify prediction errors presented in part I of this thesis are covered in the following papers:

- T. Alamo, V. Mirasierra, F. Dabbene, and M. Lorenzen, "Safe approximations of chance constrained sets by probabilistic scaling," in 2019 18th European Control Conference (ECC), pp. 1380–1385, IEEE, 2019.
- M. Mammarella, V. Mirasierra, M. Lorenzen, T. Alamo, and F. Dabbene, "Chance-constrained sets approximation: A probabilistic scaling approach," Automatica, vol. 137, p. 110108, 2022.
- V. Mirasierra, M. Mammarella, F. Dabbene, and T. Alamo, "Prediction error quantification through probabilistic scaling," IEEE Control Systems Letters, vol. 6, pp. 1118–1123, 2021.

The tight immersion as well as the heuristic associated with it, both presented in chapter 4, have not been yet submitted to neither a journal nor a congress and they therefore constitute a completely novel result of this thesis.

The one-layer and periodic modifier-adaptation schemes proposed in part II of this dissertation are covered in the following publications:

- J. D. Vergara-Dietrich, V. Mirasierra, A. Ferramosca, J. E. Normey-Rico, and D. Limon, "A modifier-adaptation approach to the one-layer economic MPC," IFAC-PapersOnLine, vol. 53, no. 2, pp. 6957–6962, 2020.
- V. Mirasierra, J. D. Vergara-Dietrich, and D. Limon, "Real-time optimization of periodic systems: A modifier-adaptation approach," IFACPapersOnLine, vol. 53, no. 2, pp. 1690–1695, 2020.
- V. Mirasierra and D. Limon, "Modifier-adaptation for real-time optimal periodic operation," arXiv preprint arXiv:2309.09680, 2023.

The robust one-layer modifier-adaptation proposed in chapter 6 and the proposal of digital twins to calculate the plant information required to apply modifier-adaptation schemes presented in chapter 7 have not been yet submitted to neither a journal nor a congress.

Part I

Safe Region Estimation

Chapter 2

Prediction Error Quantification

2.1 Introduction and Problem Formulation

Quantifying the error related to the process of approximating a set of given data with a prescribed prediction method represents a fundamental requirement, which has given rise to an entire research area known as uncertainty quantification, see e.g. [61, 62] and references therein.

Motivated by this necessity, methods for directly constructing predictive models with prescribed robustness guarantees have recently gained popularity. For instance, [63] presents several methods based on interval analysis to construct intervals which are guaranteed to contain the true value, under the assumption of deterministically bounded noise. Similarly, data-based approaches exploiting the availability of random samples, providing probabilistic guarantees, are being developed. These methods extend classical quantile regression [64]. In particular, we point out the probabilistic interval predictions proposed in [65, 66].

All these methods require to design (or re-design) the estimator using a specific ad-hoc model. However, this approach may not result practical when data-analysts have already constructed a model exploiting a "preferred" technique (e.g. one based on deep learning or support vector machines) and they want to assess, before deployment, the actual uncertainty of their model.

For this reason, research on post-processing methods for quantifying the uncertainty of a given predictor has grown in popularity. In these methods, no new methodology is proposed to construct a regression model, since the model (or a family of candidate ones) is considered given. This philosophy is exactly the one pursued in uncertainty quantification methods, see e.g. the recent approaches based on polynomial chaos [61], or the conformal predictors [67]. These methods typically use additional validation (or calibration) data to determine precise levels of confidence in new predictions [68].

In this chapter, we move a step further in this direction and present sample-based techniques for assessing the corresponding error in a computationally efficient way. Indeed, this approach extends recent results on probabilistic scaling,

e.g. [69, 70], and only requires a number of randomized samples independent of the complexity of the prediction model (i.e. the dimension of the regressor).

In particular, we consider that given a predictor variable $x \in \mathbb{R}^{n_x}$, an estimation $\hat{y}$ for the response variable $y \in \mathbb{R}$ is provided by operator $T : \mathbb{R}^{n_x} \to \mathbb{R}$. That is,

$$\hat{y} = T(x).$$

We assume that the operator T is a given predictive model that has been designed by means of any modelling methodology (first principles, linear regression, SVM regression, neural network, etc.).

We want to provide a probabilistic bound on the prediction error. More formally, we consider the random vector $w = (x, y) \in \mathbb{R}^{n_x} \times \mathbb{R} \subseteq \mathcal{W}$, with stationary probability distribution $\Pr_{\mathcal{W}}$, and we aim at constructing a function $\rho : \mathbb{R}^{n_x} \to \mathbb{R}$ such that, with probability no smaller than $(1 - \delta)$,

$$\Pr_{\mathcal{W}}\{|y - T(x)| \leq \rho(x)\} \geq 1 - \varepsilon.$$

The method relies on the possibility of accessing random observations couples $w = (x, y)$. These observations must be new data not used to construct $T(\cdot)$. We show that the sample complexity of the proposed techniques (i.e. the number of observations required) does not depend on the chosen regression model but only on the desired probabilistic levels.

The remainder of the chapter is structured as follows. In Section 2.2 we propose a first simple result, which allows to obtain an initial probabilistic bound on the prediction error via probabilistic maximization, given a predictive model. The obtained bound, which can be computed by means of a simple algorithm, is independent of the given regressor. Section 2.3, focuses on including those situations in which the expected size of the error does depend on the regressor, and proposes a probabilistic bound conditioned to the regressor. In Section 2.4, we show how kernel methods can be applied to obtain a predictor and an estimator of the prediction error variance. In Section 2.5, the approach of bounding the prediction error conditioned to the regressor is extended to the case where a "family" of candidates estimators is considered and both of the proposed approaches are illustrated by means of a running numerical example. Finally, Section 6.7 recapitulates the contributions of the chapter and draws some conclusions.

This chapter is based on the published paper [71].

2.2 Uncertainty quantification using probabilistic maximization

In this section we present an initial probabilistic bound for the prediction error $e = |y - T(x)|$ based on probabilistic maximization.

Suppose that we draw N independent and identically distributed (i.i.d.) samples $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ according to distribution $\Pr_{\mathcal{W}}$, and we denote by

$$e^{(i)} \doteq |y^{(i)} - T(x^{(i)})|, \quad i \in [N]$$

the absolute value of the corresponding prediction errors. A well established result [72] shows that the largest value in the sequence $\{e^{(i)}\}_{i=1}^{N}$, i.e. $e_{N:N}$, provides a probabilistic upper bound on the random variable $e = |y - T(x)|$. Formally, given $\varepsilon \in (0,1)$ and $\delta \in (0,1)$, [72, Theorem 1] states that if

$$N \geq \frac{1}{\varepsilon} \log(\frac{1}{\delta}) \tag{2.1}$$

then, with probability no smaller than $1 - \delta$,

$$\Pr_{\mathcal{W}} \{e > e_{N:N}\} \leq \varepsilon.$$

It is immediate to observe that this result provides a first simple probabilistic scheme for uncertainty quantification: If N i.i.d. samples $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ are drawn according to $\Pr_{\mathcal{W}}$, with N satisfying (2.1), then with probability at least $1 - \delta$,

$$\Pr_{\mathcal{W}} \{|y - T(x)| \leq e_{N:N}\} \geq 1 - \varepsilon.$$

We notice that the required sample complexity (i.e. the number of samples N) depends only on ε and δ. Moreover, no specific assumptions are required on $T(x)$ or $\Pr_{\mathcal{W}}$.

However, we also note that this scheme may provide extremely conservative results, especially if the support of the random variable $e = |y - T(x)|$ is not finite and N is large. In fact, suppose that $y - T(x)$ is a zero mean gaussian random variable. Then, the probabilistic upper bound obtained from $e_{N:N}$ will be too conservative if one of the samples $e^{(i)} = |y^{(i)} - T(x^{(i)})|$ departs considerably from zero, which occurs with a probability that increases with N. We conclude that only relying on the largest observed value of $|y - T(x)|$ hinders the computation of sharp probabilistic bounds, especially for small values of ε and δ, leading to a large number of samples N.

In order to circumvent this issue, we resort to the following result [69, Property 3], which states how to obtain a probabilistic upper bound of a random scalar variable by means of the notion of generalized max (see Definition 1).

Property 2.1. *Given probability levels $\varepsilon \in (0,1)$ and $\delta \in (0,1)$ and the discarding integer $r \geq 0$, let $N > r$ be such that*

$$B(r; N, \varepsilon) = \sum_{i=0}^{r} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i} \leq \delta. \tag{2.2}$$

Suppose that $e \in \mathcal{W} \subseteq \mathbb{R}$ is a random scalar variable with probability distribution $\Pr_{\mathcal{W}}$. Draw N i.i.d. samples $\{e^{(i)}\}_{i=1}^{N}$ from distribution $\Pr_{\mathcal{W}}$. Then, with a probability no smaller than $1 - \delta$,

$$\Pr_{\mathcal{W}}\{e > e_{N-r:N}\} \leq \varepsilon. \tag{2.3}$$

Remark 2.1 (On Property 1)**.** *This result is proved in [69] using techniques from the field of order statistics [73]. As discussed in [69], this result may be alternatively derived by applying the scenario approach with discarded constraints [53, 54]. Adaptations of this result have been used in the context of chance constrained optimization [74, 75], and stochastic model predictive control [70, 76, 77].*

2.2.1 Choice of the parameters

Several questions arise when trying to apply Property 2.1 to the probabilistic error quantification problem:
Choice of N: To choose N such that the constraint $B(r; N, \varepsilon) \leq \delta$ holds, we present the following lemma:

Lemma 2.1. *Given* $\varepsilon \in (0, 1)$ *and* $\delta \in (0, 1)$*, then in order to satisfy*

$$B(r; N, \varepsilon) \leq \delta \tag{2.4}$$

it suffices to take N *such that*

$$N \geq \frac{1}{\varepsilon}\left(r + \ln\frac{1}{\delta} + \sqrt{2r\ln\frac{1}{\delta}}\right). \tag{2.5}$$

Proof. This lemma is proved in [78, Corollary 1]. □

Thus, given r, δ, and ε, the sample size N can be obtained as the smallest integer N satisfying (2.5). Another possibility is to compute, by means of a numerical procedure, the smallest integer N satisfying $B(r; N, \varepsilon) \leq \delta$.
Choice of δ: Since $1 - \delta$ determines the probability of the satisfaction of the probabilistic constraint (2.3), it is important to choose δ sufficiently close to zero. In view of (2.5), we have that N grows logarithmically with $\frac{1}{\delta}$. This implies that significantly small values of δ (e.g. $\delta = 10^{-6}$) can be used without an excessive impact in the number of samples N.
Choice of r: If r is chosen to be too small, then the obtained probabilistic bounds might turn to be too conservative because the obtained upper bound would be determined by a reduced number of possible extreme values.

We notice from (2.5) that large values of r entails large number of required samples N. We also derive from (2.5) that $\frac{r}{N} < \varepsilon$. One reasonable choice for r with an appropriate trade off between sample complexity N and sharpness of the results is $r = \left\lfloor \frac{\varepsilon N}{2} \right\rfloor$.
Choice of ε: Parameter ε determines the size of the confidence interval in the uncertainty quantification process. In uncertainty quantification, values of ε much smaller than 0.05 are not frequent.

We now show how to obtain N in such a way that (2.2) is satisfied for the particular choice $r = \left\lfloor \frac{\varepsilon N}{2} \right\rfloor$.

Lemma 2.2. *Let $r = \lfloor \zeta \varepsilon N \rfloor$, where $\zeta \in (0,1)$, and define*

$$\kappa \doteq \left(\frac{\sqrt{\zeta} + \sqrt{2-\zeta}}{\sqrt{2}(1-\zeta)} \right)^2.$$

Then, inequality (2.4) *is satisfied for*

$$N \geq \frac{\kappa}{\varepsilon} \ln \frac{1}{\delta}. \tag{2.6}$$

In particular, the choice $\zeta = 0.5$ leads to $r = \left\lfloor \frac{\varepsilon N_\gamma}{2} \right\rfloor$ and $N_\gamma \geq \frac{7.47}{\varepsilon} \ln \frac{1}{\delta}$.

Proof. Since $r = \lfloor \zeta \varepsilon N \rfloor \leq \zeta \varepsilon N$, we obtain the sufficient condition

$$N \geq \frac{1}{\varepsilon} \left(\zeta \varepsilon N + \ln \frac{1}{\delta} + \sqrt{2\zeta \varepsilon N \ln \frac{1}{\delta}} \right) = \zeta N + \frac{1}{\varepsilon} \ln \frac{1}{\delta} + \sqrt{2\zeta N \frac{1}{\varepsilon} \ln \frac{1}{\delta}}. \tag{2.7}$$

Letting $a \doteq \sqrt{N}$ and $b \doteq \sqrt{\frac{1}{\varepsilon} \ln \frac{1}{\delta}}$, the previous expression can be rewritten as $(1-\zeta)a^2 - (\sqrt{2\zeta} b)a - b^2 \geq 0$. The largest root of this second order equation is

$$\left(\frac{\sqrt{\zeta} + \sqrt{2-\zeta}}{\sqrt{2}(1-\zeta)} \right) b.$$

Thus, (2.7) is satisfied if

$$\sqrt{N} \geq \left(\frac{\sqrt{\zeta} + \sqrt{2-\zeta}}{\sqrt{2}(1-\zeta)} \right) \sqrt{\frac{1}{\varepsilon} \ln \frac{1}{\delta}}.$$

Substituting $\zeta = 0.5$ and given that $N > 0$, we have

$$N \geq \frac{7.47}{\varepsilon} \ln \frac{1}{\delta}.$$

This proves the claim.

□

2.2.2 Probabilistic fixed-size bound on error

Property 2.1, along with the previous discussion on the choice of r, leads to Algorithm 1, which provides a simple procedure to compute a probabilistic bound on the prediction error $y - T(x)$.

Algorithm 1 Probabilistic fixed-size bound on error

1: Given a predictor $T : \mathbb{R}^{n_x} \to \mathbb{R}$, and probability levels $\varepsilon \in (0,1)$ and $\delta \in (0,1)$, choose

$$N \geq \frac{7.47}{\varepsilon} \ln \frac{1}{\delta} \quad \text{and} \quad r = \left\lfloor \frac{\varepsilon N}{2} \right\rfloor. \tag{2.8}$$

2: Draw N i.i.d. samples $\{(x^{(i)}, y^{(i)})\}_{i=1}^N$ according to $\Pr_{\mathcal{W}}$.
3: Compute $e^{(i)} = |y^{(i)} - T(x^{(i)})|$, $i \in [N]$.
4: Return $\rho = e_{N-r:N}$ as the probabilistic upper bound for $|y - T(x)|$.

The probabilistic guarantees of the upper bound generated with Algorithm 1 are provided in the next corollary.

Corollary 2.1. *The output ρ of Algorithm 1 satisfies, with probability no smaller than $1-\delta$, $\Pr_{\mathcal{W}}\{|y-T(x)|>\rho\}\leq\varepsilon$.*

Proof. From Lemma 2.1, we have that the values for N and r obtained in step 1 of Algorithm 1 guarantee that $B(r;N,\varepsilon)\leq\delta$. Thus, we conclude from Property 2.1 that $\rho=\left(|y^{(i)}-T(x^{(i)})|\right)_{N-r:N}$ satisfies, with probability no smaller than $1-\delta$, $\Pr_{\mathcal{W}}\{|y-T(x)|>\rho\}\leq\varepsilon$.

□

2.2.3 Numerical example: Algorithm 1

Consider the function

$$y=f(x,n_1,n_2)=(10+n_1)x+10\sin(4x)+5+n_2. \tag{2.9}$$

We assume that x is a random scalar with uniform distribution in $[-2.5,2.5]$ and n_1, n_2 are random scalars drawn from zero-mean gaussian distributions with variances 7 and 3, respectively. Suppose that the optimal predictor $T(x)=10x+10\sin(4x)+5$ for the random scalar $y=f(x,n_1,n_2)$ is available[1]. We fix the probabilistic levels to $\varepsilon=0.05$ and $\delta=10^{-6}$, which leads to $N=2,065$ and $r=51$ (see step 1 of Algorithm 1). We draw N i.i.d. samples $\{(x^{(i)},y^{(i)})\}_{i=1}^{N}$ and obtain $\rho=10.77$. Thus, according to Corollary 2.1, with probability no smaller than $1-\delta$, $\Pr_{\mathcal{W}}\{|y-T(x)|>\rho\}\leq\varepsilon$.

We notice that for this example it is not difficult to obtain the sharpest probabilistic bounds for $|y-T(x)|$ corresponding to a given x. It suffices to notice that given x, $y-T(x)$ is a zero-mean gaussian random variable with variance $7x^2+3$. Thus, using standard confidence interval analysis for a scalar gaussian variable, we obtain that

$$\Pr_{\mathcal{W}}\{|y-T(x)|>1.96\sqrt{7x^2+3}\}\leq 0.05.$$

Figure 2.1 shows, for a new validation set of N i.i.d. samples, the (fixed size) probabilistic bounds for y provided by Algorithm 1 (i.e. $\Pr_{\mathcal{W}}\{y\in[T(x)-\rho,T(x)+\rho]\}\geq 1-\varepsilon$), along with the exact probabilistic bounds. We notice that Algorithm 1 fails to capture the varying size of the exact probabilistic bounds. We address this issue in the next sections.

2.3 Conditioned uncertainty quantification

Despite the simplicity of Algorithm 1, the obtained upper bound does not depend on the regressor x. Clearly, this is not an issue if the error $e=|y-T(x)|$ is independent of x. However, in many situations, the expected size of the error

[1]The problem of determining predictor $T(\cdot)$ is addressed in Section 2.4.

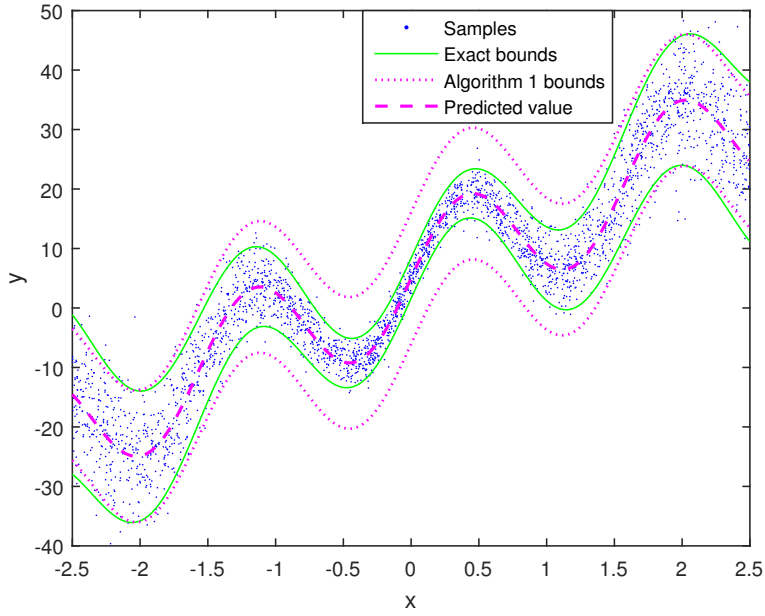


Figure 2.1: Numerical example: comparison between the probabilistic bounds obtained with Algorithm 1 with the exact ones. We notice that the size of the interval bounds provided by Algorithm 1 are independent of x.

does depend on the regressor. For example, the prediction errors are often correlated with the size of the predicted variable, which in turn is correlated with the regressor. From here, we infer that information on the expected error can often be obtained from the regressor.

Under some strong assumptions, the probability distribution of $y - T(x)$ conditioned to x, can be computed in an explicit way. This is the case, for example, when $T(x)$ is obtained by means of gaussian process regression [79] or when exponential models are employed [80]. However, we notice that, although these approaches can indeed provide estimations of the conditioned expectation

$$\sigma^2(x) = E_{\mathcal{W}}\{(y - T(x))^2 | x\},$$

the accuracy of the estimations will depend on the satisfaction of the underlying assumptions (i.e. gaussian process and exponential model, respectively) and the adequate selection of the kernels (along with their hyper-parameters) used to obtain $T(x)$. There are other possibilities to obtain conditioned error quantification, like sensitivity analysis, techniques based on Fisher information matrix, bootstrapping, etc. [61, 62].

We also mention here Parzen method [81], which serves to estimate the probability density function of a random variable. More general multivariate kernel-based generalizations are also available (see e.g. [82]). In these methods, an

estimation $\hat{\sigma}(x)$ of $\sigma(x)$ is obtained from

$$\hat{\sigma}^2(x) = \frac{\sum_{i=1}^{N_S}(y^{(i)} - T(x^{(i)}))^2\Gamma(x, x^{(i)})}{\sum_{i=1}^{N_S}\Gamma(x, x^{(i)})}, \tag{2.10}$$

where $\Gamma : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}$ is an appropriately chosen function and $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N_S}$ are i.i.d. samples drawn from $\Pr_{\mathcal{W}}$. Under non very restrictive constraints [81, 82], the provided estimation $\hat{\sigma}(x)$ converges to the actual value $\sigma(x)$ as N_S tends to infinity.

For a fixed value of x, $d = (y - T(x))^2$ is a random non-negative variable with expectation $\sigma^2(x)$. Thus, using the Markov inequality [83, 84] we obtain

$$\Pr_{\mathcal{W}}\{d \geq \xi\sigma^2(x)\} \leq \frac{1}{\xi}, \ \forall \xi > 0.$$

Thus, choosing $\xi = \frac{1}{\varepsilon}$, we obtain $\Pr_{\mathcal{W}}\left\{d \geq \frac{\sigma^2(x)}{\varepsilon}\right\} \leq \varepsilon$. Equivalently,

$$\Pr_{\mathcal{W}}\left\{|y - T(x)| \geq \frac{\sigma(x)}{\sqrt{\varepsilon}}\right\} \leq \varepsilon.$$

The obtained probabilistic upper bound suffers from the following two limitations:

- Generally $\sigma(x)$ is unknown and only a rough estimation is available (as the ones commented before).
- Markov inequality yields overly conservative results in many situations [84]. A meaningful exception to this is when the errors $y - T(x)$ are of gaussian nature, where using the chi-squared distribution [83], sharp probabilistic bounds can be obtained.

In order to avoid these limitations, we can again resort to probabilistic maximization. Suppose that an estimation $\hat{\sigma}(x)$ of $\sigma(x)$ is available. Suppose also that $\hat{\sigma}(x) > 0$, for all $x \in \mathbb{R}^{n_x}$. We could define the scaling factor γ as

$$\gamma = \frac{|y - T(x)|}{\hat{\sigma}(x)}.$$

With this definition, any probabilistic upper bound $\bar{\gamma}$ on γ would provide a probabilistic upper bound on $|y - T(x)|$. That is,

$$\Pr_{\mathcal{W}}\{\gamma > \bar{\gamma}\} \leq \varepsilon \ \Rightarrow \Pr_{\mathcal{W}}\{|y - T(x)| > \bar{\gamma}\hat{\sigma}(x)\} \leq \varepsilon.$$

2.3.1 Conditioned probabilistic bound error algorithm

In view of the previous result, we are now ready to introduce a slight modification of Algorithm 1 and obtain a novel algorithm capable of calculating a probabilistic

upper bound conditioned by the value of x. This idea is implemented in Algorithm 2.

Algorithm 2 Conditioned probabilistic bound on error

1: Given a predictor $T : \mathbb{R}^{n_x} \to \mathbb{R}$, an estimator $\hat{\sigma} : \mathbb{R}^{n_x} \to (0, \infty)$, of $\sqrt{E_{\mathcal{W}}\{(y - T(x))^2 | x\}}$, probability levels $\varepsilon \in (0, 1)$ and $\delta \in (0, 1)$, choose N and r according to (3.14).
2: Draw N i.i.d. samples $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ according to $\Pr_{\mathcal{W}}$.
3: Compute $\gamma_i = \frac{|y^{(i)} - T(x^{(i)})|}{\hat{\sigma}(x^{(i)})}$, $i \in [N]$.
4: Return $\bar{\gamma} = \gamma_{N-r:N}$, as probabilistic upper bound for $\frac{|y - T(x)|}{\hat{\sigma}(x)}$.

The following Corollary states the probabilistic guarantees of the output $\bar{\gamma}$ of Algorithm 2.

Corollary 2.2. *The output $\bar{\gamma}$ of Algorithm 2 satisfies, with probability no smaller than $1 - \delta$, that*

$$\Pr_{\mathcal{W}}\{|y - T(x)| > \bar{\gamma}\hat{\sigma}(x)\} \leq \varepsilon.$$

Proof. The proof follows the same lines as the proof of Corollary 2.1. That is, we infer from Property 2.1 and Lemma 2.1 that the proposed choice of N and r guarantees that, with probability no smaller than $1 - \delta$,

$$\Pr_{\mathcal{W}}\left\{\gamma = \frac{|y - T(x)|}{\hat{\sigma}(x)} > \bar{\gamma}\right\} \leq \varepsilon.$$

Thus, we conclude $\Pr_{\mathcal{W}}\{|y - T(x)| > \bar{\gamma}\hat{\sigma}(x)\} \leq \varepsilon$. □

Remark 2.2 (Normalization of $\hat{\sigma}(x)$)**.** *We notice that the upper bound obtained by means of Algorithm 2 provides identical results when the estimator $\hat{\sigma}(x)$ is replaced by a scaled version $\hat{\sigma}_\xi(x) = \xi\hat{\sigma}(x)$, where $\xi > 0$. Thus, multiplicative errors in the estimation of $\sigma(x)$ are corrected in an implicit way by the algorithm.*

Remark 2.3 (Difference with convex scenario approaches)**.** *Scenario approaches (see e.g. [65, 66]) obtain both the estimator and probabilistic guarantees in a single optimization problem that requires a number of samples that increases both with the dimension of the regressor used in the predictive model and the number of samples that are allowed to violate the interval predictions. Our approach can be applied to any given predictor $T(\cdot)$ and has a sample complexity that does not depend on the dimension of the regressor. This allows us to not only work with high dimensional problems, but also consider kernel approaches in a possible infinite dimensional lifted space for any problem (see Section 2.4). The ability to use any predictor $T(x)$ and embed probabilistic guarantees to it offers a valuable extra layer of functionality to problems that already have a working estimator.*

2.4 Kernel central prediction and uncertainty quantification

Suppose that N_S i.i.d. samples $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N_S}$ are available. We now address the design of the predictor $T(x)$ by means of kernel methods while guaranteeing that the procedure also provides us with an estimation of $\sigma(x)$. Given x, let us define the cost function

$$J(\theta; x) = \theta^T \Sigma_\theta \theta + \sum_{i=1}^{N_S} (y^{(i)} - \theta^T \varphi(x^{(i)}))^2 \Gamma(x, x^{(i)}), \tag{2.11}$$

where θ is a design parameter, $\varphi : \mathbb{R}^{n_x} \to \mathbb{R}^{n_\theta}$ is the regressor function and $\theta^T \Sigma_\theta \theta$ is a regularization term. A possible choice is $\Sigma_\theta = \tau \mathrm{I}$, where $\tau > 0$. Finally, $\Gamma : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}$ is an appropriately chosen weighting function. We assume that $\Gamma(x, z)$ is a decreasing function of $\|x - z\|$, where $\| \cdot \|$ is a given norm. For example, $\Gamma(x, z) = \exp(-\lambda \|x - z\|)$, where $\lambda > 0$.

As it is usual in machine learning, for given x, a central estimation for y is provided by $T(x) = \theta_c^T(x)\varphi(x)$, where $\theta_c(x)$ is given by $\theta_c(x) = \arg\min_\theta J(\theta; x)$. We notice that the proposed estimator is a weighted least square estimator with a ridge regression regularization term [85, 86].

To obtain predictor $T(x)$ and local estimations $\hat{y}^{(i)}(x) = \theta_c^T(x)\varphi(x^{(i)})$, $i \in [N_S]$ required to compute the Parzen estimator for $\sigma(x)$, two possibilities are explored:
Based on $\varphi(\cdot)$: Since $J(\theta; x)$ is a strictly convex quadratic function of θ, the optimal value $\theta_c(x)$ can be obtained determining the value of θ for which the gradient of $J(\theta; x)$ with respect to θ vanishes.
Based on a kernel formulation: Defining the kernel function $K(\cdot, \cdot)$ as $K(x_a, x_b) = \varphi^T(x_a)\Sigma_\theta^{-1}\varphi^T(x_b)$, the estimation $T(x)$, along with the local estimations $\hat{y}(x^{(i)})$, $i \in [N_S]$, can be obtained in an explicit way by means of the well-known kernel trick (see e.g. [87, 88] and references therein). In this case, the kernel formulation allows to approach the regression problem in a possibly infinite dimensional lifted space [86]. This kernel formulation of the predictor $T(x)$ and the estimator of $\sigma(x)$ is detailed in the appendix 2.7.1.

Once the local estimations $\{\hat{y}(x^{(i)})\}_{i=1}^{N_S}$ have been computed, the estimation for $\sigma(x)$ follows from the following local Parzen estimator (see also equation (2.10)):

$$\hat{\sigma}^2(x) = \frac{\sum_{i=1}^{N_S} (y^{(i)} - \hat{y}(x^{(i)}))^2 \Gamma(x, x^{(i)})}{\sum_{i=1}^{N_S} \Gamma(x, x^{(i)})}. \tag{2.12}$$

See the Appendix 2.7.1 for a detailed description on how to obtain predictors $T(x)$ and $\hat{\sigma}(x)$ for both considered possibilities (i.e. based on a regressor $\varphi(\cdot)$ or based on a kernel formulation).

As commented before, the Parzen estimator converges, under non very restrictive assumptions, to the actual value $\sigma(x)$ as N_S tends to infinity ([81, 82]). A too reduced number of samples N_S, or a non appropriate choice for weighting factors

$\Gamma(\cdot,\cdot)$, may translate into a degraded estimation of $\sigma(x)$, which will not affect the probabilistic properties of the obtained bounds (since they are always guaranteed by Theorem 2.1), but will lead to more conservative bounds. We also notice that an additional set of N i.i.d. samples is required to compute the scaling factor $\bar{\gamma}$ in Algorithm 2.

2.5 Uncertainty quantification for finite families of estimators

The probabilistic bounds proposed for error $e = |y - T(x)|$ depend not only on the intrinsic random relationship between x and y (joint probability distribution), but also on the choice of the estimators T and $\hat{\sigma}$. Since there exists a myriad of possibilities for choosing the estimators, we now analyze the problem of choosing among a finite family $\mathcal{F}$ of $n_{\mathcal{F}}$ possible pairs, i.e. $\{(T_j, \hat{\sigma}_j)\}_{j=1}^{n_{\mathcal{F}}}$, the one that minimizes the size of the obtained probabilistic bounds. The following result states the relationship between the cardinality of $\mathcal{F}$ ($n_{\mathcal{F}}$), and the probabilistic specifications (ε, δ), with the number of samples required to obtain the corresponding bounds.

Theorem 2.1. *Consider the finite family of $n_{\mathcal{F}}$ candidate estimators*

$$\mathcal{F} = \{(T_j(\cdot), \hat{\sigma}_j(\cdot))\}_{j=1}^{n_{\mathcal{F}}},$$

where $T_j : \mathbb{R}^{n_x} \to \mathbb{R}$ and $\hat{\sigma}_j : \mathbb{R}^{n_x} \to (0, \infty)$ for every $j \in [n_{\mathcal{F}}]$. Given $\varepsilon \in (0,1)$, $\delta \in (0,1)$ and $r \geq 0$, let $N > r$ be such that $B(r; N, \varepsilon) \leq \frac{\delta}{n_{\mathcal{F}}}$. Draw N i.i.d. samples $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ from distribution $\Pr_{\mathcal{W}}$ and denote

$$\bar{\gamma}_j \doteq \left(\frac{|y^{(i)} - T_j(x^{(i)})|}{\hat{\sigma}_j(x^{(i)})} \right)_{N-r:N}, \quad j \in [n_{\mathcal{F}}]. \tag{2.13}$$

Then, with a probability no smaller than $1 - \delta$,

$$\Pr_{\mathcal{W}}\{|y - T_j(x)| > \bar{\gamma}_j \hat{\sigma}_j(x)\} \leq \varepsilon, \quad j \in [n_{\mathcal{F}}].$$

Proof. Denote $\delta_{\mathcal{F}}$ the probability that at least one of the randomly obtained scalars $\{\bar{\gamma}_j\}_{j=1}^{n_{\mathcal{F}}}$, obtained from the random multi-sample $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, does not satisfy the constraint

$$E_j(\bar{\gamma}_j) \doteq \Pr_{\mathcal{W}} \left\{ \frac{|y - T_j(x)|}{\hat{\sigma}_j(x)} > \bar{\gamma}_j \right\} \leq \varepsilon. \tag{2.14}$$

Thus,

$$\begin{aligned} \delta_{\mathcal{F}} &= \Pr_{\mathcal{W}^N}\{\varepsilon < \max_{j \in [n_{\mathcal{F}}]} E_j(\bar{\gamma}_j)\} \\ &\leq \sum_{j=1}^{n_{\mathcal{F}}} \Pr_{\mathcal{W}^N}\{\varepsilon < E_j(\bar{\gamma}_j)\} \leq \sum_{j=1}^{n_{\mathcal{F}}} \frac{\delta}{n_{\mathcal{F}}} = \delta. \end{aligned}$$

We notice that the last inequality is due to the assumption $B(r; N, \varepsilon) \leq \frac{\delta}{n_{\mathcal{F}}}$, (2.13) and Property 2.1. Thus, with probability no smaller than $1 - \delta_{\mathcal{F}} \geq 1 - \delta$, inequality (2.14) is satisfied for every $j \in [n_{\mathcal{F}}]$.

□

Remark 2.4 (Sample complexity for finite families)**.** *In view of Lemma 2.1, it suffices to draw* $N = \left\lceil \frac{7.47}{\varepsilon} \log \frac{n_{\mathcal{F}}}{\delta} \right\rceil$ *i.i.d. samples from* $\Pr_{\mathcal{W}}$ *to obtain a probabilistic uncertainty quantification for the complete finite family* $\mathcal{F}$*. In order to select the best pair* $(T_j, \hat{\sigma}_j)$ *in* $\mathcal{F}$*, one could choose e.g. the index* $j \in [n_{\mathcal{F}}]$ *providing the sharpest probabilistic uncertainty bounds, i.e. the one minimizing* $\sum_{i=1}^{N} \bar{\gamma}_j \hat{\sigma}_j(x^{(i)})$*. Since* $n_{\mathcal{F}}$ *enters in a logarithmic way in the sample complexity bound, large values for* $n_{\mathcal{F}}$ *are affordable. In this case, the search for the most appropriate pair* $(T_j, \hat{\sigma}_j)$ *does not need to be exhaustive, and sub-optimal search in the finite family* $\mathcal{F}$ *could be envisaged (since the probabilistic bounds provided are valid for every member of the family* $\mathcal{F}$*).*

2.5.1 Numerical example: Kernel finite families

We revisit now the numerical example proposed in Section 2.2.3. We use the predictor T and the estimator of σ detailed in the appendix 2.7.1 with the radial basis function kernel $k(x_a, x_b) = 50 \exp(-\frac{|x_a - x_b|^2}{0.2})$, and for the estimator $\hat{\sigma}(x)$ the Parzen estimator in (2.12), where $N_S = 2,065$ and the pairs $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N_S}$ are i.i.d. samples from $\Pr_{\mathcal{W}}$. We consider a family of weighting functions $\Gamma(x, z) = \exp(-\lambda|x - z|)$, where $\lambda \in [10]$. Thus, the finite family $\mathcal{F}$ consists of each of the $n_{\mathcal{F}} = 10$ possible pairs $(T_j, \hat{\sigma}_j)$ that can be obtained with the $n_{\mathcal{F}}$ values considered for the hyper-parameter λ using the methodology proposed in Section 2.4. Setting $\varepsilon = 0.05$, $\delta = 10^{-6}$ and $n_{\mathcal{F}} = 10$, we obtain from Theorem 2.1 and Lemma 2.1 that the choice $N = 2,407$ and $r = 60$ is sufficient to obtain a probabilistic uncertainty quantification valid for all the members of the family. The value of λ minimizing the size of the obtained probabilistic bounds is attained at $\lambda = 1$. The resulting scaling parameter is $\bar{\gamma} = 2.15$. See Figure 2.2 for a comparison of the results obtained for the same validation set that was used to generate Figure 3. The ratio of violation in the validation set for the proposed finite family approach was 0.0332, whereas it was 0.0511 for the exact probabilistic bounds.

2.6 Conclusions

In this chapter, a series of sample-based approaches to obtain probabilistic bounds of the absolute value of the prediction error of a given estimator have been presented. The proposed techniques share a probabilistic maximization scheme which only requires prediction models and random samples. Moreover, the sample complexity depends only on the desired probabilistic guarantees and not on the uncertainty distribution nor the prediction models. This allows to embed probabilistic guarantees to any predictor, even when they have a very large or even infinite

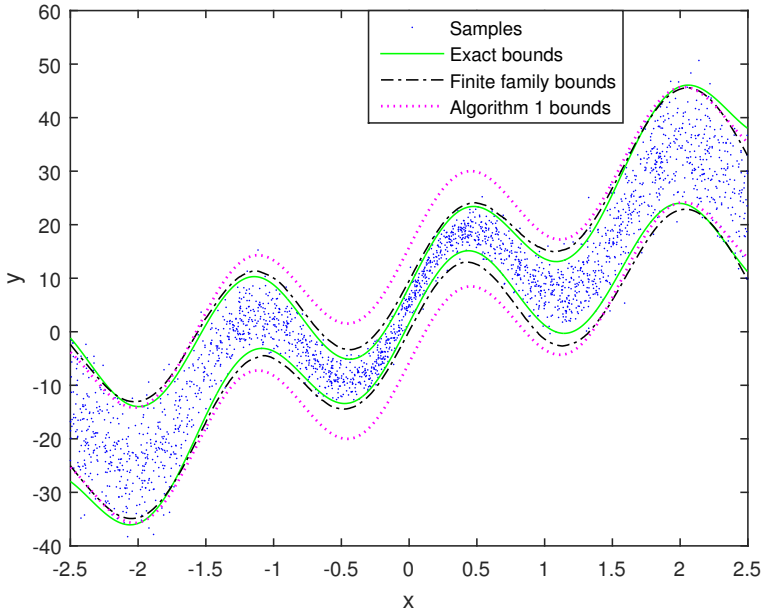


Figure 2.2: Probabilistic upper bounds obtained by means of a finite family of kernel estimators, Algorithm 1 bounds, and exact bounds.

number of decision variables. To choose the best probabilistic bounds from a family of different predictors, sample complexity has been adjusted.

Finally, the performance of the proposed approaches has been illustrated by means of a numerical example. This highlights how, although conservative, conditioned uncertainty quantification can get very close to the exact probabilistic bounds.

2.7 Appendix

2.7.1 Computation of estimators $T(x)$ and $\hat{\sigma}(x)$

Given a specific test point x and a given regressor function $\varphi(\cdot)$, the local ridge regression is calculated as $T(x) = \theta_c^T \varphi(x)$, where θ_c is the minimizer of (2.11), i.e.

$$\theta_c(x) = \arg\min_{\theta} \theta^T \Sigma_\theta \theta + \sum_{i=1}^{N_S} (y^{(i)} - \theta^T \varphi(x^{(i)}))^2 \Gamma(x, x^{(i)}).$$

We first notice that in some local regression approaches, $\Gamma(x, x^{(i)})$ is set to zero if $x^{(i)}$ does not belong to a neighbourhood of x. Similarly, function $\Gamma(x, \cdot)$ could be tuned in such a way that only $N_K \leq N_S$ samples $x^{(i)}$ satisfy $\Gamma(x, x^{(i)}) > 0$ (e.g. those closest to x). This sort of strategies are specially relevant in kernel methodologies because their implementation requires the solution of a system of

equations of N_K variables. Hence, the complexity of kernel approaches can be kept to affordable levels by adjusting the design parameter N_K.

Since $\Gamma(x, x^{(i)}) = 0$ implies that the pair $(x^{(i)}, y^{(i)})$ has no effect on the value of $\theta_c(x)$, we will consider only the pairs $(x^{(i)}, y^{(i)})$ for which $\Gamma(x, x^{(i)}) > 0$. We denote such pairs as $\{\tilde{x}^{(j)}, \tilde{y}^{(j)}\}_{j=1}^{N_K}$, where $1 \leq N_K \leq N_S$. Thus,

$$\theta_c(x) = \arg\min_{\theta} \quad \theta^T \Sigma_\theta \theta + \sum_{j=1}^{N_K} (\tilde{y}^{(j)} - \theta^T \varphi_j)^2 \Gamma_j,$$

where

$$\begin{aligned} \varphi_j &\doteq \varphi(\tilde{x}^{(j)}), \quad j = 1, \ldots, N_K, \\ \Gamma_j &\doteq \Gamma(x, \tilde{x}^{(j)}), \quad j = 1, \ldots, N_K. \end{aligned}$$

We first address the case in which regressor function $\varphi(\cdot)$ is available. Later, we address the situation in which the estimators are obtained in terms of a kernel formulation, i.e. when the estimators are not directly expressed in terms of a regressor function, but of a kernel function.

From $(\tilde{y}^{(j)} - \varphi_j^T \theta)^2 = \theta^T \varphi_j \varphi_j^T \theta - 2\tilde{y}^{(j)}(\theta^T \varphi_j) + (\tilde{y}^{(j)})^2$, we obtain that the minimizer of

$$J(\theta; x) = \theta^T \left(\Sigma_\theta + \sum_{j=1}^{N_K} \Gamma_j \varphi_j \varphi_j^T \right) \theta - 2\theta^T \left(\sum_{j=1}^{N_K} \Gamma_j \tilde{y}^{(j)} \varphi_j \right) \tag{2.15}$$

is

$$\theta_c(x) = \left(\Sigma_\theta + \sum_{j=1}^{N_K} \Gamma_j \varphi_j \varphi_j^T \right)^{-1} \left(\sum_{j=1}^{N_K} \Gamma_j \tilde{y}^{(j)} \varphi_j \right) = (\Sigma_\theta + RDR^T)^{-1} RD\tilde{y}_D, \tag{2.16}$$

where $R \in \mathbb{R}^{n_\theta \times N_K}, D \in \mathbb{R}^{N_K \times N_K}$ and $\tilde{y}_D \in \mathbb{R}^{N_K}$ are given by

$$\begin{aligned} R &= \begin{bmatrix} \varphi_1 & \varphi_2 & \ldots & \varphi_{N_K} \end{bmatrix}, \\ \tilde{y}_D &= \begin{bmatrix} \tilde{y}^{(1)} & \tilde{y}^{(2)} & \ldots & \tilde{y}^{(N_K)} \end{bmatrix}^T, \\ D &= \operatorname{diag}(\Gamma_1, \Gamma_2, \ldots, \Gamma_{N_K}) > 0. \end{aligned}$$

Thus, the estimator of y given x is

$$T(x) = \hat{y}(x) = \varphi(x)^T \theta_c(x) = \varphi(x)^T (\Sigma_\theta + RDR^T)^{-1} RD\tilde{y}_D.$$

We also define the local errors

$$\tilde{e}^{(j)}(x) = \tilde{y}^{(j)} - \varphi_j^T \theta_c(x), \quad j = 1, \ldots, N_K. \tag{2.17}$$

The resulting estimator $\hat{\sigma}(x)$ according to equation (2.12) is

$$\hat{\sigma}^2(x) = \frac{\sum_{j=1}^{N_K}(\tilde{y}^{(j)} - \hat{y}(x_j))^2\Gamma_j}{\sum_{j=1}^{N_K}\Gamma_j} = \frac{1}{\text{tr } D}\sum_{j=1}^{N_K}(\tilde{y}^{(j)} - \varphi(\tilde{x}^{(j)})^T\theta_c(x))^2\Gamma_j. \quad (2.18)$$

Defining $\|e\|_D = \sqrt{e^T De}$, we obtain

$$\begin{aligned}\hat{\sigma}(x) =& \frac{1}{\sqrt{\text{tr } D}}\left\|\begin{bmatrix}\tilde{y}^{(1)} - \varphi_1^T\theta_c(x)\\ \tilde{y}^{(2)} - \varphi_2^T\theta_c(x)\\ \vdots \\ \tilde{y}^{(N_K)} - \varphi_{N_K}^T\theta_c(x)\end{bmatrix}\right\|_D \\ =& \frac{1}{\sqrt{\text{tr } D}}\left\|\left(\mathbf{I} - R^T(\Sigma_\theta + RDR^T)^{-1}RD\right)\tilde{y}_D\right\|_D.\end{aligned} \quad (2.19)$$

Since the weighting factors $\Gamma(x,\cdot)$ depend on x, $\theta_c(x)$ also depends on x. Thus, the proposed procedure needs to be repeated each time the estimators $T(\bar{x})$ and $\hat{\sigma}(\bar{x})$ are required for a particular test point $\bar{x}$.

Now we recall the following matrix equality [89, Subsection 1.3], [87, Corollary 4.3.1]:

$$(H - RF^{-1}R^T)^{-1}RF^{-1} = H^{-1}R(F - R^TH^{-1}R)^{-1},$$

which is valid whenever H and F are non singular matrices. In view of this equality, we obtain from (2.16) the following expression for $\theta_c(x)$

$$\begin{aligned}\theta_c(x) &= (\Sigma_\theta + RDR^T)^{-1}RD\tilde{y}_D \\ &= -((-\Sigma_\theta) - RDR^T)^{-1}RD\tilde{y}_D \\ &= -(-\Sigma_\theta)^{-1}R(D^{-1} - R^T(-\Sigma_\theta)^{-1}R)^{-1}\tilde{y}_D \\ &= \Sigma_\theta^{-1}R(D^{-1} + R^T\Sigma_\theta^{-1}R)^{-1}\tilde{y}_D.\end{aligned} \quad (2.20)$$

Thus, given x and $\varphi_x = \varphi(x)$, we obtain the following estimation $\hat{y}(x) = T(x)$, where

$$\begin{aligned}T(x) = \varphi_x^T\theta_c(x) &= \varphi_x^T\Sigma_\theta^{-1}R(D^{-1} + R^T\Sigma_\theta^{-1}R)^{-1}\tilde{y}_D \\ &= \begin{bmatrix}\varphi_x^T\Sigma_\theta^{-1}\varphi_1\\ \varphi_x^T\Sigma_\theta^{-1}\varphi_2\\ \vdots \\ \varphi_x^T\Sigma_\theta^{-1}\varphi_{N_K}\end{bmatrix}^T (D^{-1} + K)^{-1}\tilde{y}_D,\end{aligned} \quad (2.21)$$

where

$$K = \begin{bmatrix}k(\tilde{x}^{(1)},\tilde{x}^{(1)}) & k(\tilde{x}^{(1)},\tilde{x}^{(2)}) & \dots & k(\tilde{x}^{(1)},\tilde{x}^{(N_K)})\\ k(\tilde{x}^{(2)},\tilde{x}^{(1)}) & k(\tilde{x}^{(2)},\tilde{x}^{(2)}) & \dots & k(\tilde{x}^{(2)},\tilde{x}^{(N_K)})\\ \vdots & \vdots & \ddots & \vdots \\ k(\tilde{x}^{(N_K)},\tilde{x}^{(1)}) & k(\tilde{x}^{(N_K)},\tilde{x}^{(2)}) & \dots & k(\tilde{x}^{(N_K)},\tilde{x}^{(N_K)})\end{bmatrix}. \quad (2.22)$$

The kernel function must satisfy the Meyer's condition, i.e. matrix K should be semidefinite positive for any collection of N_K points [90]. Popular kernel functions satisfying this condition are:

- Linear: $k(x_a, x_b) = x_a^T x_b$.
- Polynomial: $k(x_a, x_b) = (c + x_a^T x_b)^d$.
- Radial: $k(x_a, x_b) = \exp\left(\frac{-\|x_a - x_b\|^2}{d^2}\right)$.
- Sigmoidal: $k(x_a, x_b) = \tanh(c_a x_a^T x_b + c_b)$.

Using the kernel estimation in the local errors (2.17) we get

$$\begin{aligned}
\tilde{e}^{(j)}(x) =& \tilde{y}^{(j)} - \varphi_j^T \theta_c(x) = \tilde{y}^{(j)} - \varphi_j^T \Sigma_\theta^{-1} R (D^{-1} + R^T \Sigma_\theta^{-1} R)^{-1} y_D \\
=& \tilde{y}^{(j)} - \begin{bmatrix} \varphi_j^T \Sigma_\theta^{-1} \varphi_1 \\ \varphi_j^T \Sigma_\theta^{-1} \varphi_2 \\ \vdots \\ \varphi_j^T \Sigma_\theta^{-1} \varphi_{N_K} \end{bmatrix}^T (D^{-1} + K)^{-1} y_D \\
=& \tilde{y}^{(j)} - \begin{bmatrix} k(\tilde{x}^{(j)}, \tilde{x}^{(1)}) \\ k(\tilde{x}^{(j)}, \tilde{x}^{(2)}) \\ \vdots \\ k(\tilde{x}^{(j)}, \tilde{x}^{(N_K)}) \end{bmatrix}^T (D^{-1} + K)^{-1} y_D.
\end{aligned} \tag{2.23}$$

Thus, we obtain

$$\begin{bmatrix} \tilde{e}^{(1)}(x) \\ \tilde{e}^{(2)}(x) \\ \vdots \\ \tilde{e}^{(N_K)}(x) \end{bmatrix} = \left(\mathbf{I} - K(D + K)^{-1}\right) y_D.$$

We conclude from equation (2.18) that

$$\hat{\sigma}(x) = \frac{1}{\sqrt{\text{tr } D}} \| \left(\mathbf{I} - K(D + K)^{-1}\right) y_D \|_D.$$

Chapter 3

Regular Probabilistic Scaling

3.1 Introduction

The complexity of real-world applications and the random nature of data makes dealing with uncertainty essential. In many cases, uncertainty arises in the modeling phase, in others it is intrinsic to both the system and the operative environment, as for instance wind speed and turbulence in aircraft or wind turbine control [26].

Deriving results in the presence of uncertainty is of major relevance in different areas, including, but not limited to, optimization [91] and robustness analysis [38]. However, in contrast to robust approaches, where the goal is to determine a feasible solution which is optimal in some sense for all possible uncertainty instances, the goal in the stochastic framework is to find a solution that is feasible for a fraction of all possible uncertainty realizations, [50, 51]. In many situations involving random constraints, it is acceptable, up to a certain safe level, to enforce probabilistic constraints and therefore avoid the inherent conservativeness of robust constraints. The applications of the stochastic framework can be found in a myriad of fields, from finance to engineering, where the probabilistic approach has been used in unmanned autonomous vehicle navigation [92, 93] or in optimal power flow [94, 95] among others.

In the optimization framework, constraints involving stochastic parameters that are required to be satisfied with a pre-specified probability threshold are called chance constraints. In general, dealing with chance constraints implies facing two serious challenges: the solution of difficult parameterized probability integrals and the nonconvexity of the ensuing constraints [23]. Consequently, while being attractive from a modeling viewpoint, problems involving chance constraints are often computationally intractable, generally shown to be NP-hard, which seriously limits their applicability. However, being able to efficiently solve or approximate chance-constrained problems remains an important challenge, especially in systems and control.

The scientific community has devoted significant research to devising compu-

tationally efficient approaches to deal with chance constraints. We review such techniques in Section 3.3, where we highlight three mainstream approaches: exact techniques, robust approximations and sample-based approximations.

In this chapter, we present what we consider an important step forward in the sample-based approach. More precisely, our developments stem from the observation that, while in the general situation one is interested in finding an optimal solution to a chance-constrained problem, many practical applications just need inner approximations of chance-constrained sets. This can be the case, for instance, of stochastic model predictive control (SMPC), where this approximation is necessary for post-processing in real time (see e.g. [96, 97]).

Motivated by these considerations, we propose a simple and efficient strategy called regular probabilistic scaling. This approach calculates with an user-defined confidence a probabilistically guaranteed inner approximation of a chance-constrained set.

In particular, we describe a two-step procedure that involves: First, a preliminary approximation of the chance-constrained is set by means of a so-called Simple Approximating Set (SAS). Then, a sample-based scaling procedure that allows to properly scale the SAS so to guarantee the desired probabilistic properties.

The selection of a low-complexity SAS allows the designer to easily tune the complexity of the approximating set, significantly reducing the sample complexity but not the probabilistic guarantees. We propose several candidate SAS shapes, grouped in two classes: Sampled-polytopes and norm-based SAS.

The regular probabilistic scaling approach presented in this chapter distinguishes itself from the previous literature on chance constraints in the following main points.

1. It is specifically tailored towards the specific problem of approximating the chance-constrained set, as opposed to solving a specific instance of a chance-constrained problem.

2. It is very general: it applies to a very general class of uncertainty configurations. A large part of the methods available in the literature are limited to cases where the constraints depend in a "nice" way on the uncertainty. This is the case for instance of the solution proposed in [98, 39]. The reader is referred to Section 3.3 for an overview.

3. It is highly tunable: by selecting the complexity of the approximating set, the designer has a very efficient way to control the trade-off between computational complexity and potential goodness of the approximation.

The probabilistic scaling approach presented in this chapter is based on recent results on order statistics [69]. In this chapter we first perform a thorough mathematical analysis of probabilistic scaling. Then, we provide probabilistic guarantees for a more general class of norm-based SAS. After that, we consider joint chance constraints instead of independent chance constraints. This choice

is motivated by the fact that joint chance constraints, which have to be satisfied simultaneously, are a generalization of independent one and adhere better to some applications. Finally, we present here a possible application of probabilistic scaling related to probabilistic set membership identification.

This chapter is structured as follows. Section 3.2 provides a general preamble of the problem formulation and of chance-constrained optimization, including two motivating examples. An extensive overview on methods for approximating chance-constrained sets is reported in Section 3.3 whereas the probabilistic scaling approach has been detailed in Section 3.4. Section 3.5 is dedicated to the definition of selected candidate SAS, i.e. sampled-polytope and norm-based SAS. Last, in Section 3.6, we validate the proposed approach with a numerical example applying our method to a probabilistic set membership estimation problem. Main conclusions and future research directions are addressed in Section 3.7.

This chapter is based on the published paper [99] and its extended version [75].

3.2 Problem formulation

Consider a robustness problem, in which the controller parameters and auxiliary variables are parametrized by means of a decision variable vector $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$, which is usually referred to as design parameter.

Furthermore, the uncertainty vector $w \in \mathbb{R}^{n_w}$ represents one of the admissible uncertainty realizations of a random vector with given probability distribution $\Pr_{\mathcal{W}}$ and (possibly unbounded) support $\mathcal{W}$.

This chapter deals with the particular case where the design specifications can be decoded as a set of n_ℓ uncertain linear inequalities

$$g(\theta, w) \leq 0 \iff A(w)\theta \leq B(w), \tag{3.1}$$

where

$$A(w) = \begin{bmatrix} a_1^\top(w) \\ \vdots \\ a_{n_\ell}^\top(w) \end{bmatrix} \in \mathbb{R}^{n_\ell \times n_\theta}, \quad B(w) = \begin{bmatrix} b_1(w) \\ \vdots \\ b_{n_\ell}(w) \end{bmatrix} \in \mathbb{R}^{n_\ell},$$

are measurable functions of the uncertainty vector $w \in \mathbb{R}^{n_w}$.

In Section 3.7 we discuss possible extensions of this approach to more general settings, in which the constraints may be nonlinear and even nonconvex. Note that the proposed setup captures the special case of chance constraints with random right-hand side. These correspond to the choice $A(w) = A$ and $B(w) = w$. Similarly, the case of chance constraints with random technology matrix is captured by our general case.

We also note that hard linear constraints on θ may be directly incorporated by introducing deterministic inequalities of the form $a_\ell^\top \theta \leq b_\ell$, where a_ℓ and b_ℓ do not depend on the uncertainty w.

The inequality in (3.1) is to be interpreted component-wise, i.e. $a_\ell^\top(w)\theta \leq b_\ell(w)$, $\forall \ell \in [n_\ell]$. Due to the random nature of the uncertainty vector w, each realization of w corresponds to a different set of linear inequalities. Consequently, each value of w gives raise to the sample safe set

$$\Phi_0^g(w) \doteq \{\theta \in \Theta : g(\theta, w) \leq 0\} = \{\theta \in \Theta : A(w)\theta \leq B(w)\}. \tag{3.2}$$

In every application, one usually accepts a risk of violating the constraints. This often translates into a two-step strategy: First, a set $\tilde{\mathcal{W}}$ is obtained such that $w \in \tilde{\mathcal{W}}$ is satisfied with a pre-specified high probability; then, a robust design problem in which inequality (3.1) is forced to be satisfied for every $w \in \tilde{\mathcal{W}}$ is solved. This is for instance the approach taken in [98, 100].

This two-step strategy suffers from two main drawbacks which may cause the result to be conservative:

1. There is not guarantee that the ensuing robust problem is easily solvable. Indeed, it is in general very hard, and to obtain computable solutions the authors of [98] need to make additional assumptions on the dependence of A and B on the uncertainty w.

2. The approach in [98] does not provide a safe region (i.e. a probabilistic approximation of the chance-constrained set), but just a point satisfying the probabilistic constraint.

In this chapter, we observe that a less conservative solution can be found by choosing the set $\mathcal{W}$ to encompass all possible values and characterizing the region of the design space in which the fraction of elements of $\mathcal{W}$ that violate the constraints is below a specified level. This concept is rigorously formalized by means of the notion of probability of violation (see [51]).

Definition 3.1 (Probability of violation)**.** *Consider a probability measure* $\Pr_{\mathcal{W}}$ *over* $\mathcal{W}$ *and let* $\theta \in \Theta$ *be given. The probability of violation of* θ *relative to inequality* (3.1) *is defined as*

$$\mathit{Viol}(\theta) \doteq \Pr_{\mathcal{W}}\{A(w)\theta \not\leq B(w)\}.$$

Given a constraint on the probability of violation, i.e. $\mathrm{Viol}(\theta) \leq \varepsilon$, we denote as (joint) chance-constrained set of probability ε (shortly, ε-CCS) the region of the design space for which this probabilistic constraint is satisfied. This is formally stated in the next definition.

Definition 3.2 (ε-CCS)**.** *Given* $\varepsilon \in (0, 1)$*, we define the chance-constrained set of probability* ε *as follows*

$$\mathbb{X}_\varepsilon \doteq \{\theta \in \Theta : \mathit{Viol}(\theta) \leq \varepsilon\}. \tag{3.3}$$

Note that the ε-CCS represents the region of the design space Θ for which the probabilistic constraint $\text{Viol}(\theta) \leq \varepsilon$ is satisfied and it is equivalently defined as

$$\mathbb{X}_\varepsilon \doteq \{\theta \in \Theta : \Pr_{\mathcal{W}} \{A(w)\theta \leq B(w)\} \geq 1 - \varepsilon\}. \tag{3.4}$$

The chance-constrained set of probability 0 corresponds to the region of the design space for which the hard constraint $A(w)\theta \leq B(w)$ is satisfied for any value of the uncertainty $w \in \mathcal{W}$. Therefore, $\mathbb{X}_0 \subseteq \Phi_0^g(w), \forall w \in \mathcal{W}$.

As commented in section 1.2.1, a closed-form evaluation of $\mathbb{X}_\varepsilon$ is possible for only a handful of particular cases. Moreover, the ε-CCS is usually nonconvex, except for very special cases. For example, [101, Lemma 4.60] shows that the solution set of separable chance constraints can be written as the union of cones, which is nonconvex in general.

Example 3.1 (Example of nonconvex ε-CCS)**.** *To illustrate these inherent difficulties, we consider the following three-dimensional example ($n_\theta = 3$) with $w = \{w_1, w_2\}$, where the first uncertainty $w_1 \in \mathbb{R}^3$ is a three-dimensional normal-distributed random vector with zero mean and covariance matrix*

$$\Sigma = \begin{bmatrix} 4.5 & 2.26 & 1.4 \\ 2.26 & 3.58 & 1.94 \\ 1.4 & 1.94 & 2.19 \end{bmatrix},$$

and the second uncertainty $w_2 \in \mathbb{R}^3$ is a three-dimensional random vector whose elements are uniformly distributed in the interval $[0, 1]$. The set of viable design parameters is given by $n_\ell = 4$ uncertain linear inequalities of the form

$$\begin{aligned} A(w) &= \begin{bmatrix} w_1 & w_2 & (2w_1 - w_2) & w_1^2 \end{bmatrix}^T, \\ B(w) &= \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T \\ A(w)\theta &\leq B(w), \end{aligned}$$

where the square power w_1^2 is to be interpreted element-wise.

In this case, to obtain a graphical representation of the set $\mathbb{X}_\varepsilon$, we resorted to gridding the design set, and, for each point θ in the grid, to approximate the probability through a Monte Carlo method. This procedure is clearly unaffordable for higher dimensions, as the number of samples required would grow exponentially. In Figure 3.1 we report the plot of the obtained ε-CCS set for different values of ε. We observe that the set is indeed nonconvex.

3.2.1 Chance-constrained optimization

Finding an optimal $\theta \in \mathbb{X}_\varepsilon$ for a given cost function $J : \mathbb{R}^{n_\theta} \to \mathbb{R}$, leads to the chance-constrained optimization (CCO) problem

$$\min_{\theta \in \mathbb{X}_\varepsilon} J(\theta), \tag{3.5}$$

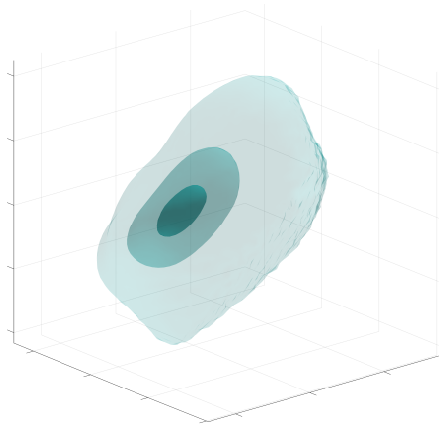


Figure 3.1: The ε-CCS set for $\varepsilon = 0.15$ (smaller set), $\varepsilon = 0.30$ (intermediate set), and $\varepsilon = 0.45$ (larger set). We observe that all sets are nonconvex, but the nonconvexity is more evident for larger values of ε, corresponding to larger levels of accepted violation, while the set $\mathbb{X}_\varepsilon$ appears "almost convex" for small values of ε. This kind of behaviour is in accordance with a recent result that proves convexity of the ε-CCS for small enough values of ε, and it is usually referred to as eventual convexity [28, 102].

where the cost-function $J(\theta)$ is usually assumed to be a convex (often even quadratic or linear) function.

We remark that the CCO problem (3.5) is in general NP-hard, for the same reasons reported before. We also note that several stochastic optimization problems arising in different application contexts can be formulated as a CCO. Typical examples are for instance the reservoir system design problem proposed in [103], where the problem is to minimize the total building and penalty costs while satisfying demands for all sites and all periods with a given probability, or the cash matching problem [104], where one aims at maximizing the portfolio value at the end of the planning horizon while covering all scheduled payments with a prescribed probability.

Chance-constrained optimization problems also frequently arise in short-term planning problems in power systems. These optimal power flow problems are routinely solved as part of the real-time operation of the power grid. The aim is determining minimum-cost production levels of controllable generators subject to reliably delivering electricity to customers across a large geographical area, see e.g. [94] and references therein.

Recently, approaches based on approximations of chance-constrained sets have emerged in the context of stochastic MPC, see [96, 17, 97]. These approaches exploit the sample-based results we summarize in Section 1.2.1 to construct offline a probabilistically guaranteed approximation of the set of all couples of control input/initial states that guarantee fulfillment of the desired input/state constraints.

The possibility of constructing a safe approximation of the chance-constrained

set offline constitutes a winning feature with respect to other sample-based approaches, since it moves all the cumbersome computation to the control design phase.

When a safe approximation is available, the online implementation can deal with the chance constraint as a deterministic set inclusion constraint. In this way, the original stochastic optimization program is reduced to an efficiently solvable convex program when the cost function J is convex. This represents an undiscussed advantage, which has been demonstrated for instance in [97]. We stress that the key element of this procedure is exactly the construction of an adequate approximation of the ε-CCS.

In the next subsection, we report an additional motivating example, which further highlights the importance of the problem at hand.

3.2.2 Motivating example: probabilistic set membership estimation

Consider the problem of finding all the values of $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$ such that

$$g(\theta, w) \leq 0 \Rightarrow |y - \theta^T \varphi(x)| \leq \rho, \quad \forall w \in \mathcal{W} \subseteq \mathbb{R}^{n_x} \times \mathbb{R},$$

where $w = (x, y)$, $\varphi : \mathbb{R}^{n_x} \to \mathbb{R}^{n_\theta}$ is a (possibly non-linear) regressor function, and $\rho > 0$ is a given hyperparameter accounting for modelling errors.

The deterministic set membership estimation problem, see [105, 106, 107], consists of computing the set of parameters θ that satisfy the constraint $|y - \theta^T \varphi(x)| \leq \rho$ for all possible values of $w \in \mathcal{W}$. In the literature, this set is usually referred to as the feasible parameter set (FPS) $\mathbb{X}_0$, i.e.

$$\mathbb{X}_0 \doteq \{\theta \in \Theta : |y - \theta^T \varphi(x)| \leq \rho, \quad \forall w \in \mathcal{W}\}.$$

We notice that $\mathbb{X}_0$ could be empty if ρ is chosen too small. For any realization of the uncertainty $w \in \mathcal{W}$, we can define the sample safe set

$$\Phi_0^g(w) = \{\theta \in \Theta : |y - \theta^T \varphi(x)| \leq \rho\},$$

and we can rewrite the feasible parameter as

$$\mathbb{X}_0 = \{\theta \in \Theta : \theta \in \Phi_0^g(w), \ \forall w \in \mathcal{W}\}.$$

The deterministic set membership problem suffers from the following limitations in real applications:

- Due to the possible non-linearity of $\varphi(\cdot)$, checking if a given $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$ satisfies the constraint $\theta \in \Phi_0^g(w)$ for every $w \in \mathcal{W}$ is often a difficult problem.
- In many situations, the information about the uncertainty w comes through a number of samples, and therefore only outer bounds of the FPS can be computed. As a result, the robust constraint can not be checked.

- Because of outliers and possible non-finite support of $\mathcal{W}$, some problem instances may not have any point in Θ that guarantees the fulfillment of every possible constraint, and thus, the feasible parameter set is empty (even for large values of ρ).

To deal with these issues, one can resort to a probabilistic relaxation of the FPS. Given a probability distribution defined on $\mathcal{W}$, the probabilistic set membership estimation problem is that of characterizing the set of parameters θ that satisfy the chance constraint

$$\Pr_{\mathcal{W}}\{|y - \theta^T \varphi(x)| \leq \rho\} \geq 1 - \varepsilon$$

for a given probability level $\varepsilon \in (0, 1)$. Hence, we can define the ε-CCS $\mathbb{X}_\varepsilon$ as the set of parameters that satisfy the previous probabilistic constraint, that is,

$$\mathbb{X}_\varepsilon = \{\theta \in \Theta : \Pr_{\mathcal{W}}\{\theta \in \Phi_0^g(w)\} \geq 1 - \varepsilon\}.$$

It is immediate to notice that this problem fits in the formulation proposed in this section: It suffices to define

$$A(w) = \begin{bmatrix} \varphi^T(x) \\ -\varphi^T(x) \end{bmatrix}, \; B(w) = \begin{bmatrix} \rho + y \\ \rho - y \end{bmatrix}.$$

3.2.3 Chance-constrained set approximations

Motivated by the discussion above, we formulate the main problem studied in this chapter.

Problem 3.1 (ε-CCS approximation)**.** *Given the set of linear inequalities* (3.1)*, and a violation parameter ε, find an inner approximation of the ε chance-constrained set $\mathbb{X}_\varepsilon$. The approximation should be simple enough, accurate enough and easily computable.*

A sample-based approximation of the ε-CCS is provided in this chapter. In particular, regarding simplicity, we present a solution of tunable complexity in which the approximating set could be represented by few linear inequalities. As for the accuracy and ease of computation, we propose a highly tunable and computationally efficient procedure for its construction (see Algorithm 3).

Before presenting our approach, in the next section we provide a literature overview of different sample-based methods presented in the literature to construct approximations of the ε-CCS set and solve the CCO.

3.3 Overview on sample-based approaches to approximate the ε-CCS

In section 1.2.1, we went through the different approaches to approximate the ε-CCS and ultimately solve the chance-constrained optimization problem. Now,

we recapitulate the sample-based approaches that approximate the ε-CCS and go through one practical application in the field of automatic control.

Sampling-based techniques are characterized by the use of a finite number N of i.i.d. samples of the uncertainty $\mathbf{w}_N = \{w^{(1)}, w^{(2)}, \ldots, w^{(N)}\}$, each of them drawn according to the probability distribution $\Pr_{\mathcal{W}}$. With each sample $w^{(i)}, i \in [N]$, we can associate the following sample safe set

$$\Phi_0^g(w^{(i)}) = \{\theta \in \Theta : A(w^{(i)})\theta \leq B(w^{(i)})\}, \tag{3.6}$$

sometimes referred to as scenario, since it represents an observed instance of the uncertain constraint.

The scenario approach presented in [52] considers the CCO problem (3.5) and approximates its solution through the following optimization problem

$$\begin{aligned} \theta_{sc}^* = \arg\min_{\theta} \quad & J(\theta) \\ \text{s.t.} \quad & \theta \in \Phi_0^g(w^{(i)}), \quad \forall i \in [N]. \end{aligned} \tag{3.7}$$

We note that, given a convex cost function $J(\theta)$, problem (3.7) becomes a linearly constrained convex program, for which very efficient solution approaches exist. Under some technical assumptions (feasibility of the problem and non-degeneracy), a fundamental result [52, 53, 54, 55] provides a probabilistic certification of the constraint satisfaction for the solution to the scenario problem. In particular, it is shown that

$$\Pr_{\mathcal{W}^N}\{\mathrm{Viol}(\theta_{sc}^*) > \varepsilon\} \leq B(n_\theta - 1; N, \varepsilon), \tag{3.8}$$

where the probability in (3.8) is measured with respect to the samples $\mathbf{w}_N = \{w^{(i)}\}_{i=1}^N$.

A few observations are at hand regarding the scenario approach and its relationship with Problem 3.1. First, if we define the multisample safe set as

$$\Phi_0^g(\mathbf{w}_N) \doteq \bigcap_{i=1}^{N} \Phi_0^g(w^{(i)}), \tag{3.9}$$

we see that the scenario approach consists in approximating the constraint $\theta \in \mathbb{X}_\varepsilon$ in (3.5) with its sampled version $\theta \in \Phi_0^g(\mathbf{w}_N)$. On the other hand, it should be remarked that the scenario approach cannot be used to derive any guarantee on the existing relationship between $\Phi_0^g(\mathbf{w}_N)$ and $\mathbb{X}_\varepsilon$.

Indeed, the nice probabilistic property in (3.8) holds only for the optimum of the scenario program θ_{sc}^*. This is a fundamental point, since the scenario results build on the so-called support constraints, which are defined for the optimum point θ_{sc}^* only.

On the contrary, the probabilistic scaling approach that will be presented in the remainder of this chapter focuses on establishing a direct relation (in probabilistic terms) between the set $\Phi_0^g(\mathbf{w}_N)$ and the ε-CCS $\mathbb{X}_\varepsilon$. This is indeed possible, but one needs to resort to results based on Statistical Learning Theory [108] and in [49, Theorem 8], summarized in the following lemma.

Lemma 3.1 (Statistical Learning Theory bound)**.** *Given probabilistic levels* $\delta \in (0,1)$ *and* $\varepsilon \in (0, 0.14)$*, if the number of samples* N *is chosen so that* $N \geq N_{LT}$*, with*

$$N_{LT} \doteq \frac{4.1}{\varepsilon}\Big(\ln\frac{21.64}{\delta} + 4.39 n_\theta \log_2\Big(\frac{8 e n_\ell}{\varepsilon}\Big)\Big), \tag{3.10}$$

then $\Pr_{\mathcal{W}^N}\{\Phi_0^q(\mathbf{w}_N) \subseteq \mathbb{X}_\varepsilon\} \geq 1-\delta$.

The lemma, whose proof is reported in Appendix 3.8.1, is a direct consequence of the statistical learning theory results on the so-called (α, k)-Boolean functions, given in [49, Corollary 4], where more general results are reported for cases in which ε is not constrained in $(0, 0.14)$.

Remark 3.1 (Sample-based stochastic MPC)**.** *The statistical learning theory approach discussed in this section has been applied in [96] to derive offline a probabilistic inner approximation of the chance-constrained set defining the couples of input/state guaranteeing the desired input/state chance. In particular, the bound* (3.10) *is a direct extension to the case of joint chance constraints of the result proved in [96] for individual chance constraints.*

Note that since we are considering multiple constraints at the same time (like in (2)), the number of constraints n_ℓ *enters into the sample size bound. To explain how the SMPC design in [96] extends to the joint chance constraints framework, we briefly recall it.*

First, we extract offline (i.e. when designing the SMPC control) N *i.i.d. samples of the uncertainty,* $\sigma_k^{(i)}$ *of* σ_k*, and we consider the sampled set*

$$\mathbb{X}^{SMPC}(\sigma_k^{(i)}) = \left\{ \begin{bmatrix} x_k \\ \mathbf{v}_k \end{bmatrix} : \begin{bmatrix} f_\ell^x(\sigma_k^{(i)}) \\ f_\ell^v(\sigma_k^{(i)}) \end{bmatrix}^T \begin{bmatrix} x_k \\ \mathbf{v}_k \end{bmatrix} \leq 1, \ell \in [n_\ell] \right\},$$

and $\mathbb{X}_N^{SMPC} \doteq \bigcap_{i=1}^{N} \mathbb{X}^{SMPC}(\sigma_k^{(i)})$*. Then, applying Lemma 3.1 with* $n_\theta = n_x + n_u T$ *(where* T *is the prediction horizon), we conclude that if we extract* $N \geq N_{LT}^{SMPC}$ *samples, it is guaranteed that, with probability at least* $1-\delta$*, the sample approximation* $\mathbb{X}_N^{SMPC}$ *is a subset of the original chance constraint* $\mathbb{X}_\varepsilon^{SMPC}$*. Exploiting these results, the SMPC problem can be approximated conservatively by the linearly constrained quadratic program*

$$\begin{aligned} \min_{\mathbf{v}_k} \quad & J_T(x_k, \mathbf{v}_k) \\ s.t. \quad & (x_k, \mathbf{v}_k) \in \mathbb{X}_N^{SMPC}. \end{aligned}$$

Hence, when the cost function J_T *is quadratic, this approximation reduces the original stochastic optimization program to an efficiently solvable quadratic program. This represents an undiscussed advantage, which has been demonstrated for instance in [97]. On the other hand, it turns out that the ensuing number of linear constraints, equals to* $n_\ell N_{LT}^{SMPC}$ *may still be too large. For example, even for a moderately sized MPC problem with* $n_x = 5$ *states,* $n_u = 2$ *inputs,*

prediction horizon of $T = 10$*, simple interval constraints on states and inputs (i.e.* $n_\ell = 2n_x + 2n_u = 14$*), and for a reasonable choice of probabilistic parameters, i.e.* $\varepsilon = 0.05$ *and* $\delta = 10^{-6}$*, we get* $N_{LT}^{SMPC} = 114,530$*, which in turn corresponds to more than 1.6 million linear inequalities. For this reason, in [96] a post-processing step was proposed to remove redundant constraints. While it is indeed true that all the cumbersome computations may be performed offline, it is still the case that, in applications with stringent requirements on the solution time, the final number of inequalities may easily become unbearable.*

Remark 3.1 motivates the approach presented in the next section. We show how the probabilistic scaling approach directly leads to approximations of user-chosen complexity, which can be directly used in applications. Furthermore, the required number of samples is independent of the dimension of the problem. This feature is specially relevant in high-dimensional problems.

3.4 The probabilistic scaling approach

We propose a novel sample-based approach, alternative to the state-of-the-art randomized procedures. This scheme allows to maintain the probabilistic guarantees of these techniques, while at the same time providing a way of tuning the complexity of the approximation.

The probabilistic scaling approach consists of a two-step procedure: First, an initial simple approximating set (SAS) $c \oplus \Omega_0$ which captures the shape of the probabilistic set $\mathbb{X}_\varepsilon$. Then, the scaling step calculates the value of γ that embeds the desired probabilistic guarantees to the scalable SAS

$$\Omega(\gamma) \doteq c \oplus \gamma\Omega_0. \tag{3.11}$$

Simple approximating sets are characterized by a scaling center c and a shape Ω_0, which constitute the design parameters of the proposed approach. By appropriately selecting the shape Ω_0, the complexity of the approximating set can be tuned. The nonnegative scalar γ controls the size of the scalable SAS $\Omega(\gamma)$: the larger γ is, the larger $\Omega(\gamma)$ will be.

Note that this initial SAS $c \oplus \Omega_0$ does not offer any guarantee of probabilistic nature. It should be able to capture somehow the shape of the ε-CCS. Some possible options for this initial set are provided in Section 3.5.

The scaling center c and the shape Ω_0 constitute the starting point of the probabilistic scaling procedure, which allows to derive an approximation of the ε-CCS which meets pre-specified probabilistic guarantees, as detailed in the next subsection.

3.4.1 Probabilistic Scaling

In this section, we address the problem of how to scale the set $\Omega(\gamma)$ around the scaling center c to guarantee, with pre-specified confidence level $\delta \in (0, 1)$,

the inclusion of the scaled set into $\mathbb{X}_\varepsilon$. Within this sample-based procedure we assume that N_γ i.i.d. random samples $\{w^{(1)}, \dots, w^{(N_\gamma)}\}$ with distribution $\Pr_{\mathcal{W}}$ are available. We will use these samples to obtain a scalar $\bar{\gamma} > 0$ such that

$$\Pr_{\mathcal{W}^{N_\gamma}}\{\Omega(\bar{\gamma}) \subseteq \mathbb{X}_\varepsilon\} \geq 1 - \delta.$$

To this end, we first define the scaling factor associated to a given realization of the uncertainty.

Definition 3.3 (Scaling factor)**.** *Given a scalable SAS $\Omega(\gamma)$ defined by a scaling center $c \in \Theta$ and a shape Ω_0, and a realization $w \in \mathcal{W}$, we define the scaling factor of $\Omega(\gamma)$ relative to the random constraint $g(\theta, w) \leq 0$ as*

$$\gamma(w) \doteq \begin{cases} 0 & \text{if } c \notin \Phi_0^g(w) \\ \max\limits_{\Omega(\gamma) \subseteq \Phi_0^g(w)} \gamma & \text{otherwise,} \end{cases} \tag{3.12}$$

with $\Phi_0^g(w) = \{\theta \in \Theta : g(\theta, w) \leq 0\}$ and $\Omega(\gamma)$ defined as in (3.11).

That is, $\gamma(w)$ represents the maximal scaling that can be applied to $\Omega(\gamma) = c \oplus \gamma\Omega_0$ around the scaling center c so that $\Omega(\gamma) \subseteq \Phi_0^g(w)$.

The following theorem, whose proof is reported in Appendix 3.8.2, states how to obtain, by means of sampling, a scaling factor $\bar{\gamma}$ that guarantees, with high probability, that $\Omega(\bar{\gamma}) \subseteq \mathbb{X}_\varepsilon$.

Theorem 3.1 (Probabilistic scaling)**.** *Given a candidate scalable SAS $\Omega(\gamma)$ defined by the scaling center $c \in \Theta$ and the shape Ω_0, suppose that the accuracy parameter $\varepsilon \in (0, 1)$, the confidence level $\delta \in (0, 1)$, the integer discarding parameter $r \geq 0$ and the integer sample size $N_\gamma \geq r$ are chosen such that*

$$B(r; N_\gamma, \varepsilon) \leq \delta. \tag{3.13}$$

Draw N_γ i.i.d. samples $\{w^{(1)}, w^{(2)}, \dots, w^{(N_\gamma)}\}$ from distribution $\Pr_{\mathcal{W}}$, compute their corresponding scaling factors

$$\gamma_i \doteq \gamma(w^{(i)}), \quad i \in [N_\gamma]$$

according to Definition 3.3, and define

$$\bar{\gamma} = \left(\{\gamma_i\}_{i=1}^{N_\gamma}\right)_{1+r:N_\gamma},$$

i.e. $\bar{\gamma}$ is the $(1 + r)$-th smallest value of $\{\gamma_i\}_{i=1}^{N_\gamma}$ (see Notation section). Under these assumptions:

(i) If $\bar{\gamma} > 0$, then with probability no smaller than $1 - \delta$,

$$\Omega(\bar{\gamma}) = c \oplus \bar{\gamma}\Omega \subseteq \mathbb{X}_\varepsilon.$$

(ii) If $c \notin \mathbb{X}_\varepsilon$ then $\bar{\gamma} = 0$ with probability no smaller than $1 - \delta$.

In the following remark, we recall a result from the Lemmas 2.1 and 2.2 of chapter 2.

Remark 3.2. *In order to satisfy*

$$B(r; N_\gamma, \varepsilon) \leq \delta$$

it suffices to take $r = \left\lfloor \frac{\varepsilon N_\gamma}{2} \right\rfloor$ *and* N_γ *such that*

$$N_\gamma \geq \frac{7.47}{\varepsilon} \ln \frac{1}{\delta}.$$

The above result leads to the following algorithm, which shows how to calculate the scaling factor $\bar{\gamma}$ so that, with probability no smaller than $1 - \delta$, the scaled SAS is contained in the ε-CCS, i.e. $\Omega(\bar{\gamma}) \subseteq \mathbb{X}_\varepsilon$.

Algorithm 3 Probabilistic SAS Scaling

1: Given a candidate scalable SAS $\Omega(\gamma)$, and probability levels ε and δ, choose

$$N_\gamma \geq \frac{7.47}{\varepsilon} \ln \frac{1}{\delta} \quad \text{and} \quad r = \left\lfloor \frac{\varepsilon N_\gamma}{2} \right\rfloor. \tag{3.14}$$

2: Draw N_γ samples of the uncertainty $w^{(1)}, \ldots, w^{(N_\gamma)}$.
3: **for** $i = 1$ to N_γ **do**
4: Compute, according to Definition 3.3, the N_γ scaling factors

$$\gamma_i \doteq \gamma(w^{(i)}),\ i \in [N_\gamma]. \tag{3.15}$$

5: **end for**
6: Return the $(1 + r)$-th smallest value of $\{\gamma_i\}_{i=1}^{N_\gamma}$, i.e. $\bar{\gamma} = \left(\{\gamma_i\}_{i=1}^{N_\gamma}\right)_{1+r:N_\gamma}$.

In step 4 of Algorithm 3, one has to solve an optimization problem for each uncertainty sample $w^{(i)}$, which amounts to finding the largest value of γ such that $\Omega(\gamma)$ is contained in the set $\Phi_0^g(w^{(i)})$ defined in (3.6). If the SAS is chosen appropriately, we can show that this problem is convex and computationally very efficient: this is discussed in Section 3.5. Then, in step 6, one can use a partial sort algorithm (see discussion below about sorting algorithms) to find the $(1 + r)$-th smallest element of the sequence $\{\gamma_1, \gamma_2, \ldots, \gamma_{N_\gamma}\}$.

Full vs. partial sort algorithms

In the context of sorting algorithms, full sort refers to an algorithm that sorts a list of values and retrieves the full list sorted [109, 110], whereas partial sort algorithms retrieve only a sorted subset of the list consisting of the m smallest (or largest) elements [111, 112].
Partial sort algorithms benefit from a reduced average complexity, specially when m is small compared to the size of the full list. Given a list of n values, partial sort algorithms can average a time complexity of $\mathcal{O}(n + m \log m)$ to retrieve the m smallest elements. Full sort algorithms would average a time complexity of $\mathcal{O}(n \log n)$ for retrieving the full sorted list. This implies that the gap in complexity becomes larger with an increasing difference between n and m.

The properties of the output of Algorithm 3 can be derived by a direct application of Theorem 3.1 and Remark 3.2. In particular, if the output $\bar{\gamma}$ is larger than zero, then $\Omega(\gamma) \subseteq \mathbb{X}_\varepsilon$ with probability no smaller than $1 - \delta$.

In the next section, we provide a handful of possible candidate SAS shapes. We remind that SAS should capture the shape of the ε-CCS while keeping its complexity low, so that the resulting approximation does not cause a major bottleneck in its applications, such as chance-constrained optimization.

3.5 Candidate SAS

3.5.1 Sampled-polytope

A straightforward way to design a candidate SAS is to resort to a sample-based procedure, i.e. draw a fixed number N_S of design uncertainty samples[1] $\tilde{\mathbf{w}}_{N_S} = \{\tilde{w}^{(i)}\}_{i=1}^{N_S}$, and use the sampled-polytope

$$c \oplus \Omega_0 = \Phi_0^g(\tilde{\mathbf{w}}_{N_S}) = \bigcap_{j=1}^{N_S} \Phi_0^g(\tilde{w}^{(j)}). \tag{3.16}$$

as the SAS.

Note that the sampled polytope $\Phi_0^g(\tilde{\mathbf{w}}_{N_S})$ by construction is defined by the intersection of $n_\ell N_S$ half-spaces. Hence, we observe that this approach provides very precise control on the final complexity of the approximation, through the choice of the number of samples N_S.

Unlike the statistical learning theory bound N_{LT} defined in Lemma 3.1, the choice of N_S is fundamentally determined by the complexity of the polytope and therefore, the resulting initial geometry does not offer any probabilistic guarantees. These guarantees are instead provided by the probabilistic scaling procedure

[1] These samples are denoted with a tilde ($\tilde{}$) to distinguish them from the samples used in the probabilistic scaling procedure.

discussed in Section 4.4. It should be also remarked that multiple heuristics may be used to build the polytope SAS. For instance, one could discard some extreme samples in order to make the SAS more robust.

In addition to the shape Ω_0, the proposed probabilistic scaling also requires a scaling center c around which to apply the scaling procedure. One sensible option would be the Chebyshev center, which for a given norm $\|\cdot\|_p$, is defined as the center of the largest ball inscribed in $\Phi_0^g(\tilde{\mathbf{w}}_{N_S})$, i.e. $c = \mathrm{Cheb}_p(\Phi_0^g(\tilde{\mathbf{w}}_{N_S}))$. Once the scaling center c has been fixed, the scaling procedure detailed in algorithm 3 can be applied to the sampled-polytope SAS defined as

$$\Omega(\gamma) \doteq c \oplus \gamma\Big(\Phi_0^g(\tilde{\mathbf{w}}_{N_S}) \ominus c\Big). \tag{3.17}$$

We note that computing the Chebyshev center of a given polytope amounts to solving a convex optimization problem, for which efficient algorithms exist, see e.g. [113]. Another option would be to use the analytic center of $\Phi_0^g(\tilde{\mathbf{w}}_{N_S})$ as the scaling center, which can also be easily computed (see [113] for further details).

Similar to the shape Ω_0, the choice of c does not embed any probabilistic guarantees and only affects the goodness of the shape. In most applications, it is impossible to know a priori which option of the scaling center results in a better SAS.

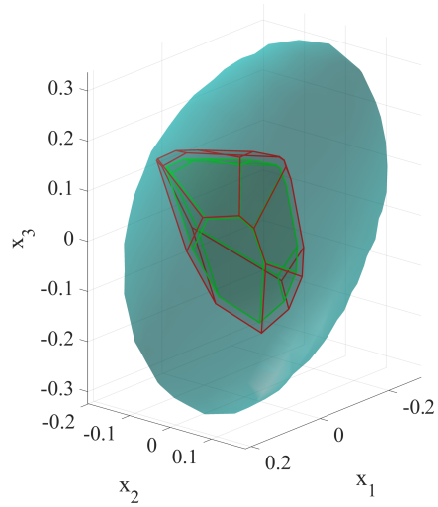
Example 3.2. *[Sample-based approximations] To illustrate how the proposed scaling procedure works in practice in the case of sampled-polytope SAS, we revisit Example 3.1. To this end, a pre-fixed number N_S of uncertainty samples were drawn, and the set of inequalities*

$$A(\tilde{w}^{(j)})\theta \leq B(\tilde{w}^{(j)}), \quad j \in [N_S],$$

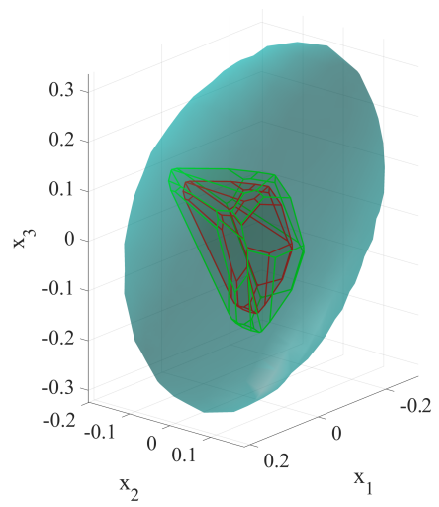
with $A(w)$ and $B(w)$ defined in (3.1) *were set, leading to the set $\Phi_0^g(\tilde{\mathbf{w}}_{N_S})$. Then, its Chebyshev center with respect to norm $\|\cdot\|_2$ was computed, and Algorithm 3 was applied to the sampled-polytope SAS $\Omega(\gamma)$ defined in* (3.17)*, with $\varepsilon = 0.05$, $\delta = 10^{-6}$, leading to $N_\gamma = 2,065$.*

We note that, in this case, the solution of the optimization problem in (3.12) *may be obtained by bisection on γ. Indeed, for given γ, checking if $\Omega(\gamma) \subseteq \Phi_0^g(w^{(i)})$ amounts to solving some simple linear programs.*

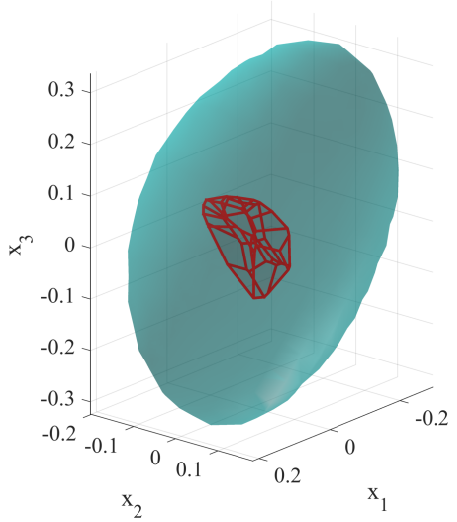
Two different situations were considered: a case where the number of inequalities is rather small $N_S = 100$, and a case where the complexity of the SAS is higher, $N_S = 1,000$. The outcome procedure is illustrated in Figure 3.2. We can observe that, for the smaller N_S (Figure 3.2a) the initial approximation is rather large (although it is contained in $\mathbb{X}_\varepsilon$, there is no guarantee that this will happen). In this case, the probabilistic scaling returns $\bar{\gamma} = 0.8954$ which is less than one. This means that, in order to obtain a set fulfilling the desired probabilistic guarantees, we need to shrink it around the scaling center. In the second case, for a larger number of sampled inequalities (Figure 3.2b) the initial set (the red one) is much smaller, and the scaling procedure enlarges the SAS by returning a value of



(a) $\Omega(\gamma)$ with $N_S = 100$. $\rightarrow \bar{\gamma} = 0.8954$



(b) $\Omega(\gamma)$ with $N_S = 1,000$. $\rightarrow \bar{\gamma} = 1.2389$



(c) $\Omega(\gamma)$ Based on Lemma 3.1 ($N_{LT} = 52,044$)

Figure 3.2: (a-b) Probabilistic scaling approximations of the ε-CCS. Scaling procedure applied to a sampled-polytope with $N_S = 100$ (a) and $N_S = 1,000$ (b). The initial sets are depicted in red, the scaled ones in green. (c) Approximation obtained by direct application of Lemma 3.1. Note that, in this latter case, to plot the set without out-of-memory errors a pruning procedure [114] of the $52,044$ linear inequalities was necessary.

$\bar{\gamma}$ greater than one, i.e. $\bar{\gamma} = 1.2389$. Note that using a larger number of samples in the computation of the initial set does not imply that the resulting SAS will capture better the shape of the ε-CCS.

Finally, we compare this approach to the scenario-like ones discussed in Subsection 1.2.1. To this end, we also draw the approximation obtained by directly applying the Statistical Learning Theory bound (3.10)*. Note that in this case, since $n_\theta = 3$ and $n_\ell = 4$, we need to take $N_{LT} = 13,011$ samples, corresponding to $52,044$ linear inequalities. The resulting set is represented in Figure 3.2c. Compared to the proposed probabilistic scaling approximation, we point out that the one made by the statistical learning theory approach is much more complex, since the number of involved inequalities is much larger and it is also much smaller, hence providing a much more conservative approximation of the ε-CCS. Hence, the ensuing chance-constrained optimization problem will be computationally harder, and will lead to a solution with a larger cost or even to an infeasible problem, in cases where the approximating set is too small.*

3.5.2 Candidate SAS: Norm-based SAS

Another option for the shape of the approximation are the so-called norm-based SAS. This family of sets have a low complexity and can also be build from samples. They are defined as

$$\Omega(\gamma) \doteq c \oplus \gamma H \mathbb{B}_p^s, \tag{3.18}$$

where $\mathbb{B}_p^s$ is an ℓ_p-ball in $\mathbb{R}^s$, $H \in \mathbb{R}^{n_\theta \times s}$, with $s \geq n_\theta$, is a design matrix (not necessarily square), and γ is the scaling factor. Note that when the matrix H is square (i.e. $s = n_\theta$) and positive definite these sets belong to the class of ℓ_p-norm based sets originally introduced in [115]. In particular, in case of ℓ_2 norm, the sets are ellipsoids. This particular choice is the one studied in [70]. Here, the authors extend this approach to a much more general family of sets, which encompasses for instance zonotopes, obtained by letting $p = \infty$ and $s \geq n_\theta$. Zonotopes have been widely studied in geometry, and have found several applications in systems and control, in particular for problems of state estimation and robust Model Predictive Control (see e.g. [116]), because of their flexibility and very efficient implementations.

Scaling factor computation for norm-based SAS

From (3.3), we know that the scaling factor of a given sample w, i.e. $\gamma(w)$, is defined as 0 if $c \notin \Phi_0^g(w)$ and as the largest value γ for which $\Omega(\gamma) \subseteq \Phi_0^g(w)$ otherwise. The following theorem, whose proof is reported in Appendix 3.8.3, provides a direct and simple way to compute in closed form the scaling factor for a given candidate norm-based SAS.

Theorem 3.2 (Scaling factor for norm-based SAS)**.** *Given a norm-based SAS* $\Omega(\gamma) = c \oplus \gamma H\mathbb{B}_p^s$ *and a realization* $w \in \mathcal{W}$*, define* $\tau_\ell(w) \doteq b_\ell(w) - a_\ell^T(w)c$ *and* $\rho_\ell(w) \doteq \|H^T a_\ell(w)\|_{p^*}$*, with* $\|\cdot\|_{p^*}$ *being the dual norm of* $\|\cdot\|_p$*.*

The scaling factor $\gamma(w)$ *can be computed as*

$$\gamma(w) = \min_{\ell \in [n_\ell]} \gamma_\ell(w),$$

with $\gamma_\ell(w)$*,* $\ell \in [n_\ell]$*, given by*

$$\gamma_\ell(w) = \begin{cases} 0 & \text{if } \tau_\ell(w) < 0, \\ \infty & \text{if } \tau_\ell(w) \geq 0 \text{ and } \rho_\ell(w) = 0 \\ \frac{\tau_\ell(w)}{\rho_\ell(w)} & \text{if } \tau_\ell(w) \geq 0 \text{ and } \rho_\ell(w) > 0 \end{cases}$$

Note that $\gamma(w)$ is equal to zero if and only if c is not included in the interior of $\Phi_0^g(w)$.

Construction of a candidate norm-based set

Similarly to the construction of the polytope SAS discussed in 3.5.1, to build the norm-based SAS we also start by drawing a fixed number N_S of design uncertainty samples $\{\tilde{w}^{(i)}\}_{i=1}^{N_S}$, and use them to construct an initial sampled approximation (sampled-polytope SAS) $\Phi_0^g(\tilde{\mathbf{w}}_{N_S})$ by means of (3.16). Again, there are multiple possibilities for the center c. Here, we take the Chebyshev center of $\Phi_0^g(\tilde{\mathbf{w}}_{N_S})$, or its analytical center as the center c for our approach.

Analogously to what was proposed in [70], given $\Phi_0^g(\tilde{\mathbf{w}}_{N_S})$, $s \geq n_\theta$ and $p \in \{1, 2, \infty\}$, the objective is to compute the largest set $c \oplus H\mathbb{B}_p^s$ included in $\Phi_0^g(\tilde{\mathbf{w}}_{N_S})$. To this end, we assume that we have a function $\mathrm{Vol}_p(H)$ that provides a measure of the size of $H\mathbb{B}_p^s$. That is, larger values of $\mathrm{Vol}_p(H)$ are obtained for increasing sizes of $H\mathbb{B}_p^s$.

Remark 3.3 (On the volume function)**.**
The function $Vol_p(H)$ *may be seen as a generalization of the classical concept of Lebesgue volume of the set* $\Phi_0^g(\tilde{\mathbf{w}}_{N_S})$*. Indeed, when* H *is a square positive definite matrix, some possibilities are* $Vol_p(H) = \log\det(H)$*, which is directly proportional to the classical volume definition, or* $Vol_p(H) = \mathrm{tr(H)}$*, which for* $p = 2$ *becomes the well known sum of ellipsoid semiaxes (see [117] and [113]). These measures can be easily generalized to non square matrices. It suffices to compute the singular value decomposition. If* $H = U\Sigma V^T$*, we could use the measures* $Vol_p(H) = tr(\Sigma)$ *or* $Vol_p(H) = \log\det(\Sigma)$*.*
For non square matrices H*, specific results for particular values of* p *are known. For example, we remind that if* $p = \infty$ *and* $H \in \mathbb{R}^{n_\theta \times s}$*,* $s \geq n_\theta$*, then* $c \oplus H\mathbb{B}_\infty^s$ *is a zonotope. Then, if we denote as generator each of the columns of* H*, the volume of a zonotope can be computed by means of a sum of terms (one for each different way of selecting* n_θ *generators out of the* s *generators of* H*); see [118], [119].*

Another possible measure of the size of a zonotope $c \oplus H\mathbb{B}_\infty^s$ is the Frobenious norm of H [118].

Given an initial design set $\Phi_0^g(\tilde{\mathbf{w}}_{N_S})$, the candidate scalable SAS is the norm-based SAS with the largest volume contained in $\Phi_0^g(\tilde{\mathbf{w}}_{N_S})$. Formally, this rewrites as the following optimization problem

$$\begin{aligned}(c, H) = \arg\max_{c,H} \quad & \mathrm{Vol}_p(H) \\ \text{s.t.} \quad & c \oplus H\mathbb{B}_p^s \subseteq \Phi_0^g(\tilde{\mathbf{w}}_{N_S}).\end{aligned} \tag{3.19}$$

As it has been shown (see Appendix 3.8.3), problem (3.19) is equivalent to

$$\begin{aligned}(c, H) = \arg\min_{c,H} \quad & -\mathrm{Vol}_p(H) \\ \text{s.t.} \quad & a_\ell^T(\tilde{w}^{(j)})c + \|H^T a_\ell(\tilde{w}^{(j)})\|_{p^*} - b_\ell(\tilde{w}^{(j)}) \leq 0 \\ & \ell \in [n_\ell] \\ & j \in [N_S],\end{aligned} \tag{3.20}$$

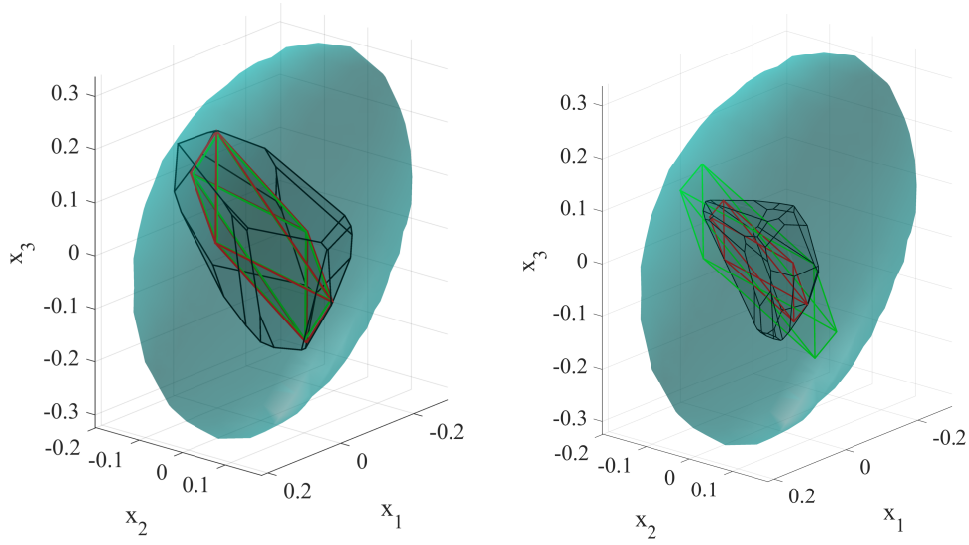
where the maximization of $\mathrm{Vol}_p(H)$ has been replaced with the minimization of -$\mathrm{Vol}_p(H)$.

We notice that the constraints are convex on the decision variables, and the objective function to minimize is convex under particular assumptions. For example when H is assumed to be square and positive definite and $\mathrm{Vol}_p(H) = \log\det(H)$. For non square matrices, the constraints remain convex, but the convexity of the functional to be minimized is often lost. In this case, local optimization algorithms may be employed to obtain a possibly sub-optimal solution.

Example 3.3 (Norm-based SAS)**.** *We revisit again Example 3.1 to show the use of norm-based SAS. We note that, in this case, the designer can control the approximation outcome by acting upon the number of design samples N_S used for constructing the initial approximation Ω_0. In Figure 3.3 we report two different norm-based SAS, respectively with $p = 1$ and $p = \infty$, and for each of them we consider two different values of N_S, respectively $N_S = 100$ and $N_S = 1,000$. Similar to the results of Example 3.2, we see that for larger N_S, the ensuing initial set becomes smaller. Consequently, we have a shrinkage process for small N_S and an inflating one for large N_S. However, we observe that in this case, the final number of inequalities is independent of N_S (8 inequalities for ℓ_1 and 6 for ℓ_∞).*

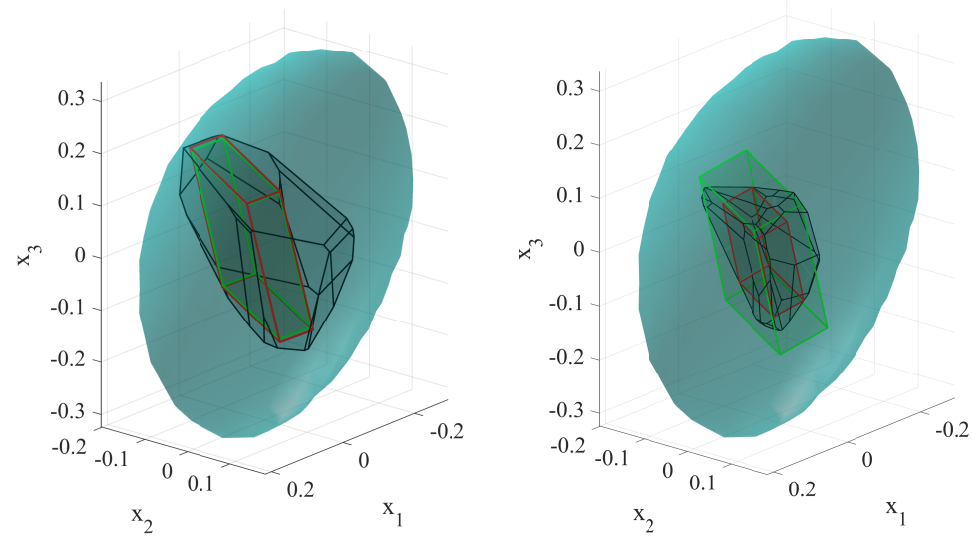
Relaxed computation

It is worth remarking that the minimization problem of the previous subsection might be infeasible. In order to guarantee the feasibility of the problem, a soft-constrained optimization problem is proposed. With a relaxed formulation, c is not longer guaranteed to satisfy all the sampled constraints. Note that $c \in \Phi_0^g(\tilde{\mathbf{w}}_{N_S})$ is not a necessary condition in the probabilistic scaling procedure.



(a) $\bar{\gamma} = 0.9701$

(b) $\bar{\gamma} = 1.5995$



(c) $\bar{\gamma} = 0.9696$

(d) $\bar{\gamma} = 1.5736$

Figure 3.3: Scaling procedure applied to (a) ℓ_1-SAS with $N_S = 100$, (b) ℓ_1-SAS with $N_S = 1,000$, (c) ℓ_∞-SAS with $N_S = 100$, and (d) ℓ_∞-SAS with $N_S = 1,000$. The initial set is depicted in red, the final one in green. The sampled polytope constructed from N_S samples (see equation (3.16)) is represented in black.

Given $\xi > 0$, the relaxed version of optimization problem (3.20) is

$$\begin{aligned} \underset{c,H,\eta_1,\ldots,\eta_{N_S}}{\arg\min} \quad & -\mathrm{Vol}_p(H) + \xi \sum_{j=1}^{N_S} \max\{\eta_j, 0\} \\ \text{s.t.} \quad & a_\ell^T(\tilde{w}^{(j)})c + \|H^T a_\ell(\tilde{w}^{(j)})\|_{p^*} - b_\ell(\tilde{w}^{(j)}) \leq \eta_j \\ & \ell \in [n_\ell] \\ & j \in [N_S]. \end{aligned} \tag{3.21}$$

The parameter ξ serves to provide an appropriate trade-off between satisfaction of the sampled constraints and the size of the obtained region. A possibility to choose ξ would be to choose it in such a way that the fraction of violations n_{viol}/N_S (where n_{viol} is the number of elements η_j larger than zero) is smaller than $\varepsilon/2$.

3.5.3 Further alternatives

The sampled-polytope and the norm-based SAS presented in sections 3.5.1 and 3.5.2 are just two possible options for the SAS. Later, in the following chapter (Section 4.7.1), a new ellipsoidal SAS will also be explored. These three families of geometries explored in this thesis can lead to good approximations of the ε-CCS because of their simplicity and, since they are based of samples of the system, they should somehow capture the shape of the ε-CCS. However, infinite possibilities exist for the SAS. Because the SAS does not embed probabilistic guarantees, every convex shape is valid as an initial set as long as the scaling center c is contained in the ε-CCS.

In some cases, previous knowledge about the system can be used to construct the SAS and intricate (convex) geometries may be taken into account in favour of obtaining a better fit to the ε-CCS.

3.6 Numerical example: Probabilistic set membership estimation

We now present a numerical example in which the results of the chapter are applied to the probabilistic set membership estimation problem, introduced in subsection 3.2.2. We consider the universal approximation functions given by gaussian radial basis function networks (RBFN) [120].

Given M nodes $\{x_1, x_2, \ldots, x_M\}$ and the variance parameter v, the corresponding gaussian radial basis function network is defined as

$$\mathrm{RBFN}(x, \theta) = \theta^T \varphi(x),$$

where $\theta = \begin{bmatrix} \theta_1 & \ldots & \theta_M \end{bmatrix}^T$ represents the weights and

$$\varphi(x) = \left[\exp\left(\frac{-\|x-x_1\|^2}{v}\right) \quad \exp\left(\frac{-\|x-x_2\|^2}{v}\right) \quad \dots \quad \exp\left(\frac{-\|x-x_M\|^2}{v}\right)\right]^T$$

is the regressor function. Given probabilities $\varepsilon \in (0,1)$ and $\delta \in (0,1)$, the objective is to approximate the set $\mathbb{X}_\varepsilon$ so that, with probability no smaller than $1-\delta$, the approximating set $\Omega(\bar{\gamma})$ is contained in the ε-CCS $\mathbb{X}_\varepsilon$, which is the set of parameters $\theta \in \Theta$ that satisfies

$$\Pr_{\mathcal{W}}\{|y - \theta^T\varphi(x)| \leq \rho\} \geq 1-\varepsilon, \tag{3.22}$$

where $\rho = 5$, x is a random scalar with uniform distribution in $[-5,5]$, and

$$y = \sin(3x) + \sigma,$$

with σ being a random scalar with a normal distribution with mean 5 and variance 1.

We use the procedure detailed in Sections 3.4 and 3.5 to obtain a SAS of $\mathbb{X}_\varepsilon$. We have taken a grid of $M = 20$ points in the interval $[-5,5]$ to serve as nodes for the RBFN, and a variance parameter of $v = 0.15$. We have taken $N_S = 350$ random samples $w = (x,y)$ to compute the initial geometry, which has been chosen to be an ℓ_∞ norm-based SAS of dimension 20 with a relaxation parameter of $\xi = 1$ (see equation (3.21)). The chosen initial geometry is $c \oplus H\mathbb{B}_\infty^{20}$, where H is constrained to be a diagonal matrix.

When the initial geometry is obtained, we scale it around its center by means of probabilistic scaling with Algorithm 3. The number of samples required for the scaling phase to achieve $\varepsilon = 0.05$ and $\delta = 10^{-6}$ is $N_\gamma = 2,065$ and the resulting scaling factor is $\bar{\gamma} = 0.3803$. The scaled geometry $c \oplus \bar{\gamma}H\mathbb{B}_\infty^{20}$ is, with a probability no smaller than $1-\delta$, an inner approximation of $\mathbb{X}_\varepsilon$. Since it is a transformation of an ℓ_∞ norm ball with a diagonal matrix H, we can write it as

$$\Omega(\bar{\gamma}) = \{\theta : \theta^- \leq \theta \leq \theta^+\},$$

where the extreme values $\theta^-, \theta^+ \in \mathbb{R}^{20}$ are represented in Figure 3.4, along with the central value $c \in \mathbb{R}^{20}$.

Once the approximation $\Omega(\bar{\gamma})$ has been computed, we use its center c to make the point estimation $y \approx c^T\varphi(x)$. We can also obtain probabilistic upper and lower bounds of y by means of equation (3.22). That is, every point $\theta \in \Omega(\bar{\gamma})$ satisfies, with confidence $1-\delta$:

$$\begin{aligned} &\Pr_{\mathcal{W}}\{y \leq \theta^T\varphi(x) + \rho\} \geq 1-\varepsilon, \\ &\Pr_{\mathcal{W}}\{y \geq \theta^T\varphi(x) - \rho\} \geq 1-\varepsilon. \end{aligned} \tag{3.23}$$

We notice that the tightest probabilistic bounds are obtained with θ^+ for the lower bound and θ^- for the upper one. That is, we finally obtain that, with confidence

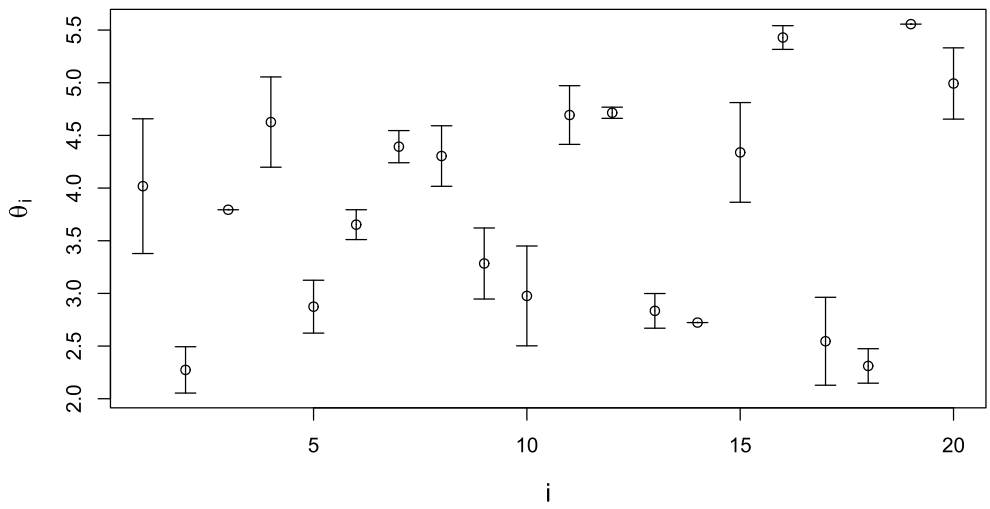


Figure 3.4: Representation of the extreme values θ^+ and θ^- along with the central value c of $\Omega(\bar{\gamma})$.

$1-\delta$:

$$\begin{aligned} &\Pr_{\mathcal{W}}\{y \leq \theta^{-T}\varphi(x) + \rho\} \geq 1-\varepsilon, \\ &\Pr_{\mathcal{W}}\{y \geq \theta^{+T}\varphi(x) - \rho\} \geq 1-\varepsilon. \end{aligned} \tag{3.24}$$

Figure 3.5 shows the results of both the point estimation and the probabilistic interval estimation.

3.7 Concluding remarks

In this chapter, a general approach to construct probabilistically guaranteed inner approximations of the chance-constrained set $\mathbb{X}_\varepsilon$ has been proposed. The approach is very general and flexible. In this section, we report a few final remarks on some important aspects of the presented methodology.

3.7.1 On scalability of the proposed approach

The proposed framework provides different schemes with different computational requirements. In particular, regarding the norm-based sets discussed in Section 3.5.2, Theorem 3.2 provides a closed-form expression for the scaling computations. Hence, the approach scales extremely well when the initial candidate set $c \oplus H\mathbb{B}_p^s$ is given.

When the initial set is a polytope (see section 3.5.1), the scaling computation is indeed more involved, since usually there is not a close-form expression. In

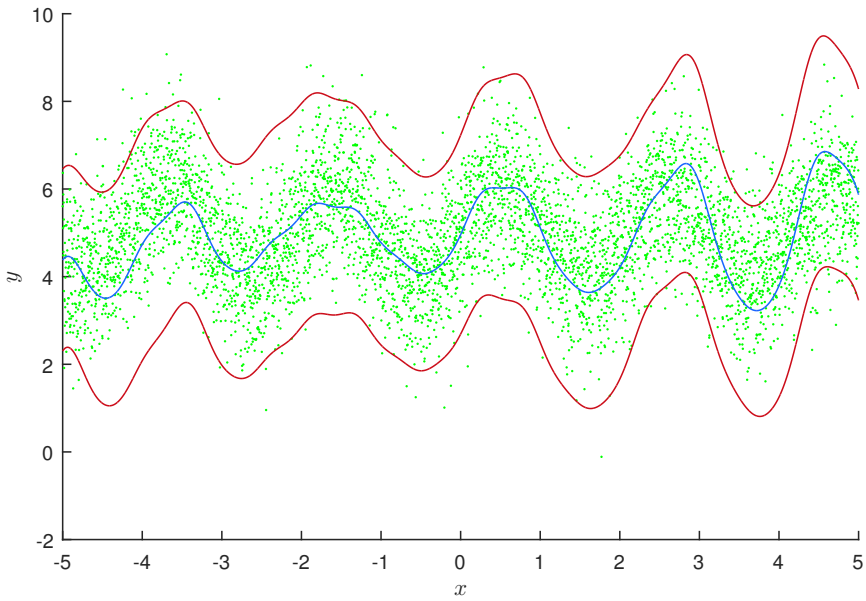


Figure 3.5: Real values of y (green dots), central estimation (blue line) and interval prediction bounds (red line).

this case, the solution of the optimization problem in (3.15) may be obtained by bisection on γ. Note that in this case, given γ, checking if $\Omega(\gamma) \subseteq \Phi_0^g(w^{(i)})$ amounts to solving a linear program.

Otherwise, when the set $c \oplus H\mathbb{B}_p^s$ is not available, its computation will clearly constitute the most demanding step of our scheme. In this case, as detailed in Section 6, c and H can be obtained by means of a convex optimization problem when H is a square matrix. Depending on the choice on H, the number of decision variables increases linearly with the dimension of c (e.g. H is a diagonal matrix), or quadratically (if H is a full matrix). The richer is the family of initial candidate sets (e.g. when the initial set is a zonotope), the more demanding will be its computation.

The proposed approach does not always lead to better approximations than other state-of-the-art approaches. There will be situations where the solutions discussed in Section 3.3 may be preferable. On the other hand, a nice and distinctive feature of the proposed approach is that it can be complementary to these approaches: for instance, given any convex approximation of the ε-CCS, one could use this set as initial SAS to which apply the probabilistic scaling procedure. If the resulting scaling factor $\bar{\gamma}$ is greater than 1, then the initial approximation can be enlarged while keeping the desired probabilistic guarantees.

Moreover, it should be remarked that the tunability of the proposed approach, while allowing high flexibility, entails by definition the problem of parameter selection. In our case, the main degree of freedom is the choice of the initial scalable

set. In this case, the trade-off is evident: the more complex the set is, the tighter the obtained approximation may be, at the expense of a possibly larger computational effort. Besides this clear implication, a more detailed analysis, both theoretical and experimental, is needed to understand the effect of specific choices of the initial set (as those introduced in Section 3.5). This is an important point that however goes beyond the scope of this thesis, and is the subject of ongoing research.

3.7.2 Extensions to nonlinear setups

We remark that the proposed scaling approach is not limited to sets defined by linear inequalities, and it may be extended to more general sets using very similar arguments. Indeed, we may consider a generic binary performance function $I^g : \Theta \times \mathcal{W} \to \{0, 1\}$ defined as[2]

$$I^g(\theta, w) = \begin{cases} 0 & \text{if } \theta \text{ meets design specifications for } w \\ 1 & \text{otherwise.} \end{cases} \tag{3.25}$$

In this case, the violation probability may be written as $\text{Viol}(\theta) \doteq \Pr_{\mathcal{W}}\{I^g(\theta, w) = 1\}$, and we can still define the set $\mathbb{X}_\varepsilon$ as in (3.3). Then, given an initial SAS candidate, Algorithm 3 still provides a valid approximation. However, it should be remarked that, even if we choose a very simple approximating set as those previously introduced, the nonconvexity of I^g will most probably render step 4 of the algorithm 3 intractable for many problems.

To further elaborate on this point, let us focus on the case when the design specification may be expressed as a (nonlinear) inequality of the form

$$g(\theta, w) \leq 0.$$

Then, the computation of each scaling factor γ_i of step 4 of Algorithm 3 consists, provided that $c \in \Phi_0^g(w^{(i)})$, in solving the following nonconvex optimization problem

$$\begin{aligned} \gamma_i \doteq \arg\max \quad & \gamma \\ \text{s.t.} \quad & c \oplus \gamma\Omega_0 \subseteq \Phi_0^g(w^{(i)}) = \left\{\theta \in \Theta \mid g(\theta, w^{(i)}) \leq 0\right\}. \end{aligned}$$

We note that this is generally a hard problem. However, there are cases when this problem is still solvable. In particular, we remark that whenever $g(\theta, w)$ is a convex function of θ for fixed w and the set Ω_0 is also convex, the above optimization problem may be formulated as a convex program.

[2] This nonlinear formulation encompasses the discussed setup, obtained by simply setting $I^g(\theta, w) = \begin{cases} 0 & \text{if } A(w)\theta \leq B(w) \\ 1 & \text{otherwise.} \end{cases}$

3.7.3 Future directions

In the previous subsection, we discussed how the proposed method might be extended to nonlinear setups. One may wonder whether the approach could also be extended to the important class of problems involving integer values, such as the mixed-integer programming studied in [32]. This is a problem currently under investigation, however the extension in this case is far from being trivial. While the presented approach is generalizable in theory, there is still no computationally efficient implementation for it.

3.8 Appendix

3.8.1 Proof of Lemma 3.1

To prove the lemma, we first recall the following definition from [49].

Definition 3.4 ((α, k)-Boolean function)**.** *The function $h : \Theta \times \mathcal{W} \to \mathbb{R}$ is an (α, k)-Boolean function if for fixed w it can be written as an expression consisting of Boolean operators involving k polynomials $p_1(\theta), p_2(\theta), \ldots, p_k(\theta)$, in the components θ_i, $i \in [n_\theta]$ and the degree with respect to θ_i of all these polynomials is no larger than α.*

Let now define the binary functions

$$h_\ell(\theta, w) \doteq \left\{ \begin{array}{ll} 0 & \text{if } a_\ell(w)\theta \leq b_\ell(w) \\ 1 & \text{otherwise} \end{array} \right\}, \quad \ell \in [n_\ell].$$

Introducing the function $h(\theta, w) \doteq \max_{\ell=1,\ldots,n_\ell} h_\ell(\theta, w)$, we see that the violation probability can be alternatively written as $\text{Viol}(\theta) \doteq \Pr_{\mathcal{W}}\{h(\theta, w) = 1\}$. We notice that $h(\theta, w)$ is an $(1, n_\ell)$-Boolean function, since it can be expressed as a function of n_ℓ boolean functions, each of them involving a polynomial of degree 1. The proof now follows from Theorem 8 in [49] that states that if $h : \Theta \times \mathcal{W} \to \mathbb{R}$ is an (α, k)-Boolean function and $\varepsilon \in (0, 0.14)$ then, with probability greater than $1 - \delta$, we have $\Pr_{\mathcal{W}}\{h(\theta, w) = 1\} \leq \varepsilon$ if N is chosen such that

$$N \geq \frac{4.1}{\varepsilon}\Big(\ln \frac{21.64}{\delta} + 4.39 n_\theta \log_2 \Big(\frac{8e\alpha k}{\varepsilon} \Big) \Big).$$

□

3.8.2 Proof of Theorem 3.1

To prove the theorem, we first prove the following property.

Property 3.1. *Given $\varepsilon \in (0, 1)$, $\delta \in (0, 1)$, and $r \geq 1$, let $N \geq r$ be such that $B(r; N, \varepsilon) \leq \delta$. Draw N i.i.d. samples $\{w^{(1)}, w^{(2)}, \ldots, w^{(N)}\}$ from a distribution $\Pr_{\mathcal{W}}$. For $i \in [N]$, let $\gamma_i \doteq \gamma(w^{(i)})$, with $\gamma(\cdot)$ as in Definition 3.3, and suppose*

that $\bar{\gamma} = \left(\{\gamma_i\}_{i=1}^N\right)_{1+r:N} > 0$. *Then, with probability no smaller than* $1-\delta$, *it holds that* $\Pr_{\mathcal{W}}\{c \oplus \bar{\gamma}\Omega_0 \not\subseteq \Phi_0^g(w)\} \leq \varepsilon$.

Proof: It has been proven in [54, 55] that if one discards no more than s constraints on a convex problem with N random constraints, then the probability of violating the constraints with the solution obtained from the random convex problem is no larger than $\varepsilon \in (0,1)$, with probability no smaller than $1-\delta$, where

$$\delta = \binom{s+d-1}{s} \sum_{i=0}^{s+d-1} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i},$$

and d is the number of decision variables. We apply this result to the following optimization problem

$$\begin{aligned} \max_{\gamma} \quad & \gamma \\ \text{s.t.} \quad & c \oplus \gamma\Omega_0 \subseteq \Phi_0^g(w^{(i)}), \quad i \in [N]. \end{aligned} \tag{3.26}$$

From Definition 3.3, we could rewrite this optimization problem as

$$\begin{aligned} \max_{\gamma} \quad & \gamma \\ \text{s.t.} \quad & \gamma \leq \gamma(w^{(i)}), \ i \in [N]. \end{aligned} \tag{3.27}$$

We first notice that the problem under consideration is convex and has a unique scalar decision variable γ. That is, $d = 1$. Also, the non-degeneracy and uniqueness assumption required in the application of the results of [54] and [55] are satisfied. We notice that $\bar{\gamma} = \left(\{\gamma_i\}_{i=1}^N\right)_{1+r:N}$, is the optimal solution to the optimization problem when $s = r$ constraints are discarded. Thus, we have that with probability no smaller than $1-\delta$, where

$$\delta = \binom{r}{r} \sum_{i=0}^{r} \binom{N}{i} \varepsilon^i (1-\varepsilon)^{N-i} = B(r; N, \varepsilon),$$

the choice $\bar{\gamma} = \left(\{\gamma_i\}_{i=1}^N\right)_{1+r:N}$ satisfies $\Pr_{\mathcal{W}}\{\bar{\gamma} > \gamma(w)\} \leq \varepsilon$.

We conclude from this, and Definition 3.3, that with probability no smaller than $1-\delta$, $\Pr_{\mathcal{W}}\{c \oplus \bar{\gamma}\Omega_0 \not\subseteq \Phi_0^g(w)\} \leq \varepsilon$.

□

Proof of Theorem 1

We consider first the case $\bar{\gamma} > 0$. From Property 3.1, we have that $\bar{\gamma} > 0$ satisfies, with probability no smaller than $1-\delta$, that

$$\Pr_{\mathcal{W}}\{\Omega(\bar{\gamma}) \not\subseteq \Phi_0^g(w)\} \leq \varepsilon.$$

Equivalently,

$$\Pr_{\mathcal{W}}\{\Omega(\bar{\gamma}) \subseteq \Phi_0^g(w)\} \geq 1-\varepsilon,$$

which can be rewritten as

$$\Pr_{\mathcal{W}}\{A(w)\theta \leq B(w), \quad \forall \theta \in \Omega(\bar{\gamma})\} \geq 1-\varepsilon,$$

and it implies that the probability of violation in $c \oplus \bar{\gamma}\Omega_0$ is no larger than ε, with probability no smaller than $1-\delta$. This proves the first claim.
Suppose now that $c \notin \mathbb{X}_\varepsilon$. This is equivalent to $\mathrm{Viol}(c) = \bar{\varepsilon}_c > \varepsilon$. Suppose that the sample constraints $c \in \Phi_0^g(w^{(i)})$, $i \in [N_\gamma]$ are violated n_{viol} times. This would imply, because of the definition of scaling factor, that there are at least n_{viol} scaling factors $\gamma(w^{(i)})$ equal to zero. From this and $\mathrm{Viol}(c) = \bar{\varepsilon}_c > \varepsilon$, we obtain

$$\begin{aligned}\Pr_{\mathcal{W}^{N_\gamma}}\{\bar{\gamma} > 0\} &= \Pr_{\mathcal{W}^{N_\gamma}}\{\Big(\{\gamma_i\}_{i=1}^{N_\gamma}\Big)_{1+r:N} > 0\} \\ &\leq \Pr_{\mathcal{W}^{N_\gamma}}\{n_{\mathrm{viol}} < r\} \\ &= B(r; N_\gamma, \bar{\varepsilon}_c) \leq B(r; N_\gamma, \varepsilon) \leq \delta.\end{aligned}$$

From here we conclude that $c \notin \mathbb{X}_\varepsilon$ implies

$$\Pr_{\mathcal{W}^{N_\gamma}}\{\bar{\gamma} = 0\} = 1 - \Pr_{\mathcal{W}^{N_\gamma}}\{\bar{\gamma} > 0\} \geq 1-\delta.$$

□

3.8.3 Proof of Theorem 3.2

Note that, by definition, the condition $c \oplus \gamma H \mathbb{B}_p^s \subseteq \Phi_0^g(w)$ is equivalent to

$$\max_{z \in \mathbb{B}_p^s} \quad a_\ell^T(w)(c + \gamma H z) - b_\ell(w) \leq 0, \ \ell \in [n_\ell].$$

Equivalently, from the dual norm definition, we have

$$a_\ell^T(w)c + \gamma \|H^T a_\ell(w)\|_{p^*} - b_\ell(w) \leq 0, \ \ell \in [n_\ell].$$

Denote by γ_ℓ the scaling factor γ_ℓ corresponding to the ℓ-th constraint

$$a_\ell^T(w)c + \gamma_\ell \|H^T a_\ell(w)\|_{p^*} - b_\ell(w) \leq 0.$$

With the notation introduced in the theorem, this constraint rewrites as

$$\gamma_\ell \rho_\ell(w) \leq \tau_\ell(w).$$

The result follows noting that the corresponding scaling factor $\gamma_\ell(w)$ can be computed as

$$\gamma_\ell(w) = \max_{\gamma_\ell \rho_\ell(w) \geq \tau_\ell(w)} \gamma_\ell,$$

and that the value for $\gamma(w)$ is obtained from the most restrictive one.

□

Chapter 4

Tight Immersed Probabilistic Scaling

4.1 Introduction

In the previous chapter, a sample-based methodology to inner approximate the chance constrained set of probability ε (ε-CCS) named *probabilistic scaling* was presented. This approach computes first a simple approximating set, which is then scaled to meet the required probabilistic guarantees. These operations are all performed *offline* and the trade-off between the number of samples required and the tightening of the approximation can be adjusted by the user.

In this chapter, we present the pack-based probabilistic scaling approach and define a novel measure of the tightening of the approximating set. Then, we show how to design the approximating set to meet the required probabilistic guarantees plus a given tightening constraint. This allows the user to control both the complexity and the fitting of the resulting approximating set, in exchange of the number of samples required and the complexity of the approximation problem (which is computed offline). The tightening constraints makes it possible to get larger approximations of the ε-CCS compared to regular probabilistic scaling, while still guaranteeing the inclusion in the original set.

The remainder of the chapter is structured as follows: In Section 4.2 we introduce the problem of approximating the chance constrained set and the numerical example used to compare the different approaches, in Section 4.3 we go through statistical learning theory solutions to the problem and in Section 4.4 we introduce the regular probabilistic scaling approach. In Sections 4.5 and 4.6 we present the tight immersed pack-based probabilistic scaling. In Section 4.8 we compare the performance of the different approaches and finally in Section 6.7 we discuss the main conclusions.

The pack-based probabilistic scaling is inspired by the published paper [74].

4.2 Problem setup

Let the parameters and variables that define the control problem be parameterized by means of the design parameter $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$. Consider the uncertainty vector $w \in \mathcal{W} \subseteq \mathbb{R}^{n_w}$, which represents one admissible realization of the uncertainty with probability distribution $\Pr_{\mathcal{W}}$. Consider general constraints of the form

$$g(\theta, w) \leq 0, \tag{4.1}$$

with $g : \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_w} \to \mathbb{R}$.

The probabilistic nature of the constraints may make them conservative and difficult to impose in a robust fashion, e.g. when the support $\mathcal{W}$ is unbounded [78]. To circumvent this issue, it is possible to allow some violations and relax (4.1) with the following chance constraint formulation:

$$\Pr_{\mathcal{W}}\{g(\theta, w) \leq 0\} \geq 1 - \varepsilon, \tag{4.2}$$

where probability level $\varepsilon \in (0, 1)$ represents the desired bound on the probability of violation of constraint (4.1).

In this chapter, the ε-CCS (defined in section 1.2.1) will be approximated using a sample-based approach. First, we define the multisample $\mathbf{z} \in \mathcal{W}^N$ as a pack of $N \geq 1$ realizations of the uncertainty so that $\mathbf{z} = \{w^{(1)}, \ldots, w^{(N)}\}$, where $\mathcal{W}^N = \mathcal{W} \times \cdots \times \mathcal{W}$ (N times). Its associated probability distribution is denoted as $\Pr_{\mathcal{W}^N}$. It can be observed that $\mathcal{W}^1 = \mathcal{W}$.

Note that given ε, $\mathbb{X}_\varepsilon$ is always a fixed set, whereas sample-based approximations have a random nature, which depend on the random multisample $\mathbf{z} \in \mathcal{W}^N$.

In the next subsection, we present an illustrative example to test the performance of the proposed approximations.

4.2.1 Illustrative example

To illustrate the pack-based probabilistic scaling, we now present the numerical example first proposed in [96]. For a given dimension n_θ, let each individual chance constraint be defined as a half-space tangent to the unit ball at a random point drawn from a uniform distribution, with the origin being always a safe point. Figure 4.1 illustrates an instance of this system for dimension $n_\theta = 2$. Note that the unit ball is the safe region with probability 100% whereas the ε-CCS gets (slightly) larger as ε increases. Besides, we can observe that because of symmetry, the ε-CCS of this system is always a scaled version of the unit ball. As shown in Figure 4.2, the radius of the ε-CCS depends greatly on the dimension on the problem. The exact computation of the ε-CCS radius has been obtained by Monte-Carlo simulation, exploiting the symmetry of the problem.

The problem of approximating the ε-CCS is tackled:

- Without any prior knowledge on the structure of the uncertainty.

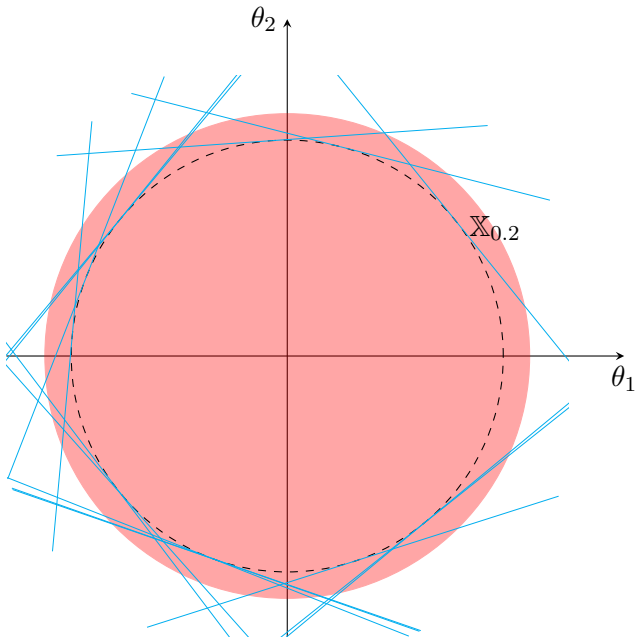


Figure 4.1: Representation of $\mathbb{X}_{0.2}$ (red circle), unit ball (dashed black circle) and random constraint samples (cyan lines).

- With a constraint generator which can generate the random samples.

This chapter provides sample-based approximations to the ε-CCS of tunable complexity which do not require any previous knowledge of the problem, e.g. symmetry. The performance of the proposed approximations is tested against the illustrative example.

4.3 Statistical learning theory

In this section, we show how to use the results from statistic learning theory [108, 49] to obtain an inner approximation of the ε-CCS.

First, we introduce the definition of indicator function.

Definition 4.1 (Indicator function)**.** *The indicator function of constraint* (4.1) $I^g : \Theta \times \mathcal{W} \to \{0, 1\}$ *is defined as*

$$I^g(\theta, w) \doteq \begin{cases} 0 & \text{if } g(\theta, w) \leq 0 \\ 1 & \text{otherwise.} \end{cases}$$

In statistical learning theory, given a multisample $\mathbf{z} \in \mathcal{W}^N$, approximations of the ε-CCS are often calculated by means of a constraint on the empirical mean, i.e.

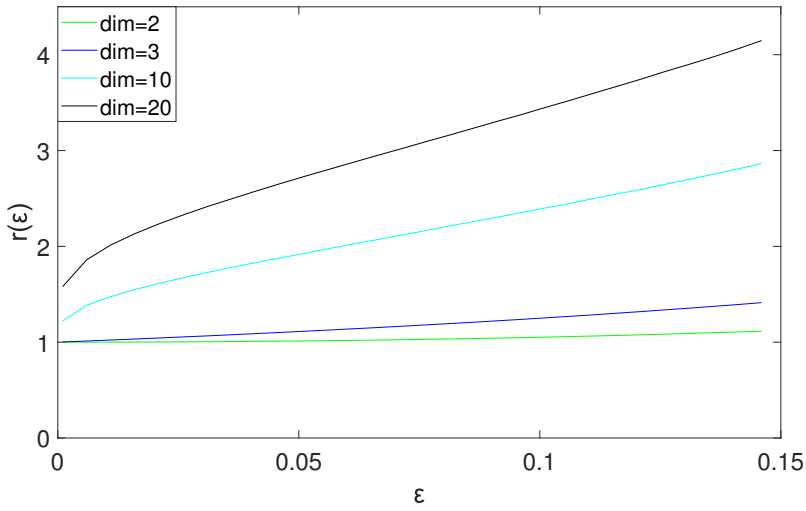


Figure 4.2: Radius of the set $\mathbb{X}_\varepsilon$ as a function of ε for different dimensions.

$$\frac{1}{N}\sum_{i=1}^{N} I^g(\theta, w^{(i)}).$$

Given a discarding parameter r, then $\rho = \frac{r}{N}$ bounds the empirical mean so that the set

$$\Omega_{LT}(\rho, \mathbf{z}) = \{\theta \in \Theta : \frac{1}{N}\sum_{i=1}^{N} I^g(\theta, w^{(i)}) \leq \rho\}$$

constitutes an approximation of $\mathbb{X}_\varepsilon$. Assuming that the indicator function I^g has finite Vapnik-Chervonenkis (VC) dimension [108] and that $\rho < \varepsilon$, the probability that $\Omega_{LT}(\rho, \mathbf{z})$ constitutes an inner approximation of $\mathbb{X}_\varepsilon$, i.e.

$$\mathrm{Pr}_{\mathcal{W}^N}\{\Omega_{LT}(\rho, \mathbf{z}) \subseteq \mathbb{X}_\varepsilon\},$$

converges to 1 as the number of samples N converges to infinity. In [49], sample complexity bounds for N are explicitly calculated, which guarantee that $\Omega_{LT}(\rho, \mathbf{z})$ is included in $\mathbb{X}_\varepsilon$ with a given confidence $\delta \in (0, 1)$, i.e. $\mathrm{Pr}_{\mathcal{W}^N}\{\Omega_{LT}(\rho, \mathbf{z}) \subseteq \mathbb{X}_\varepsilon\} \geq 1 - \delta$. The resulting sample complexity grows linearly with the VC dimension of I^g multiplied by a factor larger than $\frac{1}{\varepsilon}$. However, the resulting approximation is generally non-convex and is often non-connected. This may hinder its practical application and makes it generally unsuitable for real-time problems.

Provided that the indicator function I^g is convex, the particular case $\rho = 0$ always leads to a convex approximation $\Omega_{LT}(0, z)$. However, not allowing any

violation in a sample-based approach makes the result very sensitive to extreme events and therefore usually leads to conservative results or even empty sets [78].

To approximate the ε-CCS and deal with some of the issues present in classical statistical learning theory, we recall the regular probabilistic scaling approach presented in the previous chapter.

4.4 Regular probabilistic scaling

The probabilistic scaling approach consider a simple approximating set (SAS) $c \oplus \Omega_0$ and scale it around a given scaling center c to achieve the desired probabilistic guarantees. The resulting set $\Omega(\bar{\gamma}) = c \oplus \bar{\gamma}\Omega_0$ constitutes an inner approximation of the ε-CCS. Since the final approximating set $\Omega(\bar{\gamma})$ is a linear transformation of the SAS (scaling), its complexity can be tuned by choosing an appropiate SAS. The ability to tune the set complexity beforehand makes probabilistic scaling techniques ideal for a variety of applications in which complexity may be a limiting factor [96, 97].

Given a shape Ω_0 and a scaling center c, the goal of probabilistic scaling is to find the largest scaling factor $\bar{\gamma}$ such that

$$\Pr_{\mathcal{W}}\{c \oplus \bar{\gamma}\Omega_0 \subseteq \mathbb{X}_\varepsilon\} \geq 1 - \delta. \tag{4.3}$$

and therefore, the chance constraint (4.2) is also met with the same probability $1 - \delta$.

Now, we introduce the pack formulation of some of the concepts introduced in previous chapters. This generalization will be used to define both the regular and the pack-based probabilistic scaling.

Definition 4.2 (Pack indicator function)**.** *Given integers s and L such that $0 \leq s < L$ and given a multisample $\mathbf{z} \in \mathcal{W}^L$, the pack indicator function $I_s^g : \Theta \times \mathcal{W}^L \to \{0, 1\}$ is defined as*

$$I_s^g(\theta, \mathbf{z}) \doteq \begin{cases} 0 & \text{if } \sum_{\ell=1}^{L} I^g(\theta, w^{(\ell)}) \leq s \\ 1 & \text{otherwise.} \end{cases} \tag{4.4}$$

$I_s^g(\theta, \mathbf{z})$ indicates whether the point θ violates more than s of the constraints associated to the uncertainty realizations of the multisample $\mathbf{z}$.

Definition 4.3 (Pack safe region)**.** *The pack safe region $\Phi_s^g(\mathbf{z})$ is defined as the set of points which violates no more than s of the constraints associated to the uncertainty realizations of multisample $\mathbf{z}$ and can be expressed as:*

$$\Phi_s^g(\mathbf{z}) \doteq \{\theta \in \Theta : I_s^g(\theta, \mathbf{z}) = 0\}.$$

Definition 4.4 (Pack scaling factor)**.** *Given a scalable SAS $\Omega(\gamma)$ defined by a scaling center $c \in \Theta$ and a shape Ω_0, and a multisample $\mathbf{z} \in \mathcal{W}^L$, we define the pack scaling factor of $\Omega(\gamma)$ relative to the random constraints $g(\theta, w^{(i)}) \leq 0, \forall w^{(i)} \in \mathbf{z}$ as*

$$\gamma^s(c, \Omega_0, \mathbf{z}) \doteq \begin{cases} 0 & \text{if } c \notin \Phi_s^g(\mathbf{z}) \\ \max\limits_{c \oplus \gamma\Omega_0 \subseteq \Phi_s^g(\mathbf{z})} \gamma & \text{otherwise.} \end{cases} \tag{4.5}$$

Now we introduce a property that will be used throughout the chapter:

Property 4.1. *Given the accuracy parameter $p \in (0,1)$ and the confidence level $\delta \in (0,1)$, consider the discarding integer parameter $r \geq 0$ and suppose that M is chosen such that*

$$B(r; M, p) \leq \delta. \tag{4.6}$$

For each pack of constraints $i = 1, \ldots, M$, draw the i.i.d. sets $\mathbf{z}_i \in \mathcal{W}^L$ and define

$$\gamma_i \doteq \gamma^s(c, \Omega_0, \mathbf{z}_i)$$

and suppose that $\gamma \geq 0$ and $\gamma \leq \gamma_{1+r:M} > 0$. Then, with probability no smaller than $1 - \delta$,

$$\Pr_{\mathcal{W}^L}\{c \oplus \gamma\Omega_0 \not\subseteq \Phi_s^g(\mathbf{z})\} \leq p.$$

Proof. The property can be proved particularizing the results of convex scenario [54], [55] to the case of a scalar decision variable.

Consider the following optimization problem

$$\begin{aligned} \max_{\gamma} \quad & \gamma \\ s.t. \quad & c \oplus \gamma\Omega_0 \subseteq \Phi_s^g(\mathbf{z}_i), \quad i = 1, \ldots, M. \end{aligned} \tag{4.7}$$

If this problem has a valid solution, we can rewrite it using the definition of $\gamma^s(\cdot)$ as

$$\begin{aligned} \max_{\gamma} \quad & \gamma \\ s.t. \quad & \gamma \leq \gamma^s(c, \Omega_0, \mathbf{z}_i), \quad i = 1, \ldots, M. \end{aligned} \tag{4.8}$$

It has been proved in [54] and [55] that if one discards no more than r constraints on a convex problem with M random constraints, then the probability of violating the constraints with the solution obtained from the random convex problem is no larger than $p \in (0,1)$, with probability no smaller than $1 - \delta$, where

$$\delta = \binom{d + r - 1}{d - 1} B(d + r - 1; M, p),$$

and d is the number of decision variables.

We first notice that (4.8) is convex and has a unique scalar decision variable γ, i.e. $d = 1$. Also, the non-degeneracy and uniqueness assumption required in the application of the results of [54] and [55] are satisfied. Hence, if we allow r violations in the above minimization problem, we have that with probability no smaller than $1 - \delta$, where

$$\delta = B(r; M, p), \tag{4.9}$$

the optimal solution of problem (4.8) with no more than r constraint removal $\bar{\gamma}$ satisfies

$$\Pr_{\mathcal{W}^L}\{\bar{\gamma} > \gamma^s(c, \Omega_0, \mathbf{z})\} \leq p.$$

We conclude from this and the definition of $\gamma^s(\cdot)$ that with probability no smaller than $1 - \delta$,

$$\Pr_{\mathcal{W}^L}\{c \oplus \bar{\gamma}\Omega_0 \not\subseteq \Phi_s^g(\mathbf{z})\} \leq p.$$

Note that problem (4.8) with constraint removal can be solved directly by ordering the values $\gamma_i = \gamma^s(c, \Omega_0, \mathbf{z}_i)$. It is clear that if $r \geq 0$ violations are allowed, then the optimal value for γ is $\bar{\gamma} = \gamma_{1+r:N}$. Smaller values of γ would meet the inclusion constraint but will not be optimal, while larger values of γ would no longer meet the inclusion constraint.

□

Remark 4.1. *Property 4.1 can be particularized for the regular probabilistic scaling, i.e. $M = N$ and $L = 1$.*

Suppose that N is chosen such that

$$B(r; N, p) \leq \delta.$$

Let $\mathbf{z} \in \mathcal{W}^N$. For $i = 1, \ldots, N$, define

$$\gamma_i \doteq \gamma^r(c, \Omega_0, \mathbf{z}_i)$$

and suppose that $\gamma \geq 0$ and $\gamma \leq \gamma_{1:M} > 0$. Then, with probability no smaller than $1 - \delta$,

$$\Pr_{\mathcal{W}}\{c \oplus \gamma\Omega_0 \not\subseteq \Phi_0^g(w)\} \leq p.$$

Remark 4.2. *Property 4.1 can also be particularized for the case $r = 0$.*

Suppose that M is chosen such that

$$(1 - p)^M \leq \delta.$$

For each pack of constraints $i = 1, \ldots, M$, draw the i.i.d. sets $\mathbf{z}_i \in \Pr_{\mathcal{W}^L}$ and define

$$\gamma_i \doteq \gamma^s(c, \Omega_0, \mathbf{z}_i)$$

and suppose that $\gamma \geq 0$ *and* $\gamma \leq \gamma_{1:M} > 0$. *Then, with probability no smaller than* $1 - \delta$,

$$\Pr_{\mathcal{W}^L}\{c \oplus \gamma\Omega_0 \not\subseteq \Phi_s^g(\mathbf{z})\} \leq p.$$

In the previous chapter, it was shown how to sample the uncertainty and obtain a scaling factor such that, with pre-specified probabilistic levels, the scaled set $c \oplus \bar{\gamma}\Omega_0$ constitutes an inner approximation of the ε-CCS (regular probabilistic scaling). For the sake of completeness, we now recall the aforementioned property along with its correspondent proof using the pack formulation (with $s = 0$).

Property 4.2. *Given accuracy parameter* $\varepsilon \in (0, 1)$, *confidence level* $\delta \in (0, 1)$, *a discarding integer parameter* $r \geq 0$, *and an integer* N_γ *such that*

$$B(r; N_\gamma, \varepsilon) \leq \delta. \tag{4.10}$$

Let $\mathbf{z} \in \mathcal{W}^{N_\gamma}$. *For each constraint* $i = 1, \ldots, N_\gamma$, *define*

$$\gamma_i = \gamma^0(c, \Omega_0, w^{(i)}).$$

Suppose that $\bar{\gamma} = \gamma_{1+r:N_\gamma} > 0$. *Then, with probability no smaller than* $1 - \delta$,

$$c \oplus \bar{\gamma}\Omega_0 \subseteq \mathbb{X}_\varepsilon.$$

Proof. According to Remark 4.1, if we choose the parameters (N_γ, r) according to (4.10), we have that $\bar{\gamma} = \gamma_{1+r:N_\gamma} > 0$ satisfies, with probability no smaller than $1 - \delta$, that

$$\Pr_{\mathcal{W}}\{c \oplus \bar{\gamma}\Omega_0 \not\subseteq \Phi_0^g(w)\} \leq \varepsilon.$$

Equivalently,

$$\Pr_{\mathcal{W}}\{c \oplus \bar{\gamma}\Omega_0 \subseteq \Phi_0^g(w)\} > 1 - \varepsilon.$$

This can be rewritten as

$$\Pr_{\mathcal{W}}\{I_0^g(\theta, w) = 0, \ \ \forall \theta \in c \oplus \bar{\gamma}\Omega_0\} > 1 - \varepsilon.$$

According to the definition of ε-CCS (see (1.2), it can be stated as

$$c \oplus \bar{\gamma}\Omega_0 \subseteq \mathbb{X}_\varepsilon.$$

□

We refer to the action of calculating the scaling factor $\bar{\gamma}$ according to Property 4.2 and using it to create the approximating set $c \oplus \bar{\gamma}\Omega_0$ as regular probabilistic scaling (or simply PS). The choice of the discarding parameter r is up to the user. Recalling the discussion made in section 2.2, large values of r make the resulting set more insensitive to extreme values, at the expense of a larger sample

Table 4.1: Minimum sample size N_γ required to approximate the 0.05-CCS with confidence $\delta = 10^{-5}$ for different values of the discarding parameter r.

r	0	5	10	15
N_γ	225	442	613	769

complexity N_γ. Remember that in probabilistic scaling techniques, convexity of the approximating set is independent of the discarding parameter r and only depends on the choice of the SAS. Table 4.1 shows the minimum number of samples required for different values of the discarding parameter r so that, upon regular probabilistic scaling, the resulting set approximates $\mathbb{X}_{0.05}$ with a confidence of $\delta = 10^{-5}$.

Calculating approximations of the ε-CCS using regular probabilistic scaling is generally easy to compute, does not require any assumption on the underlying probabilities (such as finite VC dimension), provides probabilistic guarantees to the scaled region, and its effectiveness has been proven.

Despite all its advantages, regular probabilistic scaling may still lead to very conservative solutions for some problems. In the illustrative example proposed in section 4.2.1, let Ω_0 be the unit ball and the scaling center c be the origin. Then, the scaling factor obtained by regular probabilistic scaling is independent of ε and the dimension of the problem, and it is always equal to 1, regardless of the discard parameter r used. The reason behind this is that the scaling factors associated to every realization of the constrained set are one, i.e. $\gamma_i = 1$, for all $i \in [N_\gamma]$. Since the constraints are taken into account independently, the act of discarding some of them does not change the result in this problem.

According to the discussion from Section 4.2.1, approximating $\mathbb{X}_\varepsilon$ with the unit ball can be very conservative, especially for high dimensions and large values of ε (Figure 4.2).

In the next section, we outline the pack-based probabilistic scaling. For the same initial SAS, this variant of the regular probabilistic scaling may lead to less conservative results at the expense of (possibly) more demanding computation times.

4.5 Pack-based probabilistic scaling

The main idea behind pack-based probabilistic scaling (PBPS) is to divide the samples in a number of packs and allow some constraint violations inside each of them. As opposed to regular probabilistic scaling, where the scaling factor associated to each constraint is computed independently, in the pack-based approach the constraints inside each pack are taken into account together. Ultimately, this can lead to tighter approximations of the ε-CCS and reduced sample complexity.

Let the N_γ sampled constraints be divided into M packs of L constraints each, i.e. $\mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_M\} = \{w^{(1)}, \ldots, w^{(N_\gamma)}\}$, with $\mathbf{z} \in \mathcal{W}^{N_\gamma}$ and $\mathbf{z}_i \in \mathcal{W}^L$ for

$i \in [M]$. The following property, shows how to determine the scaling factor using a pack-based approach so that the scaled SAS is fully contained in the ε-CCS with a given confidence δ.

Property 4.3. *Consider the shape Ω_0, the scaling center c, accuracy parameter $\varepsilon \in (0,1)$, confidence level $\bar{\delta} \in (0,1)$ and non negative integers M, L and s, with $L > s$ such that*

$$B(s; L, \varepsilon)^M \leq \delta. \tag{4.11}$$

Denote $\mathbf{z}_i \in \mathcal{W}^L$ each pack of constraints, where $i \in [M]$ and define

$$\gamma_i = \gamma^s(c, \Omega_0, \mathbf{z}_i).$$

Suppose that $\gamma \geq 0$ and $\gamma \leq \gamma_{1:M} > 0$, then with probability no smaller than $1-\delta$,

$$c \oplus \gamma\Omega_0 \subseteq \mathbb{X}_\varepsilon.$$

Proof. Let $p = 1 - B(s; L, \varepsilon)$. From Remark 4.2 we know that if M is chosen such that

$$B(s; L, \varepsilon)^M \leq \delta$$

and $\gamma \geq 0$ and $\gamma \leq \gamma_{1:M} > 0$, then with probability no smaller than $1-\delta$ the following holds

$$\Pr_{\mathcal{W}^L}\{c \oplus \gamma\Omega_0 \not\subseteq \Phi_s^g(\mathbf{z})\} \leq p.$$

Equivalently,

$$\Pr_{\mathcal{W}^L}\{I_s^g(\theta, \mathbf{z}) = 1, \ \ \forall \theta \in c \oplus \gamma\Omega_0\} \leq p.$$

We know from Property 4.6 in Appendix that

$$\Pr_{\mathcal{W}^L}\{I_s^g(\theta, \mathbf{z}) = 1\} \leq p \iff \Pr_{\mathcal{W}}\{I^g(\theta, w) = 1\} \leq \varepsilon. \tag{4.12}$$

Therefore, we conclude that

$$\Pr_{\mathcal{W}}\{I^g(\theta, w) = 1\} \leq \varepsilon, \ \forall \theta \in c \oplus \gamma\Omega_0.$$

Equivalently,

$$c \oplus \gamma\Omega_0 \subseteq \mathbb{X}_\varepsilon.$$

□

We refer to the action of calculating the scaling factor $\bar{\gamma} = \gamma_{1:M}$ according to Property 4.3 and using it to create the approximating set $c \oplus \bar{\gamma}\Omega_0$ as pack-based probabilistic scaling (PBPS).

Unlike regular probabilistic scaling, the sample complexity in PBPS is given by two parameters, namely the number of packs M and the size of each pack L. The sample complexity is calculated as $N_\gamma = ML$.

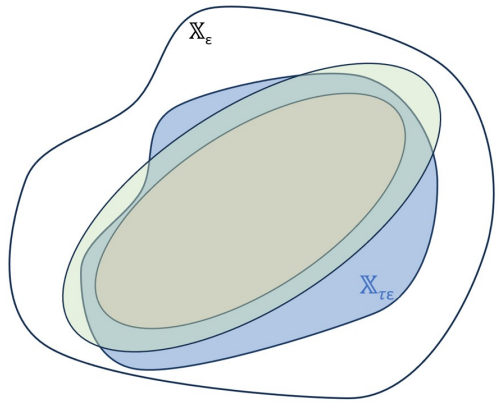


Figure 4.3: Illustration of the concept of tight immersion.

Consequently, condition (4.11) has three tunable parameters: M, L and s. Parameter s is the discarding parameter of each pack. Similar to the discarding parameter of regular probabilistic scaling (r), large values of s makes the approximating set more insensitive to extreme values. As for M and L, one could choose them according to any criterion, e.g. minimize the sample complexity N_γ. In the next section, we propose a tightening constraint to ensure that the approximation obtained through PBPS is close to the ε-CCS.

4.6 Tight immersion

In this section, a measure of tightening of an approximation called tight immersion is presented. This measure is then used along pack-based probabilistic scaling to obtain a tight inner approximation of the ε-CCS.

First, the notion of tight immersion is introduced.

Definition 4.5 (τ-tight immersed)**.** *The set $\mathcal{S}$ is τ-tight immersed in the ε-CCS $\mathbb{X}_\varepsilon$ if*

$$\mathcal{S} \subseteq \mathbb{X}_\varepsilon, \tag{4.13a}$$

$$\mathcal{S} \not\subseteq \mathbb{X}_{\tau\varepsilon} \tag{4.13b}$$

where $\tau \in [0, 1)$ is a measure of tightening.

Tight immersion guarantees not only that the approximation set is *inside* the ε-CCS (4.13a), but also that it will be *not inside* a conservative set characterized by τ (4.13b). Therefore, it imposes a more restrictive condition than the regular inner approximation.

Remark 4.3. *If the ε-CCS is strictly increasing with respect to ε, i.e.*

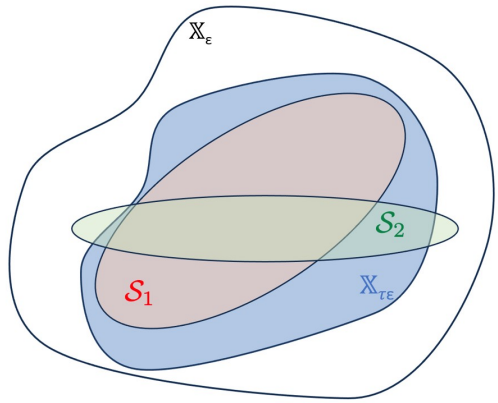


Figure 4.4: Tight immersion alone does not guarantee a better approximation. The red set $\mathcal{S}_1$ is inside $\mathbb{X}_\varepsilon$ but it is not τ-immersed in it. The green set $\mathcal{S}_2$ is both inside $\mathbb{X}_\varepsilon$ and τ-immersed in it.

$$\begin{matrix}\forall \tau_1, \tau_2 \in [0,1) \\ \tau_1 < \tau_2\end{matrix} \implies \mathbb{X}_{\tau_1\varepsilon} \subset \mathbb{X}_{\tau_2\varepsilon},$$

then the larger τ is, the larger $\mathbb{X}_{\tau\varepsilon}$ will be.

Remark 4.4. *For the same geometry, the set with the largest value of τ fits the ε-CCS better (see Figure 4.3). However, tight immersion should never be used to compare the goodness of two different geometries. As shown in Figure 4.4, tight immersion by itself does not imply good approximation.*

Property 4.4. *If the approximating set $\Omega(\bar{\gamma})$ is τ-tight immersed in the set $\mathbb{X}_\varepsilon$, then it is also $\bar{\tau}$-tight immersed in it, with $\bar{\tau} \in [0, \tau)$.*

Proof. From Definition 4.5 we know that condition (4.13a) continues to be met for $\bar{\tau}$, since it does not depend on τ.

We know from $\Omega(\bar{\gamma})$ being τ-tight immersed in the set $\mathbb{X}_\varepsilon$ that

$$\Omega(\bar{\gamma}) \not\subseteq \mathbb{X}_{\tau\varepsilon}.$$

For $\bar{\tau} \in [0, \tau)$, we know $\mathbb{X}_{\bar{\tau}\varepsilon} \subseteq \mathbb{X}_{\tau\varepsilon}$. Consequently

$$\Omega(\bar{\gamma}) \not\subseteq \mathbb{X}_{\bar{\tau}\varepsilon}.$$

Therefore, condition (4.13b) also holds.

□

Notice that equation (4.13a) holds with confidence $1-\delta$ for regular pack-based probabilistic scaling, i.e. if we sample according to the pack parameters from Property 4.3 and carry out pack-based probabilistic scaling. Now, we present a new property that shows how to determine the pack parameters so that, upon

pack-based probabilistic scaling, condition (4.13b) is met with confidence $1-\bar{\delta}$, with $\bar{\delta} \in (0,1)$.

The following property shows how to select the values of M, L and s and calculate $\bar{\gamma}$ so that the tightening constraint (4.13b) is met.

Property 4.5. *Consider the scalable SAS $\Omega(\gamma)$ with shape Ω_0 and scaling center c, accuracy parameter $\varepsilon \in (0,1)$, confidence level $\bar{\delta} \in (0,1)$, tightening parameter $\tau \in [0,1)$, and non negative integers M, L and s, with $L > s$ and*

$$B(s; L, \tau\varepsilon)^M \geq 1 - \bar{\delta}. \tag{4.14}$$

Denote $\mathbf{z}_i \in \mathcal{W}^L$ each pack of constraints, where $i \in [M]$ and define each pack of random constraints as $g(\theta, w^{(j)}) \leq 0, \forall w^{(j)} \in \mathbf{z}_i$. Define also the pack scaling factor of $\Omega(\gamma)$ relative to each pack of random constraints as

$$\gamma_i = \gamma^s(c, \Omega_0, \mathbf{z}_i).$$

Suppose $\gamma \geq \gamma_{1:M} > 0$, then with probability no smaller than $1-\bar{\delta}$,

$$c \oplus \gamma\Omega_0 \not\subseteq \mathbb{X}_{\tau\varepsilon},$$

Proof. Let $p = B(s; L, \tau\varepsilon)$. According to Property 4.8 in Appendix , if we choose the parameters according to (4.14), we have that $\gamma \geq \gamma_{1:M} > 0$ satisfies, with probability no smaller than $1-\bar{\delta}$,

$$\Pr_{\mathcal{W}^L}\{c \oplus \gamma\Omega_0 \subseteq \Phi_s^g(\mathbf{z})\} \leq p.$$

This can be rewritten as

$$\Pr_{\mathcal{W}^L}\{I_s^g(\theta, \mathbf{z}) = 0, \ \ \forall \theta \in c \oplus \gamma\Omega_0\} \leq p.$$

We know from Property 4.7 in Appendix that

$$\Pr_{\mathcal{W}^L}\{I_s^g(\theta, \mathbf{z}) = 0\} \leq p \iff \Pr_{\mathcal{W}}\{I^g(\theta, w) = 0\} \leq 1 - \tau\varepsilon.$$

Therefore, we conclude that

$$\Pr_{\mathcal{W}}\{I^g(\theta, w) = 0\} \leq 1 - \tau\varepsilon, \ \forall \theta \in c \oplus \gamma\Omega_0.$$

Equivalently, with probability no smaller than $1-\bar{\delta}$

$$c \oplus \gamma\Omega_0 \not\subseteq \mathbb{X}_{\tau\varepsilon}.$$

□

Notice that we are using the tightening confidence $1-\bar{\delta}$ instead of the original confidence $1-\delta$. This tightening confidence is user-defined and can be set lower than the original confidence to limit the sample complexity.

Given probabilistic parameters ε, δ and tightening parameters $\tau, \bar{\delta}$, we denote tight immersed pack-based probabilistic scaling (TI-PBPS) to the process of calculating $\bar{\gamma}$ according to Properties 4.3 and 4.5 and using it to obtain the set $c \oplus \bar{\gamma}\Omega_0$. In the next section we show how to sample the constraints to meet both properties.

4.6.1 Design of the pack parameters

In this section, we show how to design the parameters of the pack-based approach to meet tight immersion with confidences δ and $\bar{\delta}$ respectively. From Properties 4.3 and 4.5, we know that conditions (4.13a) and (4.13b) hold when:

$$M \ln B(s; L, \varepsilon) \leq \ln \delta \tag{4.15}$$

$$M \ln B(s; L, \tau\varepsilon) \geq \ln(1 - \bar{\delta}). \tag{4.16}$$

Remember that equation (4.13a) embeds the probabilistic guarantees, whereas (4.13b) is only used to tighten the solution.

From equation (4.15) we know that

$$M \geq \frac{\ln \delta}{\ln B(s; L, \varepsilon)}.$$

Thus, to guarantee (4.15), it suffices to take

$$M = \left\lceil \frac{\ln \delta}{\ln\ B(s; L, \varepsilon)} \right\rceil. \tag{4.17}$$

Equation (4.16) can also be expressed as

$$M \leq \frac{\ln(1 - \bar{\delta})}{\ln B(s; L, \tau\varepsilon)}. \tag{4.18}$$

For a given set of probabilistic and tightening parameters $(\varepsilon, \delta, \tau, \bar{\delta})$, there exist multiple combinations of (M, L, s) that meet (4.17) and (4.18). In particular, we propose two different criteria ζ:

- Minimize the number of possible combinations of $s+1$ constraints, i.e. $\zeta = M\binom{L}{s+1}$.
- Minimize the total sample complexity, i.e. $\zeta = ML$.

The computation of the pack parameters can be formally formulated as the solution of the following optimization problem:

$$\begin{aligned}
(M^o, L^o, s^o) = \underset{M,L,s}{\arg\min} \quad & \zeta \\
\text{s.t.} \quad & M \leq \frac{\ln(1 - \bar{\delta})}{\ln B(s; L, \tau\varepsilon)} \\
& M = \left\lceil \frac{\ln \delta}{\ln B(s; L, \varepsilon)} \right\rceil \\
& L \geq s + 1 \\
& M, L \in \mathbb{N}_{>0} \\
& s \in \mathbb{N}_{\geq 0}.
\end{aligned} \tag{4.19}$$

Because of the complexity of problem (4.19), we propose to use exhaustive search to find a good solution.

Starting from $s = 0$ and going all the way up to $s = 30$, we set M according to (4.17), test the values $L = s + 1, \cdots, s + 300$ and check if the pairs (L, s) satisfy (4.18). Among all the pairs that do satisfy (4.18), we select the one that minimizes ζ.

Table 4.2, shows the pack parameters obtained by solving problem (4.19) with this approach using both ζ criteria, i.e. minimize combinatorics $M\binom{L}{s+1}$ and minimize total sample complexity N_γ.

Table 4.2: Pack parameters, sample complexity and number of possible combinations of TI-PBPS for $\varepsilon = 0.05, \delta = 0.001, \bar{\delta} = 0.1$ and different values of τ minimizing the different criteria.

Criterion: Minimize $M\binom{L}{s+1}$					
τ	M	L	s	N_γ	$M\binom{L}{s+1}$
0.2	43	27	2	1.16e+03	1.25e+05
0.3	155	27	3	4.19e+03	2.72e+06
0.4	2681	20	4	5.36e+04	4.16e+07
0.5	15033	29	6	4.36e+05	2.35e+10

Criterion: Minimize N_γ					
τ	M	L	s	N_γ	$M\binom{L}{s+1}$
0.2	2	195	4	3.90e+02	4.46e+09
0.3	2	303	8	6.06e+02	1.05e+17
0.4	4	278	10	1.11e+03	6.28e+19
0.5	8	309	14	2.47e+03	9.68e+25

4.7 Branch-and-bound based heuristic for PBPS

In this section we present an alternative method to those presented in chapter 3 (Section 3.5) to obtain a simple approximating set (SAS) of the chance constrained set which is guaranteed to be non-empty, even when the robust safe region is empty. Then, we propose a branch-and-bound based heuristic to scale this region with pack-based probabilistic scaling using the parameters obtained in the previous section.

4.7.1 Obtaining an ellipsoidal SAS

Despite not offering any probabilistic guarantees, a good characterization of the SAS allows a tighter approximation of the CCS upon scaling it with (pack-based) probabilistic scaling (see Figure 4.4). If prior knowledge about the geometry of the CCS is known, it could be used to build the SAS.

In this section, we use a sample-based approach to design an ellipsoidal SAS of the form

$$c \oplus \Omega_0 = \{\theta : (\theta - c)^T P(\theta - c) \leq 1\}.$$

The number of random samples N_S used in the computation of the SAS is defined by the user. A larger number of samples may lead to a better characterization of the geometry of the CCS. If the constraints are linear, one could compute the analytic center and its associated ellipsoid, as proposed in [113]. However, this problem is infeasible when the constraints form an empty interior region, even if the chance constrained set is non-empty. The elimination of some restrictive constraints makes the geometry of the initial region more insensitive to extreme values.

We propose the following greedy procedure to remove the most restrictive constraints:

1. Compute the minimum value of margin υ that makes the sampled region non-empty solving an optimization problem of the form

$$\begin{aligned} \upsilon^* = \arg\min_{\upsilon,\theta} \quad & \upsilon \\ \text{s.t.} \quad & A\theta \leq b + \upsilon \end{aligned}$$

2. Remove the most restrictive constraint, which corresponds to the one with the maximum value for $A - b - \upsilon^*$.

3. Repeat the procedure until r_S constraints have been removed. The number of removals r_S is a tuning parameter, but it is reasonable to choose it similar to the number of expected violations, i.e. $r_S \approx \varepsilon N_S$.

After r_S constraints have been removed, the initial region should be non-empty (otherwise one should increase r_S). That means we can now compute the analytic center c and its associated ellipsoid Ω_0 by means of an infeasible Newton start method as shown in [121, 122].

The proposed approach does not rely on any information of the ε-CCS. This makes it suitable for any problem, at the expense of not taking advantage of that information when it is available.

4.7.2 Computing the scaling factor

Now we show how to compute the pack-based scaling factor so that the scaled SAS constitutes a tight inner approximation of the ε-CCS.

Given the SAS and the pack parameters M, L, s computed in the previous sections, we propose a branch-and-bound strategy to address the pack-based probabilistic scaling for the case of linear constraints. This strategy lies in the fact that every point in the scaled SAS must violate at most s constraints (Section

4.5), therefore, the pack scaling factor of each pack i can be computed at the contact point θ_i^* between the scaled SAS and the border of $\Phi_s^g(\mathbf{z}_i)$, i.e. the set of points that violate exactly $s+1$ constraints. Because of the non-convexity of $\Phi_s^g(\mathbf{z}_i)$, approaching the previous problem in a direct fashion can be cumbersome. We know from Theorem 4.2 that if the dimension of the problem is n_θ and we consider s constraint violations, the contact point θ_i^* can always be expressed as the intersection of at least one single constraint and at most $\min(\{s+1, n_\theta\})$ constraints, known as the set of active constraints.

The set of active constraints is computed with a expansion procedure, which we introduce in the following definition.

Definition 4.6. *The expanding procedure consists on adding a new active constraint to a given set of active constraints.*

Now, we show how to efficiently solve the scaling problem given a set of active constraints and present a ray based methodology to upper bound the scaling factor. Then, the full heuristic will be detailed, in which the expanding procedure is used to compute the optimal set of active constraints.

Scaling

Given a set of active constraints $A\theta = B$, in the scaling step we compute the minimum scaling factor so that the scaled SAS $(c \oplus \bar{\gamma}\Omega_0)$ touches them. To do it, we solve the following optimization problem

$$\begin{aligned} \theta^* = \arg\min_{\theta} \quad & \frac{1}{2}(\theta - c)^T P(\theta - c) \\ \text{s.t.} \quad & A\theta = B. \end{aligned} \tag{4.20}$$

In the expanding procedure (see definition 4.6), every new instance of the scaling problem is built adding one new constraint $\tilde{a}_j^T\theta \leq \tilde{b}_j$ to the given set of active constraints $A\theta \leq B$. This is expressed as:

$$\begin{aligned} \theta^* = \arg\min_{\theta} \quad & \frac{1}{2}(\theta - c)^T P(\theta - c) \\ \text{s.t.} \quad & A\theta = B \\ & \tilde{a}_j^T\theta \leq \tilde{b}_j. \end{aligned} \tag{4.21}$$

We can rewrite (4.21) to take advantage of the optimal solution of (4.20), which is obtained beforehand. As a result, each instance of the problem can be solved faster, especially when the dimension of the problem is high.

The dual problem of (4.21) is defined as:

$$\max_{\eta,\lambda_j} \min_{\theta} \quad \frac{1}{2}(\theta - c)^T P(\theta - c) + \eta^T(A\theta - B) + \lambda_j^T(\tilde{a}_j^T\theta - \tilde{b}_j), \tag{4.22}$$

and its solution is determined by:

$$\begin{bmatrix} P & A^T & \tilde{a}_j \\ A & 0 & 0 \\ \tilde{a}_j^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta^* \\ \eta^* \\ \lambda_j^* \end{bmatrix} = \begin{bmatrix} Pc \\ B \\ \tilde{b}_j \end{bmatrix}. \tag{4.23}$$

From the first two rows we have that:

$$\begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \theta^* \\ \eta^* \end{bmatrix} + \begin{bmatrix} \tilde{a}_j \\ 0 \end{bmatrix} \lambda_j^* = \begin{bmatrix} Pc \\ B \end{bmatrix}$$

$$\begin{bmatrix} \theta^* \\ \eta^* \end{bmatrix} = \begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix}^{-1} \left(\begin{bmatrix} Pc \\ B \end{bmatrix} - \begin{bmatrix} \tilde{a}_j \\ 0 \end{bmatrix} \lambda_j^* \right) \tag{4.24}$$

Substituting this into the third row, we have:

$$\begin{bmatrix} \tilde{a}_j^T & 0 \end{bmatrix} \begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix}^{-1} \left(\begin{bmatrix} Pc \\ B \end{bmatrix} - \begin{bmatrix} \tilde{a}_j \\ 0 \end{bmatrix} \lambda_j^* \right) = \tilde{b}_j.$$

We observe that the dual variable λ_j^* for the set expanded with constraint j can be written as:

$$\lambda_j^* = \frac{\begin{bmatrix} \tilde{a}_j^T & 0 \end{bmatrix} \begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix}^{-1} \begin{bmatrix} Pc \\ B \end{bmatrix} - \tilde{b}}{\begin{bmatrix} \tilde{a}_j^T & 0 \end{bmatrix} \begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix}^{-1} \begin{bmatrix} \tilde{a}_j \\ 0 \end{bmatrix}} \tag{4.25}$$

Quick Note

The following two expressions are equivalent and can be used to compute the P-norm of multiple vectors expressed as the rows of X at the same time:

$$\gamma = \operatorname{diag}(XPX^T)$$
$$\gamma = ((XP).^*X)\mathbf{1}_{n_P},$$

where n_P is the dimension of the square matrix P and $.^*$ stands for the element by element multiplication.

Denote $G = \begin{bmatrix} G_{1,1} & G_{1,2} \\ G_{2,1} & G_{2,2} \end{bmatrix} = \begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix}^{-1}$ and $G^{\text{ext}} = G_{1,1}Pc + G_{1,2}B$, where $G_{1,1} \in \mathbb{R}^{n_\theta \times n_\theta}$ and $G_{1,2} \in \mathbb{R}^{n_\theta \times m}$. Let all the $\tilde{m}$ new constraints in the expansion procedure be denoted by:

$$\tilde{a}_j^T \theta = \tilde{b}_j,\ j = 1, \ldots, \tilde{m}, \tag{4.26}$$

and consider they are collected in the matrices $\tilde{A}$ and $\tilde{B}$, such that:

$$\tilde{A} = \begin{bmatrix} \tilde{a}_1^T \\ \vdots \\ \tilde{a}_{\tilde{m}}^T \end{bmatrix}, \qquad \tilde{B} = \begin{bmatrix} \tilde{b}_1 \\ \vdots \\ \tilde{b}_{\tilde{m}} \end{bmatrix}. \tag{4.27}$$

Then, one could compute at the same step the dual variable for each one of the new constraints $\lambda_j^*,\ j = 1, \ldots, \tilde{m}$ in the expansion as:

$$\Lambda^* = \begin{bmatrix} \lambda_1^* & \ldots & \lambda_{\tilde{m}}^* \end{bmatrix}^T = (\tilde{A}G^{\text{ext}} - \tilde{B})./\left(((\tilde{A}G_{1,1}).^*\tilde{A})\mathbf{1}_{n_\theta}\right), \tag{4.28}$$

where $G^{\text{ext}} = G_{1,1}Pc + G_{1,2}B$ and $\mathbf{1}_{n_\theta}$ corresponds to a column vector of ones of dimension n_θ. From (4.24) and (4.28), we can compute the contact points associated to all the new sets created in the expansion procedure:

$$\begin{aligned} \Theta^* = \begin{bmatrix} \theta_1^* & \ldots & \theta_{\tilde{m}}^* \end{bmatrix} = G_{1,1}(Pc - \tilde{A}^T.^*(\Lambda^*)^T) + G_{1,2}B = \\ = G^{\text{ext}} - (G_{1,1}\tilde{A}^T).^*(\Lambda^*)^T. \end{aligned} \tag{4.29}$$

Finally, we can compute the scaling factors associated to all the new sets created in the expansion procedure:

$$\Gamma^* = \begin{bmatrix} \gamma_1^* & \ldots & \gamma_{\tilde{m}}^* \end{bmatrix}^T = ((\Theta^* - c)^T P).^*(\Theta^* - c)^T \mathbf{1}_{n_\theta}. \tag{4.30}$$

Ray search

The ray based methodology provides upper bounds for the scaling factor γ which will be used as the stop condition for the heuristic.

Given a point θ^* (e.g. the intersection point of a number of constraints), we throw a ray that starts at the center of the SAS c and passes through θ^*. Then, we calculate the first point where the ray violates more than s constraints. The scaling factor γ^{ub} associated to that point is used as an upper bound of γ.

Let the ray be given by $\theta = c + \alpha v_d$, where v_d denotes its direction. Then, the intersections between the ray and the constraints are defined by:

$$\begin{aligned} Ac + \alpha A v_d &= B \\ \alpha &= \frac{B - Ac}{Av_d}. \end{aligned}$$

Since the ray has a sense, we take the positive values of α, partial sort them and choose the s-th smallest one α_s. Then, we calculate the minimum factor by which we need to scale our geometry to include the point $\theta = c + \alpha_s v_d$. That value is an upper bound of the scaling factors. This single ray search procedure is depicted in Figure 4.5.

In the context of an expansion procedure, multiple contact points are identified (equation (4.29)), so we can perform multiple ray searches at the same time.

All the unit vectors associated to the contact points Θ^* can be computed as:

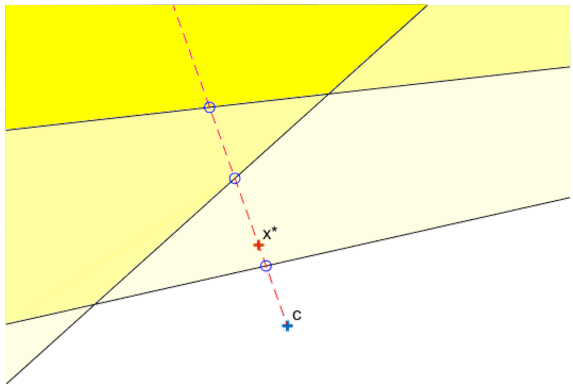


Figure 4.5: Ray search scheme given the SAS center c and the point x^*. Inequality constraints are shown as black lines and the yellow region represents their violation. More intense yellow indicates that a larger number of constraints are violated. The ray is represented by the dashed red line. Blue circles highlights the points of the ray at which the number of constraint violations changes.

$$V_d = (\Theta^* - c)./\sqrt{((\Theta^* - c).*(\Theta^* - c))^T \mathbf{1}_{n_\theta}}. \tag{4.31}$$

All the intersections between the rays and the constraints are defined by:

$$\bar{\alpha} = (\tilde{B} - \tilde{A}c)./(\tilde{A}V_d), \tag{4.32}$$

where each column represents each of the rays and each row represents each of the constraints. Since no more than s constraint violations are allowed, we partial sort the columns of $\bar{\alpha}$ and create the vector $\bar{\alpha}_s$ from their s-th smallest values. Then, we compute the contact points for all the rays as:

$$\Theta_s = c + \bar{\alpha}_s.^* V_d. \tag{4.33}$$

The associated scaling factors are computed as:

$$\Gamma_s = ((\Theta_s - c)^T P).^*(\Theta_s - c)^T \mathbf{1}_{n_\theta}. \tag{4.34}$$

Therefore, an upper bound of the algorithm is obtained as:

$$\gamma^{\text{ub}} = \min(\Gamma_s). \tag{4.35}$$

4.7.3 Heuristic

Once we have detailed the computation of the pack parameters, the scaling problem and the ray search, we now present the branch-and-bound heuristic used to compute the pack-based probabilistic scaling and achieve tight immersion for the case of linear constraints and ellipsoidal SAS.

Algorithm 4 details the branch-and-bound based heuristic used to compute the tight immersed pack-based probabilistic scaling (TI-PBPS) introduced in Section 4.6 given the parameters M, L, s that satisfy the tight immersion constraints (4.15)-(4.16).

Algorithm 4 Branch-and-bound based heuristic for computing a probabilistic inner approximation using pack-based probabilistic scaling.

For each j-th pack, with $j \in [1, M]$:

Init

1) Set up L sets of constraints $\mathbb{C}_k$, each containing one different individual constraint from the pack j.

2) For each set $\mathbb{C}_k$, compute $\gamma_{\mathbb{C}_k}$ and $x^*_{\mathbb{C}_k}$ according to Scaling (4.28)-(4.30).

Loop

3) Among all the available sets, select $\mathbb{C}_{\min}$ as the most restrictive one, i.e. the set with the smallest value of γ ($\gamma_{\mathbb{C}_{\min}}$), and identify its contact point $\theta^*_{\mathbb{C}_{\min}}$.

4) Compute a ray search using the point $\theta^*_{\mathbb{C}_{\min}}$ and the L constraints from pack j according to (4.31)-(4.35). Then make γ^{ub} equal to the smallest known upper bound.

5) IF $\gamma_{\mathbb{C}_{\min}} = \gamma^{\text{ub}}$, THEN end the loop of pack j and return $\gamma_{\mathbb{C}_{\min}}$ as its scaling factor.

6) Set up all the possible different sets $\mathbb{C}_k$ that have the same constraints as $\mathbb{C}_{\min}$ plus one additional new constraint from the pack j, then remove the set $\mathbb{C}_{\min}$.

7) For the sets of constraints set up in step 4, compute $\gamma_{\mathbb{C}_k}$ and $\theta^*_{\mathbb{C}_k}$ according to Scaling (4.28)-(4.30).

End Loop
End For

8) Given the scaling factor for each pack, use the smallest one $\bar{\gamma}$ to scale the SAS as: $c \oplus \bar{\gamma}\Omega_0$.

To visualize the heuristic described in Algorithm 4, we apply it to the simple problem shown in Figure 4.6, where the four constraints represented as lines are indeed closed half-spaces containing Ω_0 in its safe region. The aim is to scale the initial set Ω_0 such that the resulting geometry is as large as possible and every point in it violates no more than one constraint ($s = 1$).

The evolution of the branch-and-bound tree for this simple problem is shown in Figures 4.7-4.9. White nodes indicate expandable sets of active constraints, red

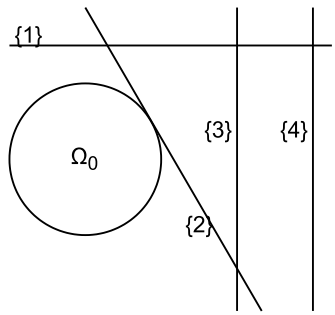


Figure 4.6: Initial SAS (Ω_0) and a set of four constraints labelled from {1} to {4}.

ones indicate the ones with a scaling factor greater than the upper bound, and green ones indicate that the set can be no longer expanded and the scaling factor is no larger than the upper bound.

In step 5) of Algorithm 4, exiting the loop before reaching the optimal solution, i.e. $\gamma_{\mathbb{C}_{\min}} < \gamma^{\text{ub}}$, would result in a scaled SAS that is a probabilistic inner approximation of the ε-CCS but may not be τ-tight immersed in it. This stems from the fact that the tight immersion constraints (4.13) are guaranteed only for the final (optimal) result. While at any iteration $\gamma_{\mathbb{C}_{\min}}$ makes the scaled set violate at most s pack constraints, the tight immersion constraint also requires that at least one point of the scaled set violates at least s pack constraints, which would not be guaranteed in early exit scenarios.

Algorithm 4 relies on the expanding procedure, which is used to enlarge the scaled SAS. This property of the expanding procedure is formally stated in the following theorem.

Theorem 4.1. *Expanding the set of active constraints cannot decrease the scaling factor.*

Proof Given a set of active constraints defined by the general function $h_1(\theta) = 0$, the scaling factor associated to them is computed as:

$$\begin{aligned} \min_{\theta,\gamma} \quad & \gamma \\ \text{s.t.} \quad & \theta \in c \oplus \gamma\Omega_0 \\ & h_1(\theta) = 0 \end{aligned} \tag{4.36}$$

If one were to expand this set (See Expanding Procedure in Definition 4.6) with a new active constraint $h_2(\theta) = 0$, the associated scaling problem would transform into:

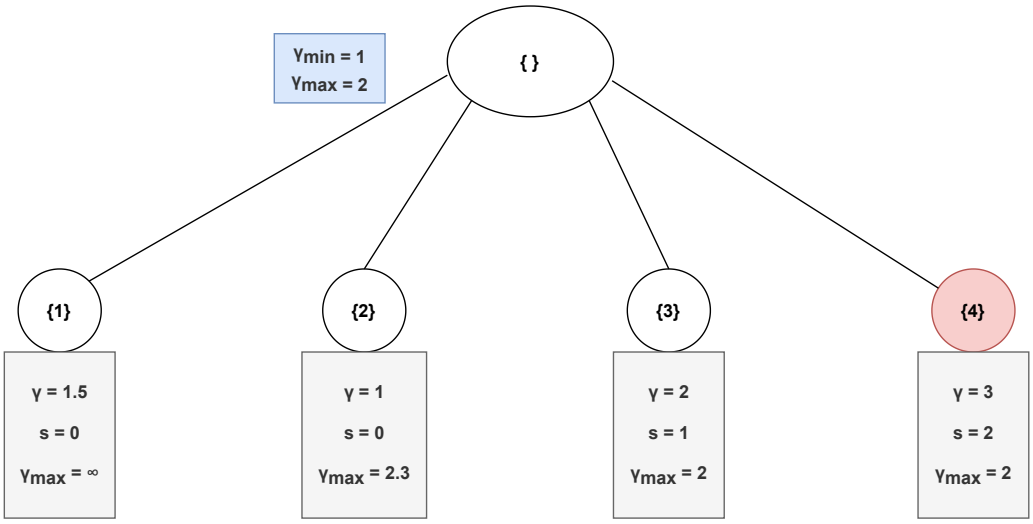


Figure 4.7: Evolution of γ and its upper bound $\gamma_{\max}$ in the branch-and-bound tree. Phase 1.

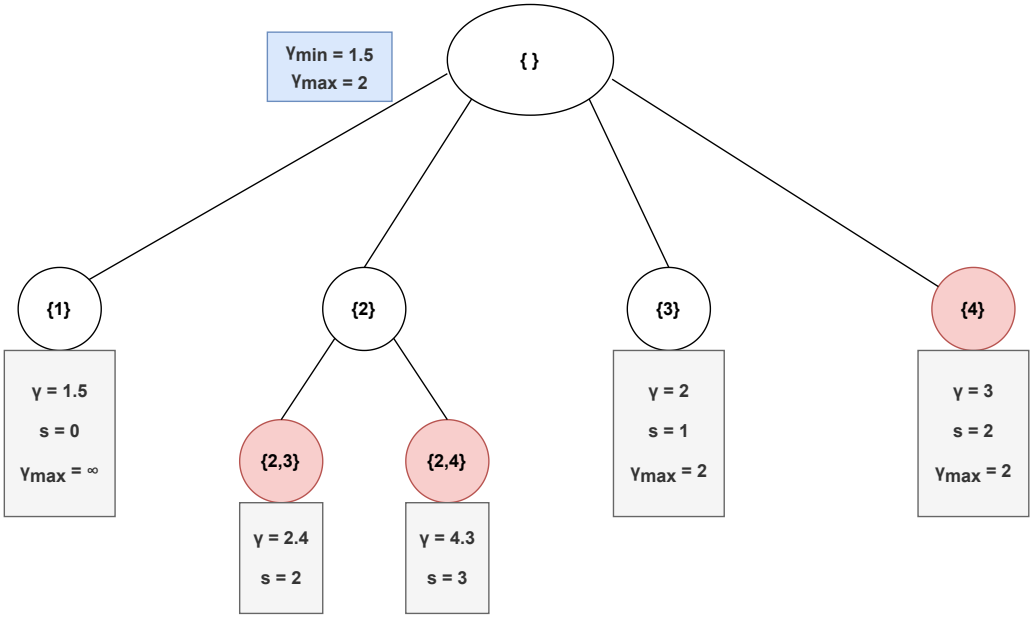


Figure 4.8: Evolution of γ and its upper bound $\gamma_{\max}$ in the branch-and-bound tree. Phase 2.

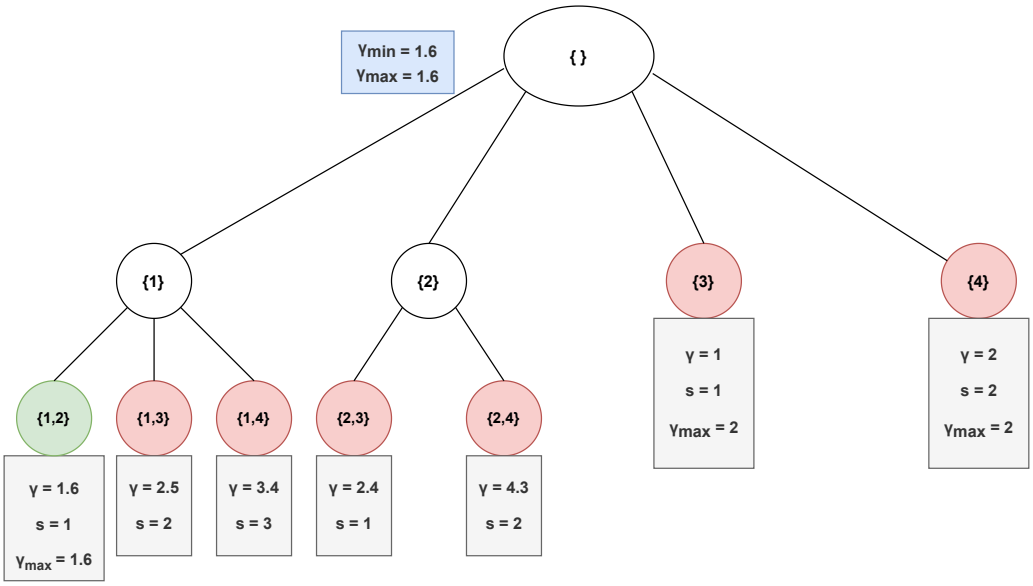


Figure 4.9: Evolution of γ and its upper bound $\gamma_{\max}$ in the branch-and-bound tree. Phase 3 (final).

$$\begin{aligned} \min_{\theta,\gamma} \quad & \gamma \\ \text{s.t.} \quad & \theta \in c \oplus \gamma\Omega_0 \\ & h_1(\theta) = 0 \\ & h_2(\theta) = 0. \end{aligned} \tag{4.37}$$

It is clear that problem (4.37) is more constrained than (4.36). Therefore, the optimal scaling factor obtained from (4.36) is never larger than that of (4.37).

□

According to Theorem 4.1, if the set with the smallest value of γ ($\gamma_{\mathbb{C}_{\min}}$) in step 3) of Algorithm 4 has an associated contact point that violates no more than s constraints, then its scaling factor constitutes a lower bound of the branch-and-bound tree. The upper bounds of the tree are computed using the ray search procedure described in Section 4.7.2.

Note that, since we are only interested in the smallest scaling factor, we can use the upper bounds from previously computed packs in the step 4 of Algorithm 4. This can speed up the heuristic, since upper bounds do not need to be computed from the ground up for each pack.

4.8 Results

In this section, we use the illustrative example presented in Section 4.2.1 and calculate approximations of its 0.05-CCS using both regular probabilistic scaling and TI-PBPS with different values of the tightening parameter τ. We use the unit ball as the SAS Ω_0 and the origin as the scaling center c. Therefore, both the resulting approximating sets and the 0.05-CCS are scaled versions of the unit ball.

Table 4.3 and Figure 4.10 show the radii of the approximation obtained for regular probabilistic scaling (PS) and tight immersed pack-based probabilistic scaling (TI-PBPS) presented in this chapter. These radii are compared with the radius of the real 0.05-CCS, which has been computed by means of a Monte-Carlo simulation.

To reduce variability, the radii of the TI-PBPS results corresponds to the median radius of five experiments, each of which containing different realizations of the constraints.

Table 4.3: Comparison of the radius of the approximation set for different dimensions n_θ of the problem given by the different approaches with $\varepsilon = 0.05$, $\delta = 0.001$, $\bar{\delta} = 0.1$.

Real Radii	$n_\theta = 10$	$n_\theta = 20$	$n_\theta = 30$
PS	1	1	1
TI-PBPS $\tau = 0.2$	1.04	1.13	1.23
TI-PBPS $\tau = 0.3$	1.17	1.36	1.48
TI-PBPS $\tau = 0.4$	1.25	1.50	1.72
TI-PBPS $\tau = 0.5$	1.41	1.80	2.15
Real	1.92	2.71	3.32

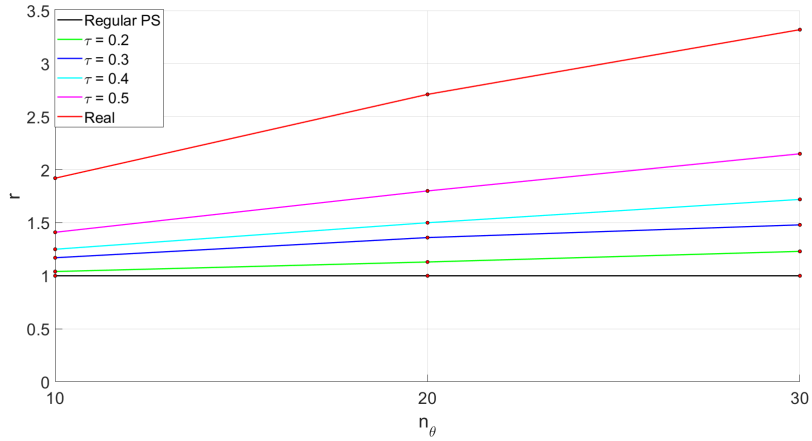


Figure 4.10: Comparison of the radius (r) of the approximation set for different dimensions (n_θ) of the numerical example obtained by the different approaches with $\varepsilon = 0.05$, $\delta = 0.001$, $\bar{\delta} = 0.1$.

In Figure 4.10, we can observe how for this particular problem, the tight immersed pack-based probabilistic scaling is able to substantially improve the result from regular probabilistic scaling, offering a radius more similar to the 0.05-CCS represented by the dashed red line. As expected, the tightening of the new approach improves as the tightening parameter τ increases. As shown in Table 4.2, the tightening improvement comes at the expense of a larger sample complexity and number of possible combinations, which turns into a higher computational cost. It is because of this high computational cost that larger values of τ have not been tested in this thesis.

4.9 Conclusions

In this chapter we have presented the pack-based probabilistic scaling (PBPS) approach to compute sample-based approximations of a chance constrained set. Similar to regular probabilistic scaling, PBPS allows the user to first choose any set and then apply a linear transformation to obtain an approximation of the safe region which meets the given probabilistic guarantees. As a result, complexity of the approximation is tuned a priori. The novel pack-based approach arranges the constraints in packs, which allows to decrease the required number of samples to achieve the same probabilistic guarantees.

We have also introduced a measure of tightening of the approximation of the probabilistic safe region called tight immersion, which allows to design a tightening constraint with which the fitting of the approximation is tuned. This presents a trade-off, since tighter approximations come at the expense of a higher sample complexity and a more intricate procedure to compute the approximation.

To solve the pack-based probabilistic scaling problem, we have presented a branch-and-bound heuristic. It computes the (exact) optimal solution for the pack-based scaling problem and can improve the computational times with respect to an exhaustive search.

Finally, the tight immersed pack-based probabilistic scaling has been tested against an academic numerical example. The proposed approach is able to compute approximations with moderate values of the tightening parameter, and it shows how it can vastly improve the tightening of the approximation with respect to regular probabilistic scaling.

Future research may study how to extend the proposed heuristic to nonlinear setups.

4.10 Appendix

4.10.1 Property 4.6

Property 4.6. *Consider the integer parameters $L > s \geq 0$, the multisample $\mathbf{z} \in \mathcal{W}^L$ and the sample $w \in \mathcal{W}$ and the probability parameter $\varepsilon \in (0,1)$, then, for any constraint $g(\cdot)$,*

$$\Pr_{\mathcal{W}^L}\{I_s^g(\theta,\mathbf{z})=1\} \leq 1 - B(s;L,\varepsilon) \iff \Pr_{\mathcal{W}}\{I^g(\theta,w)=1\} \leq \varepsilon. \tag{4.38}$$

Proof. Let $E(\theta) = \Pr_{\mathcal{W}}\{I^g(\theta,w)=1\}$, then

$$\Pr_{\mathcal{W}^L}\{I_s^g(\theta,\mathbf{z})=0\} = \sum_{i=0}^{s} \binom{L}{i} E(\theta)^i (1-E(\theta))^{L-i} = B(s;L,E(\theta)). \tag{4.39}$$

Denote $q = 1 - B(s;L,\varepsilon)$. Since $B(s;L,\varepsilon)$ is strictly decreasing with respect of ε (Property 4 of [78]), we have

$$B(s;L,E(\theta)) \geq B(s;L,\varepsilon) = 1-q \iff E(\theta) \leq \varepsilon. \tag{4.40}$$

Equivalently,

$$1 - B(s;L,E(\theta)) \leq 1 - B(s;L,\varepsilon) = q \iff E(\theta) \leq \varepsilon. \tag{4.41}$$

Therefore,

$$\Pr_{\mathcal{W}^L}\{I_s^g(\theta,\mathbf{z})=1\} \leq q \iff \Pr_{\mathcal{W}}\{I^g(\theta,w)=1\} \leq \varepsilon. \tag{4.42}$$

□

4.10.2 Property 4.7

Property 4.7. *Consider the integer parameters $L > s \geq 0$, the multisample $\mathbf{z} \in \mathcal{W}^L$ and the sample $w \in \mathcal{W}$ and the probability parameter $\varepsilon \in (0,1)$, then, for any constraint $g(\cdot)$*

$$\Pr_{\mathcal{W}^L}\{I_s^g(\theta,\mathbf{z})=0\} \leq B(s;L,\tau\varepsilon) \iff \Pr_{\mathcal{W}}\{I^g(\theta,w)=0\} \leq 1-\tau\varepsilon. \tag{4.43}$$

Proof. Let $E(\theta) = \Pr_{\mathcal{W}}\{I^g(\theta,w)=1\}$, then

$$\Pr_{\mathcal{W}^L}\{I_s^g(\theta,\mathbf{z})=0\} = \sum_{i=0}^{s}\binom{L}{i}E(\theta)^i(1-E(\theta))^{L-i} = B(s;L,E(\theta)). \tag{4.44}$$

Since $B(s;L,\tau\varepsilon)$ is strictly decreasing with respect of $\tau\varepsilon$ (Property 4 of [78]), we have

$$\begin{aligned}&\Pr_{\mathcal{W}^L}\{I_s^g(\theta,\mathbf{z})=0\} = B(s;L,E(\theta)) \leq B(s;L,\tau\varepsilon) \iff\\ &\Pr_{\mathcal{W}}\{I^g(\theta,w)=1\} = E(\theta) \geq \tau\varepsilon.\end{aligned} \tag{4.45}$$

Therefore,

$$\Pr_{\mathcal{W}^L}\{I_s^g(\theta,\mathbf{z})=0\} \leq B(s;L,\tau\varepsilon) \iff \Pr_{\mathcal{W}}\{I^g(\theta,w)=0\} \leq 1-\tau\varepsilon. \tag{4.46}$$

□

4.10.3 Property 4.8

Property 4.8. *Given the accuracy parameter $p \in (0,1)$ and the confidence level $\bar{\delta} \in (0,1)$, suppose that M is chosen such that*

$$1-p^M \leq \bar{\delta},$$

For each pack of constraints $i = 1,\ldots,M$, draw the i.i.d. sets $\mathbf{z}_i \in \Pr_{\mathcal{W}^L}$ and define

$$\gamma_i \doteq \gamma^s(c,\Omega_0,\mathbf{z}_i)$$

and suppose that $\gamma \geq \gamma_{1:M} > 0$. Then, with probability no smaller than $1-\bar{\delta}$,

$$\Pr_{\mathcal{W}^L}\{c \oplus \gamma\Omega_0 \subseteq \Phi_s^g(\mathbf{z})\} \leq p.$$

Proof. The proof is similar to that of Property 4.1.

Consider the following optimization problem:

$$\begin{aligned}\min_{\gamma}\quad & \gamma\\ s.t.\quad & c \oplus \gamma\Omega_0 \not\subseteq \Phi_s^g(\mathbf{z}_i),\ \ i=1,\ldots,M.\end{aligned} \tag{4.47}$$

From the definition of $\gamma^s(\cdot)$, if problem (4.47) has a feasible solution, we can rewrite it as

$$\begin{aligned} \min_{\gamma} \quad & \gamma \\ s.t. \quad & \gamma > \gamma^s(c, \Omega_0, \mathbf{z}_i), \quad i = 1, \ldots, M. \end{aligned} \tag{4.48}$$

As seen in [54] and [55], if one discards no more than $M-1$ constraints and the number of decision variables is one, then the probability of violating the constraints with the solution obtained from the random convex problem (4.47) is no larger than $p \in (0,1)$, with probability no smaller than $1-\bar{\delta}$, where

$$\begin{aligned} \bar{\delta} &= \binom{M-1}{M-1} B(M-1; M, p) \\ &= \sum_{i=0}^{M-1} \binom{M}{i} p^i (1-p)^{M-i} \\ &= 1 - \sum_{M}^{i=M} \binom{M}{i} p^i (1-p)^{M-i} \\ &= 1 - p^M. \end{aligned}$$

Let $\bar{\gamma}$ be the optimal solution of problem (4.48) with no more than $M-1$ constraint removal. Then

$$\Pr_{\mathcal{W}^L}\{\bar{\gamma} \leq \gamma(c, \Omega_0, \mathbf{z})\} \leq p.$$

We conclude from this, and the definition of $\gamma^s(\cdot)$, that with probability no smaller than $1-\bar{\delta}$,

$$\Pr_{\mathcal{W}^L}\{c \oplus \bar{\gamma}\Omega_0 \subseteq \Phi_s^g(\mathbf{z})\} \leq p.$$

The optimization problem under consideration (4.47) can be solved directly by ordering the values $\gamma_i = \gamma^s(c, \Omega_0, \mathbf{z})$. It is clear that if $M-1$ violations are allowed, then the optimal value for γ is $\bar{\gamma} = \gamma_{1:M}$. Smaller values of γ would no longer meet the exclusion constraint, while larger values would still meet it. □

4.10.4 Theorem 4.2

Theorem 4.2. *The optimal scaling factor for any given pack can be computed at the intersection of at most* $\min(\{s+1, n_\theta\})$ *active constraints.*

Proof. First we prove that the optimal scaling factor for any given pack can be computed at the intersection of active constraints.

Let all the L constraints in a given pack be expressed by the linear expression $A_k(\theta) \leq b_i,\ k = 1, 2, \cdots, L$. The optimal scaling factor obtained from the pack based methodology guarantees that the scaled SAS is as large as possible while none of the points in it violate more than s constraints of that pack.

Let T be a subset of constraint indexes, i.e. $T \subseteq \{1, 2, \ldots, L\}$, then, the optimal scaling factor corresponds to the solution of the following optimization problem:

$$\begin{aligned} \gamma^* = \min_{x,\gamma,T} \quad & \gamma \\ \text{s.t.} \quad & x \in c \oplus \gamma\Omega_0 \\ & A_i(\theta) > b_i, \quad \forall i \in T \\ & A_j(\theta) \leq b_j, \quad \forall j \notin T \\ & \text{card}(T) \leq s, \end{aligned} \tag{4.49}$$

where $\text{card}(T)$ refers to the cardinality of the set T.

Since the constraints are linear, the optimal solution of (4.49) lies in the intersection of active constraints.

Any point in an $\mathbb{R}^{n_\theta}$ space can be expressed as the intersection of n_θ surfaces of dimension $\mathbb{R}^{n_\theta - 1}$. Therefore, n_θ constitutes an upper bound to the number of active constraints to consider.

Constraints being active at one point implies that the number of constraint violations changes in the infinitesimal neighborhood of that point. This change depends on the number of active constraints. Thus, if s constraints are active, the difference on the number constraints violations in this neighborhood is at most s. Since only s constraint violations are allowed, the maximum number of active constraints to consider is $s+1$, where the number of violations could change from 0 to $s+1$ in the infinitesimal neighborhood (there might be more active constraints, but they would be redundant). Therefore, $s+1$ constitutes another upper bound to the number of active constraints to consider.

Part II

Real-Time Optimization under Plant-Model Mismatch

Chapter 5

Periodic Modifier-Adaptation

5.1 Introduction

Economic optimization plays a major role in most industries, since it allows to optimize the performance of the real plant operation [123]. To achieve optimal performance, optimization problems leverage system data and models to compute the trajectory that minimizes the economic cost. To arrange the system information into an optimization framework, multidisciplinary teams are often involved. They must have deep knowledge about the real systems and be able to build detailed models that mirror their behaviour. The complexity and possible change over time of real systems (e.g. due to deterioration) make the identification task expensive and prone to errors, which leads to plant-model mismatch and ultimately may lead to a loss of performance in the controlled system.

In two-layer control schemes (see section 1.2.2), the economic optimization is splitted in the real-time optimization layer (RTO), which computes the optimal steady behaviour, and the advanced control layer, which calculates the inputs required to take the system to that reference. To transform the optimal reference computed by the RTO into a valid reference to the advance controller, often an intermediate layer known as the steady-state target optimization (SSTO) [124] is used.

One of the strengths of two-layer control schemes is their ability to use different models with different time scales for the different layers, being the model from the RTO layer usually more complex and global, while the one from the advanced control layer is generally faster and able to quickly react to disturbances. This allows to keep the high steady performance from the detailed RTO, while maintaining the control fast thanks to the advanced control.

Standard formulations of the RTO deal with the optimization of the plant operated at equilibrium points. However, there exist many scenarios where the plant operates optimally with a periodic behaviour, such as HVAC systems, solar plants, water distribution networks, electric networks, among others. For these systems with periodic nature, a dynamic RTO is better suited because of its

ability to calculate not only the optimal steady-state, but also the optimal periodic trajectory. This constitutes a generalization of the standard RTO and usually comes at the expense of an increased complexity in the control problem because of the larger number of variables and constraints. While dynamic RTO schemes may theoretically converge to the optimal steady operation, they are still sensitive to plant-model mismatch, thus making them susceptible to a performance decrease.

In order to cope with the issues derived from the plant-model mismatch, modifier-adaptation (MA) formulations of the RTO emerged and have been studied over the last decades [10, 125, 126, 127] with promising results. They update the model-based RTO problem with affine modifiers that incorporate information of the real system. Upon convergence of the modifiers, the modified problem is able to calculate either a steady operation of the real system that satisfies the necessary conditions of optimality or the an input profile of a batch process that satisfies the first order necessary conditions of optimality (NCO) [128] from an initially inaccurate model. Modifier-adaptation schemes have been mainly built upon the standard RTO to compute the optimal steady-state of a system. In this work we present a periodic modifier-adaptation scheme which is built upon a dynamic RTO and is able to calculate, upon convergence of the modifiers, the periodic trajectory of a real system that satisfies the NCO of the real plant. The proposed approach can be seen as a generalization of the MA scheme proposed in [9] to include optimal periodic behaviour.

This chapter is structured as follows: In Section 5.2 we introduce the problem under consideration, along with the two-layer control scheme. Then, in Section 7.2.3 we show how to modify the dynamic RTO so that, upon convergence, its solution matches the NCO of the optimal control problem. Then, in Section 5.5 we detail how to transform the optimal operation computed by the dynamic RTO into a valid reference for the advanced control layer. Section 5.4 shows a way to design the advanced layer to follow a dynamic reference. In Section 5.6 we present the full algorithm required to implement the two-layer control scheme with periodic modifier-adaptation. A simplified version of this algorithm is used in Section 5.7 on the quadruple tank benchmark example to test the performance of the proposed approach. Finally, Section 5.8 discusses the conclusions.

This chapter is based on the results of the published paper [125] and the manuscript [129].

5.2 Problem formulation

Consider the following discrete system:

$$x_{k+1} = f_{p,k}(x_k, u_k), \tag{5.1}$$

where $x_k \in \mathbb{R}^{n_x}$ and $u_k \in \mathbb{R}^{n_u}$ are respectively the states and inputs of the system at time k, and $f_{p,k} : \mathbb{R}^{n_x \times n_u} \to \mathbb{R}^{n_x}$ represents the dynamics of the real system at time k. Each step in k represents t_T seconds.

Let system (5.1) be periodic with known period Tt_T seconds, i.e. $f_{p,k} = f_{p,k+T}$, and let x_0 be the initial state. At the first step of each period, given the sequence of T next inputs $\mathbf{u}_T = \begin{bmatrix} u_0^T & u_1^T & \cdots & u_{T-1}^T \end{bmatrix}^T \in \mathbb{R}^{Tn_u}$, then the T following states of the system (5.1) are defined by the time-invariant function $F_p : \mathbb{R}^{n_x \times Tn_u} \to \mathbb{R}^{Tn_x}$ so that:

$$\mathbf{x}_T = \begin{bmatrix} x_1^T & x_2^T & \cdots & x_T^T \end{bmatrix}^T = F_p(x_0, \mathbf{u}_T). \tag{5.2}$$

At any time k, the states and inputs of system (5.1) can be subject to (possibly nonlinear) constraints of the form:

$$g_k(x_k, u_k) \leq 0, \tag{5.3}$$

which are also periodic with period Tt_T seconds. Considering the periodic constraint $x_0 = x_T$, at the first step of each period the constraints (5.3) can also be expressed by its compact form:

$$G(\mathbf{x}_T, \mathbf{u}_T) \leq 0,$$

where $G : \mathbb{R}^{Tn_x \times Tn_u} \to \mathbb{R}$.

Let the economic cost of operating the previous system at any given time k be given by the stage cost function $\phi_k(x_k, u_k)$. The optimal economic control problem calculates the infinite sequence of inputs that, when applied to the system (5.1), minimizes the economic cost given by the stage cost function $\phi_k(x_k, u_k)$ over time. Let the stage cost function ϕ_k be periodic with period Tt_T seconds and consider the periodic constraint $x_0 = x_T$, then at the first step of each period, the time-invariant cost function Φ represents the sum of stage cost functions ϕ_k over the T future steps and is defined as:

$$\Phi(\mathbf{x}_T, \mathbf{u}_T) = \sum_{i=0}^{T-1} \phi_i(x_i, u_i). \tag{5.4}$$

Given the initial state x_0 of the system, the optimal economic control problem is formulated as follows:

$$\begin{aligned} \min_{\mathbf{u}_\infty} \quad & \sum_{k=0}^{\infty} \phi_k(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f_{p,k}(x_k, u_k), \quad \text{for all } k = 0, 1, \ldots, \infty \\ & g_k(x_k, u_k) \leq 0, \quad \text{for all } k = 0, 1, \ldots, \infty. \end{aligned} \tag{5.5}$$

In real applications, the previous fomulation is seldom implemented because of two main reasons: (i) the real system dynamics (i.e. $f_{p,k}$) are usually unknown, and (ii) the infinite number of decision variables hinders the implementation for most practical cases.

Figure 5.1: Control diagram of the two-layer scheme.

5.2.1 Two-layer control scheme

In practice, problem (5.5) is often tackled using a two-layer control scheme (Figure 5.1). In this scheme, the upper layer, also known as real-time optimization (RTO), calculates the optimal operation of the system. Whereas the lower one, known as advanced control, computes the input sequence required to take the system from its current state to a given reference. These layers are separated and are solved with different time scales, they usually use different models and time horizons, so we consider also an intermediate layer called steady trajectory target optimization (STTO) which turns the optimal operation computed by the upper layer into a valid steady reference for the lower layer.

The basis of the two layer architecture is to split the main control problem into two smaller problems of different complexities and which are solved with different frequencies. On one hand, the RTO usually works with a complex and accurate model of the global plant. This model typically describes the fundamental and static behaviour of the plant, which results in large time scales and low update frequency for the RTO. On the other hand, the advanced control generally uses a local dynamic model of system. It uses simple and fast models and its time scales are short. One of the benefits of this scheme is that the upper layer does not need to be recalculated with the same frequency as the lower one. This reduces computational costs, while keeping the control fast.

5.2.2 Dynamic real-time optimization for periodic operation

In this work we use dynamic real-time optimization (DRTO) as the upper layer. Unlike standard RTO, which aims to calculate the optimal steady setpoint (x^s, u^s), the objective of the DRTO is to compute the optimal periodic trajectory with a predefined period of Tt_T seconds $(\hat{\mathbf{x}}_T^{\text{drto}}, \mathbf{u}_T^{\text{drto}})$. The optimal periodic trajectory can be seen as a generalization of the optimal steady setpoint, since they lead to the same solution for $T = 1$. Consequently, the DRTO can lead to better steady performance than the standard RTO, at the expense of it being a more intricate problem. In the case of periodic systems, it has been proven that unlike the RTO, the DRTO formulation is able to capture their optimal steady operation [130].

The DRTO uses a model of the real system F_m, instead of the real system dynamics F_p described in (5.2):

$$\hat{\mathbf{x}}_T = \begin{bmatrix} \hat{x}_1^T & \hat{x}_2^T & \cdots & \hat{x}_T^T \end{bmatrix}^T = F_m(x_0, \mathbf{u}_T), \tag{5.6}$$

where $\hat{x}_k$ is the state predicted by the model at time k. Like (5.2), each step in k

equals t_T seconds. Because of the complexity of real systems, models are usually unable to perfectly capture the real dynamics, leading to plant-model mismatch, i.e. $x_{k+1} \neq \hat{x}_{k+1}$. If not treated correctly, this discrepancy causes performance and stability issues and may render the control unusable.

In the two-layer scheme, one iteration of the DRTO is solved every $t_D = D(Tt_T)$ seconds. For simplicity, we consider parameter D to be a positive integer. Given the period T, the DRTO problem can be formulated as:

$$\begin{aligned}(x_0^{\text{drto}}, \hat{\mathbf{x}}_T^{\text{drto}}, \mathbf{u}_T^{\text{drto}}) = \\ \underset{x_0, \hat{\mathbf{x}}_T, \mathbf{u}_T}{\arg\min} \quad & \sum_{i=0}^{T-1} \phi_i(\hat{x}_i, u_i) \\ \text{s.t.} \quad & \hat{\mathbf{x}}_T = F_m(x_0, \mathbf{u}_T) \\ & G(\hat{\mathbf{x}}_T, \mathbf{u}_T) \leq 0 \\ & \hat{x}_T = x_0.\end{aligned} \tag{5.7}$$

The aforementioned formulation of the DRTO computes the optimal periodic operation for the available model of the system. However, due to plant-model mismatch, we know that this operation may not be optimal for the real system and might even lead to constraint violation. In the next section we present a reformulation of (5.7) which uses gradient-based modifiers to update the base model so that, upon convergence, the solution of the modified DRTO matches the NCO of the optimal periodic operation. Later, in Sections 5.4 and 5.5, the advanced control and the steady trajectory target optimization layers will be detailed.

5.3 Periodic modifier-adaptation

Modifier-adaptation (MA) methodologies arose to correct the plant-model mismatch at the RTO level [9, 10]. They use measures and gradients from the system to build modifiers that update the RTO with affine terms. Upon convergence, MA schemes guarantee the satisfaction of the first order necessary conditions for optimality of the optimal problem. Traditionally, MA schemes have been built upon the standard RTO, which ultimately calculates the steady setpoint that satisfy the plant's NCO. In this section we generalize state-of-the-art approaches and show how to apply modifier-adaptation to the DRTO problem (5.7) and correct (locally) the plant-model mismatch for the case of optimal periodic trajectories. Zeroth and first order modifiers will be presented to update the dynamic model and ensure that, upon convergence, the optimal solution of the modified DRTO matches the NCO of the optimal periodic trajectory of the real system.

Let each iteration of the DRTO be labelled by index l. Then, given the modifiers $\lambda_l^x \in \mathbb{R}^{Tn_x \times n_x}$, $\lambda_l^u \in \mathbb{R}^{Tn_x \times Tn_u}$ and $\epsilon_l \in \mathbb{R}^{Tn_x}$, we introduce the periodic modifier-adaptation (P-MA) formulation of the DRTO at iteration l:

$$
\begin{aligned}
(x_0^{\text{drto}}, \hat{\mathbf{x}}_T^{\text{drto}}, \mathbf{u}_T^{\text{drto}}) = \\
\underset{x_0, \hat{\mathbf{x}}_T, \mathbf{u}_T}{\arg\min} \quad & \Phi(\hat{\mathbf{x}}_T, \mathbf{u}_T) \\
\text{s.t.} \quad & \hat{\mathbf{x}}_T = F_m(x_0, \mathbf{u}_T) + \lambda_l^x x_0 + \lambda_l^u \mathbf{u}_T + \epsilon_l \\
& G(\hat{\mathbf{x}}_T, \mathbf{u}_T) \leq 0 \\
& M\hat{\mathbf{x}}_T = x_0,
\end{aligned}
\tag{5.8}
$$

where M represents the constant matrix that ensures that the periodic constraint meets, i.e. $\hat{x}_T = x_0$.

Assumption 5.1. *At every iteration l, Problem* (5.8) *is feasible and has an unique minimizer.*

After solving problem (5.8), the DRTO identifies a set of variables $\mathbf{r}_e^{\text{drto}} \in \mathbb{R}^{Tn_r}$ that univocally defines the (model) optimal economic trajectory

$$
\mathbf{r}_e^{\text{drto}} = r_e(\hat{\mathbf{x}}_T^{\text{drto}}, \mathbf{u}_T^{\text{drto}}) \tag{5.9}
$$

and passes it to the STTO, which then transforms it into a valid reference for the MPC $(\hat{\mathbf{z}}_{N,j}^{\text{ref}}, \mathbf{v}_{N,j}^{\text{ref}})$ (See Figure 5.1).

Now, we show how to calculate the modifiers λ_l^x, λ_l^u and ϵ_l so that, upon convergence, the NCO conditions of problem (5.8) converge to those of the optimal problem.

5.3.1 KKT matching

In this section we show how to update the modifiers λ_l^x, λ_l^u and ϵ_l so that the first order necessary conditions of optimality (NCO), also known as KKT conditions, of the P-MA formulation of the DRTO (5.8) match with those of the optimal economic problem.

In order to make the KKT matching possible, we make the following assumption:

Assumption 5.2. *Both the real system F_p^θ and the base model F_m^θ are C^1 continuous on $\theta \in \Theta \subset \mathbb{R}^{n_x \times Tn_u}$, being Θ the feasible region of θ. This means that their partial derivatives in this region are both defined and continuous.*

Given period T, the real optimal periodic trajectory $(\mathbf{x}_T^{\text{opt}}, \mathbf{u}_T^{\text{opt}})$ can be computed as the optimal solution of the following optimization problem:

$$
\begin{aligned}
(x_0^{\text{opt}}, \mathbf{x}_T^{\text{opt}}, \mathbf{u}_T^{\text{opt}}) = \\
\underset{x_0, \mathbf{x}_T, \mathbf{u}_T}{\arg\min} \quad & \Phi(\mathbf{x}_T, \mathbf{u}_T) \\
\text{s.t.} \quad & \mathbf{x}_T = F_p(x_0, \mathbf{u}_T) \\
& G(\mathbf{x}_T, \mathbf{u}_T) \leq 0 \\
& M\mathbf{x}_T = x_0.
\end{aligned}
\tag{5.10}
$$

For the sake of simplicity and comparison, we define $\theta = \begin{bmatrix} x_0 \\ \mathbf{u}_T \end{bmatrix}$ and reformulate (5.10) as:

$$\begin{aligned} \min_{\theta} \quad & \Phi^{\theta}(F_p^{\theta}(\theta), \theta) && \text{(5.11a)} \\ \text{s.t.} \quad & G^{\theta}(F_p^{\theta}(\theta), \theta) \leq 0 && \text{(5.11b)} \\ & M_1 F_p^{\theta}(\theta) + M_2 \theta = \mathbf{0}, && \text{(5.11c)} \end{aligned}$$

where $\Phi^{\theta}, F_p^{\theta}$, G^{θ}, $M1$ and $M2$ are functions derived from rewritting the ones in (5.10) in terms of θ, e.g. $\mathbf{x}_T = F_p^{\theta}(\theta)$, and (5.11c) corresponds to the periodic constraint.

We also define the modified version of the dynamic RTO (5.8) at step l:

$$\begin{aligned} \min_{\theta} \quad & \Phi^{\theta}(F_m^{\theta}(\theta) + (\Lambda_l^{\theta})^T \theta + \epsilon_l, \theta) \\ \text{s.t.} \quad & G^{\theta}(F_m^{\theta}(\theta) + (\Lambda_l^{\theta})^T \theta + \epsilon_l, \theta) \leq 0 \\ & M_1(F_m^{\theta}(\theta) + (\Lambda_l^{\theta})^T \theta + \epsilon_l) + M_2 \theta = \mathbf{0}, \end{aligned} \tag{5.12}$$

where ϵ_l and $\Lambda_l^{\theta} = \begin{bmatrix} \lambda_l^x & \lambda_l^u \end{bmatrix}^T$ refers to the zeroth and first order modifiers respectively at step l.

The Lagrangian function associated to the problem (5.11) is:

$$\begin{aligned} Ł_p(\theta) =& \Phi^{\theta}(F_p^{\theta}(\theta), \theta) + \pi_1^T \Big(G^{\theta}(F_p^{\theta}(\theta), \theta) \Big) + \\ & \pi_2^T \Big(M_1 F_p^{\theta}(\theta) + M_2 \theta \Big), \end{aligned}$$

and its gradient with respect to the decision variable θ is:

$$\begin{aligned} \frac{\partial Ł_p}{\partial \theta} =& \frac{\partial \Phi^{\theta}}{\partial F_p^{\theta}} \Big(\frac{\partial F_p^{\theta}}{\partial \theta} \Big) + \frac{\partial \Phi^{\theta}}{\partial \theta} + \pi_1^T \Big[\frac{\partial G^{\theta}}{\partial F_p^{\theta}} \Big(\frac{\partial F_p^{\theta}}{\partial \theta} \Big) + \frac{\partial G^{\theta}}{\partial \theta} \Big] + \\ & \pi_2^T \Big(M_1 \Big(\frac{\partial F_p^{\theta}}{\partial \theta} \Big) + M_2 \Big). \end{aligned}$$

Analogously, the gradient of the Lagrangian function associated to problem (5.12) is the following:

$$\begin{aligned} \frac{\partial Ł_m}{\partial \theta} =& \frac{\partial \Phi^{\theta}}{\partial F_m^{\theta}} \Big(\frac{\partial F_m^{\theta}}{\partial \theta} + \Lambda_{\infty}^{\theta} \Big) + \frac{\partial \Phi^{\theta}}{\partial \theta} + \\ & \pi_1^T \Big[\frac{\partial G^{\theta}}{\partial F_m^{\theta}} \Big(\frac{\partial F_m^{\theta}}{\partial \theta} + \Lambda_{\infty}^{\theta} \Big) + \frac{\partial G^{\theta}}{\partial \theta} \Big] + \\ & \pi_2^T \Big(M_1 \Big(\frac{\partial F_m^{\theta}}{\partial \theta} + \Lambda_{\infty}^{\theta} \Big) + M_2 \Big). \end{aligned}$$

Let θ^* be the (a priori unknown) optimal operation of the system, then the KKT conditions associated to problem (5.11) are:

$$\frac{\partial \text{Ł}_p}{\partial \theta}(\theta^*) = 0 \tag{5.13a}$$
$$G^\theta(F_p^\theta(\theta^*), \theta^*) \leq 0 \tag{5.13b}$$
$$M_1 F_p^\theta(\theta) + M_2 \theta = \mathbf{0} \tag{5.13c}$$
$$\pi_1^*, \pi_2^* \geq 0 \tag{5.13d}$$
$$(G^\theta(F_p^\theta(\theta^*), \theta^*))\pi_1^* = 0 \tag{5.13e}$$
$$(M_1 F_p^\theta(\theta) + M_2\theta)_j \pi_{2,j}^* = 0, \;\; j = 0, 1, \ldots, n_x. \tag{5.13f}$$

Analogously, the KKT conditions associated to problem (5.12) are:

$$\frac{\partial \text{Ł}_m}{\partial \theta}(\theta^*) = 0 \tag{5.14a}$$
$$G^\theta(F_m^\theta(\theta^*) + (\Lambda_l^\theta)^T\theta^* + \epsilon_l, \theta^*) \leq 0 \tag{5.14b}$$
$$M_1(F_m^\theta(\theta^*) + (\Lambda_l^\theta)^T\theta^* + \epsilon_l) + M_2\theta^* = \mathbf{0} \tag{5.14c}$$
$$\pi_1^*, \pi_2^* \geq 0 \tag{5.14d}$$
$$(G^\theta(F_m^\theta(\theta^*) + (\Lambda_l^\theta)^T\theta^* + \epsilon_l, \theta^*))\pi_1^* = 0 \tag{5.14e}$$
$$\left(M_1(F_m^\theta(\theta^*) + (\Lambda_l^\theta)^T\theta^* + \epsilon_l) + M_2\theta^*\right)_j \pi_{2,j}^* = 0, \;\; j = 0, 1, \ldots, n_x. \tag{5.14f}$$

Therefore, the KKT conditions of both problems match upon convergence of the modifiers (represented by $l = \infty$) if and only if:

$$\frac{\partial \text{Ł}_p}{\partial \theta}(\theta^*) = \frac{\partial \text{Ł}_m}{\partial \theta}(\theta^*) = \mathbf{0} \tag{5.15a}$$
$$F_p^\theta(\theta^*) = F_m^\theta(\theta^*) + (\Lambda_\infty^\theta)^T\theta^* + \epsilon_\infty. \tag{5.15b}$$

To meet (5.15a), we need to set the first order modifiers Λ^θ so that:

$$\frac{\partial F_p^\theta}{\partial \theta}(\theta^*) = \frac{\partial F_m^\theta}{\partial \theta}(\theta^*) + \Lambda_\infty^\theta.$$

Thus, the optimal modifiers $(\lambda_\infty^x, \lambda_\infty^u)$ must be computed as:

$$\Lambda_\infty^\theta = \begin{bmatrix}\lambda_\infty^x & \lambda_\infty^u\end{bmatrix}^T = \frac{\partial F_p^\theta}{\partial \theta}(\theta^*) - \frac{\partial F_m^\theta}{\partial \theta}(\theta^*). \tag{5.16}$$

To converge to the optimal modifiers, we follow an update policy similar to the one proposed in [9]. Let θ_l^{drto} be the solution of the DRTO (5.8) at iteration l, then the modifiers at iteration $l+1$ are calculated as:

$$\Lambda_{l+1}^\theta = \begin{bmatrix}\lambda_{l+1}^x & \lambda_{l+1}^u\end{bmatrix}^T = \frac{\partial F_p^\theta}{\partial \theta}(\theta_l^{\text{drto}}) - \frac{\partial F_m^\theta}{\partial \theta}(\theta_l^{\text{drto}}). \tag{5.17}$$

While the gradients of the model can usually be easily computed with user-defined precision, e.g. by numeric or analytical differentiation (Section 5.3.2), the gradients of the real system often are cumbersome and rely on noisy measures to carry out estimations. The estimation of such gradients is currently a focal point in the modifier-adaptation research [131, 132]. In chapter 7, an approach to the computation of the gradients of real-systems will be proposed.

Given the modifiers Λ_∞^θ computed in (5.16), in order to meet (5.15b), the modifier ϵ_∞ must be set as:

$$\epsilon_\infty = F_p^\theta(\theta^*) - \left(F_m^\theta(\theta^*) + (\Lambda_\infty^\theta)^T\theta^*\right). \tag{5.18}$$

Applying an update like the one from (5.17), we get to the following update for ϵ_l:

$$\epsilon_{l+1} = F_p^\theta(\theta_l^{\text{drto}}) - \left(F_m^\theta(\theta_l^{\text{drto}}) + (\Lambda_l^\theta)^T\theta_l^{\text{drto}}\right). \tag{5.19}$$

5.3.2 Gradients of a linear model

In this section, we derive the analytical expression for the gradients of a linear model.

Given the discrete-time linear model

$$\hat{x}_{k+1} = f_{m,k}(x_k, u_k) = A_k\hat{x}_k + B_k u_k, \tag{5.20}$$

we have that

$$\hat{\mathbf{x}}_T = F_m(x_0, \mathbf{u}_T) = \mathcal{F}^x x_0 + \mathcal{F}^u \mathbf{u}_T, \tag{5.21}$$

or equivalently

$$\hat{\mathbf{x}}_T = F_m^\theta(\theta) = \begin{bmatrix}\mathcal{F}^x & \mathcal{F}^u\end{bmatrix}\theta, \tag{5.22}$$

where

$$\mathcal{F}^x = \begin{bmatrix} A_0 \\ A_0A_1 \\ \vdots \\ \prod_{i=0}^{T-1} A_i \end{bmatrix}, \quad \mathcal{F}^u = \begin{bmatrix} B_0 & & & \\ A_1B_0 & B_1 & & \\ \vdots & \vdots & \ddots & \\ (\prod_{i=1}^{T-1} A_i)B_0 & (\prod_{i=2}^{T-1} A_i)B_1 & \dots & B_{T-1} \end{bmatrix}. \tag{5.23}$$

Therefore, the gradients of the model are constant and can be explicitly computed as $\frac{\partial F_m^\theta}{\partial\theta} = \begin{bmatrix}\mathcal{F}^x & \mathcal{F}^u\end{bmatrix}^T$.

5.4 MPC for periodic operation

Model predictive controllers (MPCs) are one of the multiple choices for the bottom layer of the two-layer scheme introduced in Section 5.2, often referred to as advanced control. Their objective is to calculate the control sequence that takes the system from its current state z_j to the reference given by the STTO.

Contrary to the DRTO, the MPC generally has a more local and fast nature, which may make its correspondent real system different from the one presented in (5.1). Let the (real) local system be defined as

$$z_{j+1} = f_{p,j}^{\text{mpc}}(z_j, v_j), \tag{5.24}$$

where $z_j \in \mathbb{R}^{n_z}$ and $v_j \in \mathbb{R}^{n_v}$ represent respectively the states and inputs of the local system at time j, and $f_{p,j}^{\text{mpc}} : \mathbb{R}^{n_z \times n_v} \to \mathbb{R}^{n_z}$ represents the dynamics of the real system at time j. Note that the local system is parameterized by j to indicate that the discretization time of the local system (t_N seconds) is generally different than that of the global system parameterized by k (t_T seconds). Therefore, the system is periodic with period $t_N L = t_T T$ seconds.

The MPC solves an optimization problem at each time step j to calculate the sequence of control inputs that minimize a given cost function. Given a reference at time j ($\hat{\mathbf{z}}_{N,j}^{\text{ref}}, \mathbf{v}_{N,j}^{\text{ref}}$), we use an offset-free MPC formulation based on [133, 134].

Let the MPC model of the local system (5.24) be defined as

$$\hat{z}_{j+1} = f_m^{\text{mpc}}(\hat{z}_j, v_j, d_j), \tag{5.25}$$

where f_m^{mpc} is usually a linear model which allows fast MPC implementations and $d_j \in \mathbb{R}^{n_z}$ is the so-called disturbance at time j. In contrast to the local system (5.24), we consider that the model f_m^{mpc} is time-invariant and its dependence of time comes through the disturbances d_j. Local constraints are also considered, but for the sake of simplicity, only as box constraints on the inputs v_j, i.e. $v^L \leq v_j \leq v^U$. More general constraints require robust formulations of the MPC to guarantee recursive feasibility [135, 136, 137] and are out of the scope of this work.

In order to match the MPC model with the real periodic system, the disturbances d_j should be periodic over the periodic horizon L and satisfy the following equality:

$$f_m^{\text{mpc}}(z_j, v_j, d_j) = f_{p,j}^{\text{mpc}}(z_j, v_j). \tag{5.26}$$

Now we present a simple way to estimate the disturbances

$$d_{j+L} = d_j + K^d \left(f_{p,j}^{\text{mpc}}(z_j, v_j) - f_m^{\text{mpc}}(z_j, v_j, d_j) \right), \tag{5.27}$$

where $K^d \in \mathbb{R}^{n_z \times n_z}$ is a filtering matrix that must be designed so that (5.27) is stable.

Given the current state z_j and the sequence of future disturbances $\mathbf{d}_N = \begin{bmatrix} d_j & d_{j+1} & \dots & d_{j+N-1} \end{bmatrix}$, the offset-free periodic MPC at time step j is formulated as follows:

$$\begin{aligned} &(\mathbf{z}_N^*, \mathbf{v}_N^*) = \\ &\underset{\hat{\mathbf{z}}_N, \mathbf{v}_N}{\arg\min} \quad \ell^{\text{mpc}}(\hat{\mathbf{z}}_N, \mathbf{v}_N, \hat{\mathbf{z}}_{N,j}^{\text{ref}}, \mathbf{v}_{N,j}^{\text{ref}}) \\ &\quad \text{s.t.} \quad \hat{z}_{i+1} = f_m^{\text{mpc}}(\hat{z}_i, v_i, d_{j+i}), \text{ for all } i = 0, 1, \dots, N-1 \\ &\qquad\quad v^L \leq v_i \leq v^U, \text{ for all } i = 0, 1, \dots, N-1 \\ &\qquad\quad \hat{z}_0 = z_j, \end{aligned} \tag{5.28}$$

where ℓ^{mpc} is a cost function that penalizes the distance between the reference sequences of states and inputs, and the sequences $\hat{\mathbf{z}}_N = \begin{bmatrix} \hat{z}_1 & \hat{z}_2 & \dots & \hat{z}_N \end{bmatrix}$, $\mathbf{v}_N = \begin{bmatrix} v_0 & v_1 & \dots & v_{N-1} \end{bmatrix}$. The current local state z_j is considered known. The MPC follows a receding horizon scheme, which means that only the first computed input v_0^* is applied to the system at each iteration of the MPC.

5.5 Steady trajectory target optimization (STTO)

The optimal solution of the DRTO presented in Section 7.2.3 leads to the reference trajectory $\mathbf{r}_e^{\text{drto}}$. The objective of the STTO is to transform this reference trajectory into a valid target $(\hat{\mathbf{z}}_{L,j}^{\text{ref}}, \mathbf{v}_{L,j}^{\text{ref}})$ for the MPC defined in Section 5.4, i.e. one that is feasible for the MPC constraints.

The first step is to match the time scale of the DRTO (t_T) with that of the MPC (t_N). Usually, the DRTO works with longer time steps than the MPC ($t_T > t_N$). Therefore, one must transform the reference given by the DRTO into one with the same time scale of the MPC. The reference given by the DRTO spans a total duration of Tt_T seconds. To transform it into the time scale of the MPC, just divide it into segments of t_N seconds and check which value of the reference trajectory $\mathbf{r}_e^{\text{drto}}$ corresponds to each segment. To avoid dealing with segments that comprise two or more values, we assume that the DRTO sampling time t_T is a multiple of the MPC sampling time t_N. The new reference with time scale t_N is denoted $\mathbf{r}_e^{\text{stto}}$ and its length is L. Then, this reference is shifted to match the current time step j.

Let $\ell^{\text{stto}} : \mathbb{R}^{Ln_z \times Ln_v \times Ln_r} \to \mathbb{R}$ be a function that penalizes the distance between the trajectories of states and inputs of the MPC $(\hat{\mathbf{z}}_L, \mathbf{v}_L)$ and the reference trajectory $(\mathbf{r}_e^{\text{stto}})$. Then, the STTO problem at each step j can be formulated as:

$$
\begin{aligned}
(\hat{\mathbf{z}}^{\text{ref}}_{L,j}, \mathbf{v}^{\text{ref}}_{L,j}) = & \\
\underset{\hat{\mathbf{z}}_L, \mathbf{v}_L}{\arg\min} \quad & \ell^{\text{stto}}(\hat{\mathbf{z}}_L, \mathbf{v}_L, \mathbf{r}^{\text{stto}}_e) \\
\text{s.t.} \quad & \hat{z}_{i+1} = f^{\text{mpc}}_m(\hat{z}_i, v_i, d_{j+i}), \text{ for all } i = 0, 1, \ldots, L-1 \\
& v^L \leq v_i \leq v^U, \text{ for all } i = 0, 1, \ldots, L-1 \\
& \hat{z}_L = \hat{z}_0,
\end{aligned}
\tag{5.29}
$$

where $\mathbf{d}_L = \begin{bmatrix} d_j & d_{j+1} & \ldots & d_{j+L-1} \end{bmatrix}$ are the disturbances estimated in (5.27).

Finally, we usually have that the periodic horizon is larger than the control horizon, i.e. $L > N$ and therefore the obtained reference $(\hat{\mathbf{z}}^{\text{ref}}_{L,j}, \mathbf{v}^{\text{ref}}_{L,j})$ must be trimmed to match the control horizon N. In the opposite case, i.e. $L < N$, the obtained reference must be repeated until it matches the control horizon N. The new reference trajectory will be referred to as $(\hat{\mathbf{z}}^{\text{ref}}_{N,j}, \mathbf{v}^{\text{ref}}_{N,j})$ and constitutes a valid reference for the MPC layer which is guaranteed to be feasible for the MPC constraints.

As commented in Section 5.2, the STTO layer needs to be computed before every MPC iteration to guarantee that the reference trajectory is admissible for the most recent values of disturbances. Since the DRTO reference $\mathbf{r}^{\text{drto}}_e$ is periodic by definition, endless shifting is possible and the STTO/MPC loop can always control the system to the latest available reference trajectory.

In the next section, we detail the full algorithms for the DRTO/STTO and the MPC layers.

5.6 Periodic modifier-adaptation algorithm

In this section, we go through the full periodic modifier-adaptation algorithm. As commented in Section 5.2, the full scheme can be splitted into two main layers with different time scales. Algorithm 5 goes through the DRTO layer, which updates each t_D seconds. It shows how to compute the reference for the STTO and MPC layers. Besides, Algorithm 6 details the STTO and the MPC layers which work with a shorter time scale t_N and calculates every control signal that is applied to the system. Both algorithms run as long as automatic control of the system is required. Upon convergence to the optimal predicted trajectory, the controlled system is guaranteed to reach a periodic behaviour that satisfies the necessary conditions of optimality of the real system.

Remark 5.1. *An optional filtering of the modifiers can be performed after step* (iv) *of Algorithm 5. Filtering the modifiers is used to influence on the speed and stability of the convergence process:*

$$
\begin{aligned}
\tilde{\lambda}^x_{l+1} &= K^x \lambda^x_{l+1} + (\mathrm{I} - K^x)\tilde{\lambda}^x_l \\
\tilde{\lambda}^u_{l+1} &= K^u \lambda^u_{l+1} + (\mathrm{I} - K^u)\tilde{\lambda}^u_l \\
\tilde{\epsilon}_{l+1} &= K^\epsilon \epsilon_{l+1} + (\mathrm{I} - K^\epsilon)\tilde{\epsilon}_l,
\end{aligned}
$$

Algorithm 5 DRTO algorithm (executed each t_D seconds)

Init

(i) Initialize $l = 0$ and the modifiers $\hat{\lambda}_0^x, \hat{\lambda}_0^u, \hat{\epsilon}_0$ to zero.

Loop

(ii) Given the modifiers $\hat{\lambda}_l^x, \hat{\lambda}_l^u, \hat{\epsilon}_l$, compute the optimal trajectory with the DRTO defined in (5.8) and (5.9) and obtain $\mathbf{r}_e^{\text{drto}}$.

(iii) Pass $\mathbf{r}_e^{\text{drto}}$ as a reference to the STTO and MPC Algorithm.

(iv) Estimate the gradients of the model and the real system and update the modifiers according to (5.17) and (5.19):

$$\begin{aligned}
\lambda_{l+1}^x =& \frac{\partial F_p}{\partial x_0}\Big|_{(x_0^{\text{drto}}, \mathbf{u}_T^{\text{drto}})} - \frac{\partial F_m}{\partial x_0}\Big|_{(x_0^{\text{drto}}, \mathbf{u}_T^{\text{drto}})} \\
\lambda_{l+1}^u =& \frac{\partial F_p}{\partial \mathbf{u}_T}\Big|_{(x_0^{\text{drto}}, \mathbf{u}_T^{\text{drto}})} - \frac{\partial F_m}{\partial \mathbf{u}_T}\Big|_{(x_0^{\text{drto}}, \mathbf{u}_T^{\text{drto}})} \\
\epsilon_{l+1} =& F_p(x_0^{\text{drto}}, \mathbf{u}_T^{\text{drto}}) - \\
& \left(F_m(x_0^{\text{drto}}, \mathbf{u}_T^{\text{drto}}) + \hat{\lambda}_l^x x_0^{\text{drto}} + \hat{\lambda}_l^u \mathbf{u}_T^{\text{drto}} \right).
\end{aligned}$$

(v) Wait until next iteration and update $l = l + 1$.

End Loop

Algorithm 6 STTO and MPC algorithm (executed each t_N seconds)

Init

(i) Initialize disturbances of the first period to zero, i.e. $d_j = 0$, where $j = 0, 1, \ldots, L-1$, and set $j = 0$.

Loop

(ii) Get the current state z_j.

(iii) Given $\mathbf{r}_e^{\text{drto}}$ from the DRTO Algorithm, shift it to match the current time step j and solve the STTO layer detailed in Section 5.5 to obtain $(\hat{\mathbf{z}}_{N,j}^{\text{ref}}, \mathbf{v}_{N,j}^{\text{ref}})$.

(iv) Given the reference trajectory $(\hat{\mathbf{z}}_{N,j}^{\text{ref}}, \mathbf{v}_{N,j}^{\text{ref}})$, compute the MPC control input v_j from (5.28).

(v) Apply input v_j to the local system.

(vi) Estimate the disturbance for the next period d_{j+L} using (5.27).

(vii) Wait until next iteration and update $j = j + 1$.

End Loop

where K^x, K^u and K^ϵ are stable filter matrices and $(\tilde{\ })$ notation refers to the filtered modifiers.

Remark 5.2. *The formulations of the STTO and the MPC in steps* (iii) *and* (iv) *of Algorithm 6 are not required to achieve optimal steady performance and alternatives to the ones presented on this chapter may be equally acceptable.*

In the next section, we show the performance of P-MA in a periodic version of the quadruple tank benchmark.

5.7 Illustrative example: Periodic quadruple tank

In this section we show the performance of the periodic modifier-adaptation formulation of the DRTO introduced in this chapter. For the sake of clarity, we focus on showing how the reference trajectory computed by the P-MA formulation of the DRTO converges to the optimal periodic trajectory.

To study the performance of the proposed approach, we test it against a periodic version of the quadruple tank process. This benchmark was first proposed in [138], it has been studied in multiple ocasions due to its interest as a control problem [139, 140, 125] and has been proposed as a benchmark in [141]. The quadruple-tank system scheme is shown in Figure 5.2 and consists of four interconnected tanks that share water according to the following physical equations:

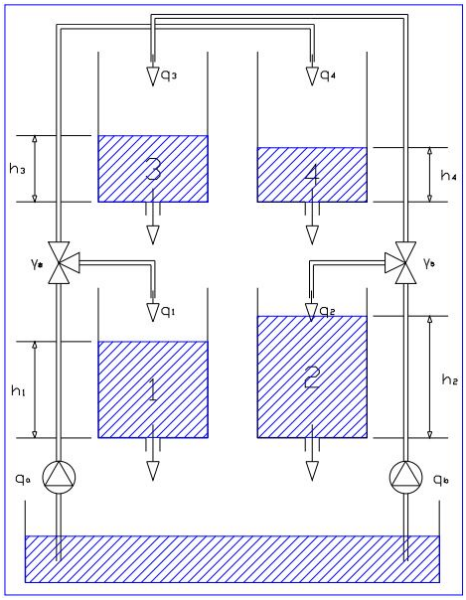


Figure 5.2: Quadruple-tank system diagram, reproduced from [141].

$$\begin{aligned} S\frac{dh_1}{dt} &= -a_1\sqrt{2gh_1} + a_3\sqrt{2gh_3} + \frac{\gamma_a q_a}{3600} \\ S\frac{dh_2}{dt} &= -a_2\sqrt{2gh_2} + a_4\sqrt{2gh_4} + \frac{\gamma_b q_b}{3600} \\ S\frac{dh_3}{dt} &= -a_3\sqrt{2gh_3} + (1-\gamma_b)\frac{q_b}{3600} \\ S\frac{dh_4}{dt} &= -a_4\sqrt{2gh_4} + (1-\gamma_a)\frac{q_a}{3600}. \end{aligned} \tag{5.30}$$

And are subject to the following box constraints:

$$\mathbf{h}_{\min} \leq \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} \leq \mathbf{h}_{\max}, \qquad \mathbf{q}_{\min} \leq \begin{bmatrix} q_a \\ q_b \end{bmatrix} \leq \mathbf{q}_{\max}. \tag{5.31}$$

The quadruple tank process has some properties that makes it both an interesting and a challenging benchmark for control systems:

- It presents large coupling between its subsystems.
- It dynamics are nonlinear.
- States can be measured.
- States and inputs are hard constrained.

	Value	Unit	Description
S	0.03	m^2	Cross-section of the tanks
$\mathbf{a}$	$\begin{bmatrix} 1.31 \\ 1.51 \\ 0.927 \\ 0.882 \end{bmatrix} 10^{-4}$	m^2	Discharge constants
$\mathbf{h}_{\max}$	$\begin{bmatrix} 1.36 \\ 1.36 \\ 1.30 \\ 1.30 \end{bmatrix}$	m	Maximum water level
$\mathbf{h}_{\min}$	$\begin{bmatrix} 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \end{bmatrix}$	m	Minimum water level
$\mathbf{q}_{\max}$	$\begin{bmatrix} 3.6 \\ 4.0 \end{bmatrix}$	m^3/h	Maximum water flow
$\mathbf{q}_{\min}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	m^3/h	Minimum water flow
g	9.81	$\mathrm{m/s}^2$	Gravity acceleration

Table 5.1: Parameters of the plant

- Its real gradients can be analytically computed with the physical equations.

We use a compact notation to define the parameters of the plant:

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix}, \mathbf{u} = \begin{bmatrix} q_a \\ q_b \end{bmatrix}, \boldsymbol{\gamma} = \begin{bmatrix} \gamma_a \\ \gamma_b \end{bmatrix},$$

where water levels $\mathbf{x}$ corresponds to the states and water flows $\mathbf{u}$ to the inputs of the system. Information about the parameters is collected in Table 5.1 and (5.32). The periodic nature of the system is induced through parameter $\boldsymbol{\gamma}$, whose cycle is shown in (5.32), where each column represent a constant value of $\boldsymbol{\gamma}$ for $t_T = 3600$ seconds. Therefore, the plant is periodic with period $T = 7$ hours.

$$\boldsymbol{\gamma}_{\text{cycle}} = \begin{bmatrix} 0.3 & 0.4 & 0.5 & 0.7 & 0.6 & 0.4 & 0.2 \\ 0.6 & 0.5 & 0.4 & 0.2 & 0.3 & 0.5 & 0.7 \end{bmatrix}. \tag{5.32}$$

The model of the system (5.30) is a discrete stationary linear model with discretization time set to 5 seconds and linearized at the point:

$$x_0 = \begin{bmatrix} 0.7293 \\ 0.8102 \\ 0.6594 \\ 0.9408 \end{bmatrix}, \quad u_0 = \begin{bmatrix} 1.948 \\ 2.00 \end{bmatrix}, \quad \boldsymbol{\gamma}_0 = \begin{bmatrix} 0.3 \\ 0.4 \end{bmatrix}$$

Therefore, the model can be written as:

$$\begin{aligned} x_{k+1} = & \begin{bmatrix} 0.945 & 0 & 0.040 & 0 \\ 0 & 0.940 & 0 & 0.032 \\ 0 & 0 & 0.959 & 0 \\ 0 & 0 & 0 & 0.967 \end{bmatrix} (x_k - x_0) + \\ & \begin{bmatrix} 0.0135 & 0.0006 \\ 0.0005 & 0.0180 \\ 0 & 0.0272 \\ 0.0319 & 0 \end{bmatrix} (u_k - u_0) + x_0. \end{aligned} \tag{5.33}$$

At every time step, the system is subject to the box constraints on inputs and states from (5.31), i.e.

$$\mathbf{h}_{\min} \leq \mathbf{x} \leq \mathbf{h}_{\max}, \qquad \mathbf{q}_{\min} \leq \mathbf{u} \leq \mathbf{q}_{\max}. \tag{5.34}$$

Given the economic parameters $c = 1$ and $p = 20$, the economic cost of operating the plant at each discrete time step is given by

$$\phi(x_k, u_k) = (q_a^2 + cq_b^2) + p\frac{0.012}{S(h_1 + h_2)}.$$

We apply Algorithm 5 to compute the optimal periodic trajectory for the system. The control process, i.e. step (iii) and Algorithm 6, is omited for the sake of clarity. The periodic constant is taken as $T = 7$ and the optional filtering of the modifiers proposed in Remark 5.1 has not been taken into account.

The integration of the real process as well as the computation of the optimal trajectory and the P-MA DRTO reference trajectory have been computed using the CasADi optimization tool in Matlab [142]. The gradients of the real process have been computed using numerical differentiation on the real system (5.30), while those of the linear model have been computed using the results from Section 5.3.2.

Figures 5.3 and 5.4 show the optimal trajectory computed by the DRTO with no first order modifiers. The inclusion and convergence of the zeroth order modifier ϵ_l guarantees that the predicted trajectory matches the response of the real system. However, the lack of first order modifiers entails that, upon convergence, the computed input sequence is not optimal. Moreover, since the KKT conditions of this DRTO does not change with time, the sequence of inputs predicted at each iteration is constant over time.

Notice how in Iteration 1 (Figures 5.3 and 5.4), all the modifiers are set to zero and the optimal predicted behaviour is a single steady-state. This is due to

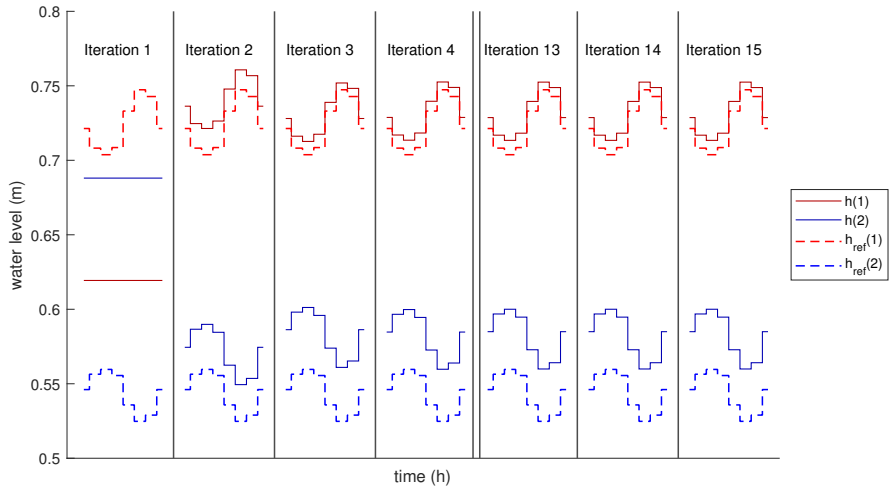


Figure 5.3: Sequence of states computed by the DRTO with the zeroth order modifier in different iterations (solid lines) vs optimal sequence of states (dashed lines).

the time-invariant model used by the DRTO, which differs vastly from the real periodic behaviour of the system.

Figure 5.5 shows how the P-MA DRTO achieves convergence to the optimal sequence of states, and Figure 5.6 shows that this convergence is also achieved with the optimal sequence of inputs. After 15 iterations, the sequences of states and inputs computed by the P-MA DRTO are sufficiently close to the optimal sequences.

5.8 Conclusions

In this chapter we have presented a control scheme which is able to control a periodic system given inaccurate models of it.

First, a reference trajectory is computed by the P-MA DRTO. This layer uses information of the real system to modify the dynamic real-time optimization layer with first and zeroth order modifiers so that, upon convergence of these modifiers, the necessary conditions of optimality of the P-MA DRTO converge to those of the optimal operation of the real system. Then, a steady trajectory target optimization (STTO) translates the trajectory computed by the P-MA DRTO into a feasible reference for the MPC. Finally, the MPC layer uses a disturbance estimator to adapt its model to the real system and be able to converge to the optimal reference.

The proposed P-MA DRTO has been tested against a periodic version of the quadruple tank process, showing that its solution can converge to the optimal

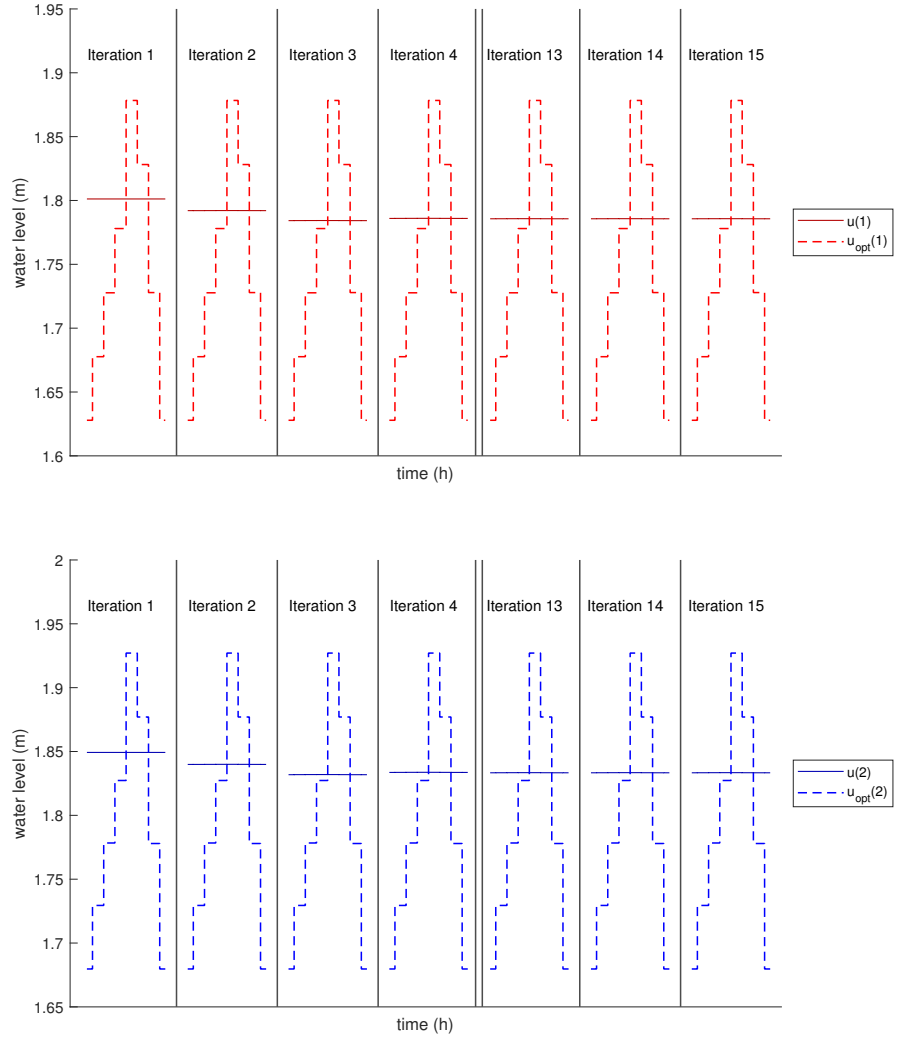


Figure 5.4: Sequence of inputs calculated by the DRTO with the zeroth order modifier in different iterations (solid lines) vs optimal sequence of inputs (dashed lines).

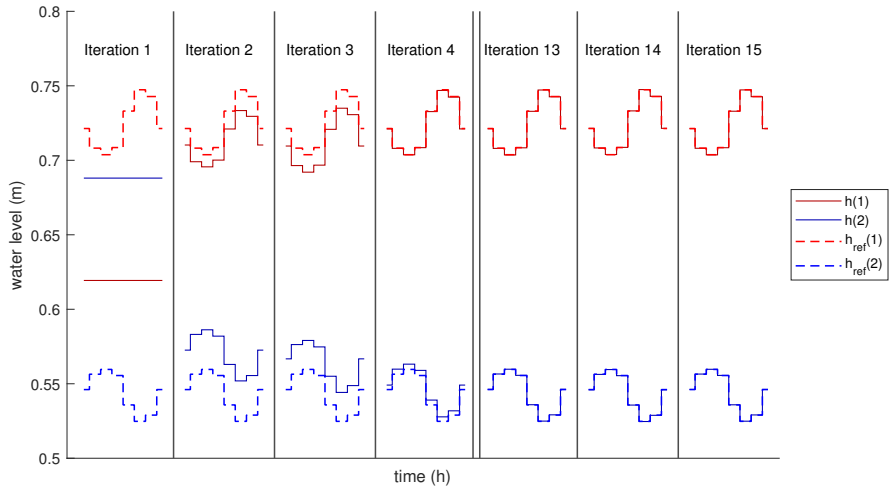


Figure 5.5: Sequence of states computed by the P-MA DRTO in different iterations (solid lines) vs optimal sequence of states (dashed lines).

periodic behaviour given that the gradients of the real system are computed with enough accuracy.

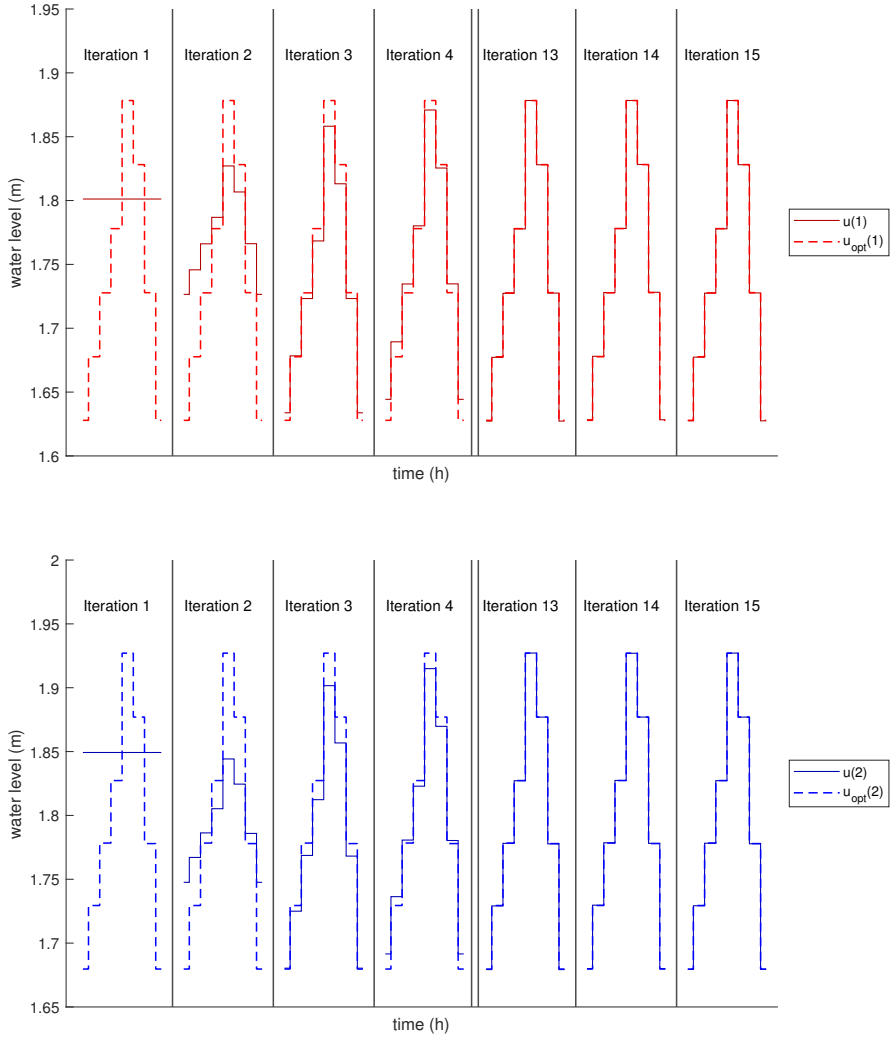


Figure 5.6: Sequence of inputs calculated by the P-MA DRTO in different iterations (solid lines) vs optimal sequence of inputs (dashed lines).

Chapter 6

Robust One-Layer Modifier-Adaptation

6.1 Introduction

The main focus of the hierarchical scheme detailed in section 1.2.2 is to take the system to the minimum cost setpoint. Often, the transient cost is overlooked and traditional MPCs regulate the system to the optimal steady-state as fast as possible. While this can be a good solution when the transient is cheap compared to the steady-state operation, an optimization of the transient cost is essential to further improve the performance. The economic MPC [143, 144] arised to take into account the economic cost of the transient behaviour. Similar to the standard MPC, it requires an RTO to compute the optimal steady operation. But in order to take the plant to this target, the economic MPC takes into account the economic cost of the transient behaviour. While this approach has been studied and used for some time, inconsistencies between control layers may still worsen the economic performance of the transient compared to the optimal control. The integrated one-layer approach proposed in [145] eliminates these inconsistencies by solving the full control problem in a single layer at every sample time and shows an improvement in the overall economic performance. This means that the integrated approach always remains feasible, even in the case of changing economic criteria. However, in order to solve the full control problem at each sample time, the control needs to be fast, and therefore the models need to be very simple. One consequence of using simple models is that the integrated approach struggles to capture the intricate dynamics of real systems, leading to large plant-model mismatch. This mismatch in turn causes a deficient calculation of the optimal operation and ultimately leads to suboptimal control.

One soft spot of model-based methods is how to deal with the mismatch that arises from model inaccuracy. Since the available models used in the RTO and the MPC differ from the real system, not only the optimality may be lost, but even the feasibility. To adress this, multiple solutions have been proposed. In the RTO

layer, we have solutions like parameter adaptation [146, 147], modifier-adaptation [9, 10] and direct input adaptation [148, 149, 150]. In the MPC layer, the mismatch correction, better known as offset-free control, has also received much attention. Most formulations of offset-free control are based on observers and disturbance models [133, 134, 151, 152], but there are also alternative approaches based on variations between different time steps [153].

In this chapter, we propose an integrated one-layer MPC approach to the control problem, which uses modifier-adaptation techniques to update a base control model and achieve an economic steady operation of the plant that satisfies the necessary conditions of optimality of the real economic problem. When a linear base model is considered, we show that the proposed control problem boils down to solving a quadratic programming at each time step, which facilitates a very fast control. Moreover, given an outer compact approximation of the uncertainty (which includes the mismatch between the real plant and the base model), the robustness and convergence to the optimal operation of the proposed controller can be guaranteed.

This chapter is organized as follows: Section 6.2 introduces the control problem and the notation used to adress the inaccuracy of the models. Sections 6.3 and 6.4 introduce the integrated one-layer formulation of the control problem and presents the robust one-layer (ROL) and the robust one-layer modifier-adaptation (ROLMA) formulations of the control problem. Section 6.5 studies the recursive feasibility and convergence of the proposed approach. In Section 6.6, the quadruple tank benchmark is used to test the performance of the ROLMA controller. Finally in Section 6.7 conclusions about the proposed formulations are drawn.

The results of this chapter extend those presented in the published paper [127].

6.2 Problem definition

Consider that we want to control the following discrete and time-invariant system:

$$x_{i+1} = f_m(x_i, u_i) + d^*(x_i, u_i) = Ax_i + Bu_i + d^*(x_i, u_i), \tag{6.1}$$

where $x_i \in \mathcal{X} \subset \mathbb{R}^{n_x}$ and $u_i \in \mathcal{U} \subset \mathbb{R}^{n_u}$ are the state and input of the system at step i, $A \in \mathbb{R}^{n_x \times n_x}$ and $B \in \mathbb{R}^{n_x \times n_u}$ are parameters of the known base model $f_m : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$, and $d^* : \mathcal{X} \times \mathcal{U} \to \mathcal{W}_f$ is a deterministic and unknown function that gathers the mismatch between the base model and the real system and maps it into the known compact region $\mathcal{W}_f \subset \mathbb{R}^{n_x}$, which can be seen as an outer approximation of the effect of the mismatch.

Assumption 6.1. *All the mismatch between the base model and the real system is deterministic and confined in a known compact region $\mathcal{W}_f$ containing the origin in its interior.*

Consider also an alternative formulation of the previous system at time i given

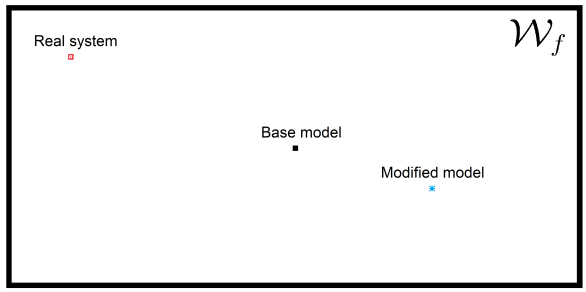


Figure 6.1: Visualization of the mismatch region $\mathcal{W}_f$. The state predicted by the base model lies in the origin, while the one predicted by the modified model and the real next state lie in the neighbourhood $\mathcal{W}_f$.

by a modified model f_m^Λ defined as:

$$\begin{aligned} x_{i+1} =& f_m^\Lambda(x_i, u_i, p_k) + d_\Lambda(x_i, u_i, p_k) = \\ =& f_m(x_i, u_i) + \Lambda^f(x_i, u_i, p_k) + d_\Lambda(x_i, u_i, p_k), \end{aligned} \tag{6.2}$$

where $\Lambda^f(x_i, u_i, p_k) \in \mathcal{W}_f$ is a modifier term that depends on the parameters $p_k \in \mathcal{P}$ and which will ideally converge locally to $d^*(x_i, u_i)$ as k increases. Index k being different from i indicates that the update of the modifiers does not need to coincide with the update of the states. The mismatch between the real plant and the modified model f_m^Λ is captured by $d_\Lambda \in \mathcal{W}_f \oplus -\Lambda^f(x, u, p)$ and is 0 when the modified model matches the real system dynamics.

The modifications made in the modified model f_m^Λ may sometimes handicap the base model, however, the real system will always lie within the robust region defined in (6.1). Figure 6.1 shows all the possible values for the mismatch d^*. At each step, the next state from both the real dynamics and the modified model can be expressed as the sum of the base model plus a mismatch value from the safe region $\mathcal{W}_f$.

Analogously to the states definition, consider the outputs of the system are given by:

$$y_i = h_m(x_i, u_i) + d_h^*(x_i, u_i) = Cx_i + Du_i + d_h^*(x_i, u_i), \tag{6.3}$$

where $y_i \in \mathbb{R}^{n_y}$ represents the outputs of the system at time i, $C \in \mathbb{R}^{n_y \times n_x}$ and $D \in \mathbb{R}^{n_y \times n_u}$ are parameters of a known base model $h_m : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$, and $d_h^* : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathcal{W}_h$ is a deterministic and unknown function that gathers the mismatch between the base output model and the real output system. We consider the plant-model mismatch for the ouputs d_h^* is bounded in the compact set $\mathcal{W}_h \subset \mathbb{R}^{n_y}$, i.e. $d_h^* \in \mathcal{W}_h$.

Consider the feasible outputs are bounded in the compact region $y \in \mathcal{Y}$, then, we can use (6.3) to tighten up the constraints and define the robust safe region

$$\overline{\mathcal{Z}}_m \doteq \{(x, u) : h_m(x, u) \in \mathcal{Y} \ominus \mathcal{W}_h\}, \tag{6.4}$$

such that, for any $(x, u) \in \overline{\mathcal{Z}}_m$, the output of the system stays in $\mathcal{Y}$. The robust safe region is assumed to be time-invariant.

Safe regions have also been studied in a probabilistic way when the system is subject to chance constraints. Probabilistic safe regions are guaranteed to meet the constraints with a prespecified probability (see e.g. chapter 3) and are used in practice to reduce conservativeness.

The objective of the control problem is to take the system (6.1) from its initial state to the equilibrium point that minimizes a given economic function $J(x, u)$, while satisfying the original constraints $y \in \mathcal{Y}$ at every step.

Assumption 6.2. *The gradient of the cost function J is Lipschitz continuous in the feasible region of equilibrium points of the real plant.*

One common approach to automatic control is to compute the fastest path to the optimal setpoint. This usually makes the advanced control simpler, since the cost function consists of a user-defined distance to the target, which often is a quadratic function. This approach is simple and performs very well in cases where the optimal control trajectory involves a cheap transient behaviour. However, to further improve the performance, the economic cost of the transient should also be taken into account. This is the main objective of the economic control, which is the focus of this chapter. While this may increase the difficulty of the problem, economic control constitutes a better solution in terms of economic performance.

6.3 One-layer approach

In this work, we take a similar approach to the one proposed in [145] known as the one-layer approach, in which both the target setpoint and the control signal are computed in the same optimization problem. First, we define the equilibrium constraint of both the modified and the base models.

Given the steady-state x^s, its associated steady input u^s and the modification term Λ^f, the following constraint defines the equilibrium for the modified model:

$$x^s = f_m(x^s, u^s) + \Lambda^f(x^s, u^s, p_k). \tag{6.5}$$

We also define the equilibrium points for the base linear model, which are defined by the following expression:

$$(x^s, u^s) = M_\theta \theta \iff x^s = Ax^s + Bu^s,$$

where $\theta \in \mathbb{R}^{n_\theta}$ is the variable that defines the equilibrium point and $M_\theta \in \mathbb{R}^{(n_x+n_u)\times n_\theta}$ is a matrix that characterizes the equilibrium points of the nominal base model.

One of the reasons why the optimization of the setpoint is traditionally implemented in a different layer of control is because the optimization of the steady operation point is usually nonlinear and highly complex. This means that optimizing it in an online setting could result in a very slow control. A way to

circunvent this issue is to use a quadratic approximation of the economic cost to speed up the solution [145]. Since the original cost function is available, quadratic approximations can be computed with ease and will offer similar performance in a neighborhood of the approximation point.

The one-layer cost function V_N is defined in the same way as in [145]: a weighted sum of the economic cost of an artificial reference $(\tilde{x}^s, u^s)$ and a tracking cost to this reference.

$$\begin{aligned} &V_N(\tilde{\mathbf{x}}, \mathbf{u}, \tilde{x}^s, u^s; x_k^e, u_k^e) = \\ &\sum_{j=0}^{N-1} ||\tilde{x}_j - \tilde{x}^s||_Q^2 + ||u_j - u^s||_R^2 + \|\tilde{x}_N - \tilde{x}_s\|_P^2 + J(x_k^e, u_k^e) + \\ &\nabla_x J(x_k^e, u_k^e)^T(\tilde{x}^s - x_k^e) + \nabla_u J(x_k^e, u_k^e)^T(u^s - u_k^e) + \\ &\frac{\rho_x}{2}||\tilde{x}^s - x_k^e||^2 + \frac{\rho_u}{2}||u^s - u_k^e||^2, \end{aligned} \tag{6.6}$$

where $(\tilde{\mathbf{x}}, \mathbf{u})$ stands for the predicted state and input trajectories and (x_k^e, u_k^e) stands for the approximation point of the economic cost function at step k. Matrices Q, R and P are design parameters that weight the states, inputs and terminal state respectively.

As suggested in [145], a sensible choice for the approximation point is to take the optimal artificial target state and input from the previous iteration of the controller, i.e.

$$(x_k^e, u_k^e) = (\tilde{x}_{k-1}^s, u_{k-1}^s). \tag{6.7}$$

ρ_x and ρ_u are the Lipschitz constants of the cost function with respect to the states and inputs. We assume that suitable choices for the initial parameters p_0 and approximation point (x_0^e, u_0^e) are available.

The cost function V_N can be splitted into its regulation (V_ℓ) and economic (V_O) parts as follows:

$$V_N = V_\ell + V_O \tag{6.8a}$$

$$V_\ell = \sum_{j=0}^{N-1} \|\tilde{x}_j - \tilde{x}^s\|_Q^2 + \|u_j - u^s\|_R^2 + \|\tilde{x}_N - \tilde{x}_s\|_P^2 \tag{6.8b}$$

$$\begin{aligned} V_O =& J(x_k^e, u_k^e) + \nabla_x J(x_k^e, u_k^e)^T(\tilde{x}^s - x_k^e) + \\ &+ \nabla_u J(x_k^e, u_k^e)^T(u^s - u_k^e) + \frac{\rho_x}{2}\|\tilde{x}^s - x_k^e\|^2 + \frac{\rho_u}{2}\|u^s - u_k^e\|^2. \end{aligned} \tag{6.8c}$$

The main disadvantage of the one-layer approach presented in [145] is that, due to the simplicity of the models used, it may control the system to a non optimal steady-state. Furthermore, it does not take into consideration the robustness of the controller. In the next section we will propose a robust formulation of this integrated approach that modifies the model so that, upon convergence, the necessary conditions of optimality are guaranteed.

6.3.1 Robust one-layer (ROL) control for inaccurate models

In this section, we generalize the approach taken in [127] to the case of inaccurate models updated online and propose a robust framework to the one-layer control problem. In particular, we propose a robust tube-based approach which is guaranteed to be recursively feasible. Under the assumption that the modified model eventually converges locally to the real dynamics, convergence to the necessary conditions of optimality (NCO) can also be guaranteed. Note that no assumption is made on the improvement rate of the model, so the modification could temporarily worsen the model without compromising the steady operation. There are no assumptions on the structure of the modification term, but in Section 6.4 an affine update policy based on gradients will be proposed and the benefits of using it will be discussed.

To achieve robustness, we follow the approach described in [154, 155], which tightens up the constraints through the propagation of the uncertainty. Safe tubes are derived from assumption 6.1, which states that all the uncertainty of the base model is constrained in a compact set $d^* \in \mathcal{W}_f$. Therefore, given a control law we can compute the potential region of the states where the system could lie at the following step. We know that, since $\Lambda^f(x_i, u_i, p_k) \in \mathcal{W}_f$, this potential region will contain both the state given by the real system and the state predicted by any possible modified model. If the propagation of the potential region through an horizon N is always contained in the admissible region $\overline{\mathcal{Z}}_m$, then the controlled system will always meet the constraints.

Now we introduce the two different trajectories we will use in the formulation of the controller. The robust trajectory, denoted by $\bar{\mathbf{x}}$ stands for the sequence of N states predicted using the base model f_m. Since the uncertainty region $\mathcal{W}_f$ is defined around the base model, the robust trajectory will constrain the system, for all possible realizations of the uncertainty, into the admissible region. Secondly, the performance trajectory, denoted by $\tilde{\mathbf{x}}$ is the sequence of N states predicted by the modified model f_m^Λ. It is labelled performance because this trajectory should make the system converge to an economic steady operation that satisfies the NCO of the real problem.

The sequence of inputs for both trajectories is shared and determined by the same variable $\mathbf{c} = \begin{bmatrix} c_0^T & c_1^T & \cdots & c_{N-1}^T \end{bmatrix}^T$. The control law is based on a pre-stabilization of the system and is defined as:

$$u_k = K\bar{x}_k + L_\theta \theta + c_k, \tag{6.9}$$

where $L_\theta = \begin{bmatrix} -K & I_{n_\theta} \end{bmatrix} M_\theta$ with I_{n_θ} being the identity matrix of dimension n_θ.

Given the economic cost function $J(x, u)$ and the base (6.1) and modified (6.2) models, we present the Robust One-Layer (ROL) controller, which uses tubes to propagate the uncertainty and grant robustness to the control.

$$
\begin{aligned}
\mathcal{P}_N :& V_N^o(x_k, x_k^e, u_k^e, p_k) = \\
\min_{\mathbf{c},\theta} \quad & V_N(\tilde{\mathbf{x}}, \mathbf{u}, \tilde{x}^s, u^s; x_k^e, u_k^e) && \text{(6.10a)}\\
\text{s.t.} \quad & \tilde{x}_{i+1} = A\tilde{x}_i + Bu_i + \Lambda^f(\tilde{x}_i, u_i, p_k) && \text{(6.10b)}\\
& \tilde{x}^s = A\tilde{x}^s + Bu^s + \Lambda^f(\tilde{x}^s, u^s, p_k) && \text{(6.10c)}\\
& \tilde{x}_0 = x_k, && \text{(6.10d)}\\
& \bar{x}_{i+1} = A\bar{x}_i + Bu_i && \text{(6.10e)}\\
& (\bar{x}_i, u_i) \in \overline{\mathcal{Z}}_m \ominus (\mathcal{H}(i) \times K\mathcal{H}(i)) && \text{(6.10f)}\\
& (\bar{x}_N, \theta) \in \Omega^t && \text{(6.10g)}\\
& u_i = K\bar{x}_i + L_\theta\theta + c_i && \text{(6.10h)}\\
& \bar{x}_0 = x_k && \text{(6.10i)}\\
& \text{for all } i = 0, 1, \ldots, N-1,
\end{aligned}
$$

where $\mathcal{H}(i)$ is the feasible region for the mismatch through the control horizon given by:

$$\mathcal{H}(i) = \bigoplus_{j=0}^{i-1} (A + BK)^j \mathcal{W}_f, \tag{6.11}$$

and $\mathcal{H}(0)$ is the empty set. K is a linear feedback gain, such that $A_K = (A+BK)$ is Hurwitz. The design of the control gain K is not immediate, as it should maximize the domain of attraction while ensuring the stability of the control. The computation of K is out of the scope of this thesis, but the reader can refer to [156] for a detailed explanation of this parameter and how to design it.

We also define the set

$$\mathcal{X}^i = \{(x, \theta) : (x, Kx + L_\theta\theta) \in \overline{\mathcal{Z}}_m \ominus (\mathcal{H}(i) \times K\mathcal{H}(i)), M_\theta\theta \in \lambda\mathcal{Z}_s^m\},$$

where $\mathcal{Z}_s^m \doteq \{(x, u) \in \overline{\mathcal{Z}}_m : (A - I_n)x + Bu = \mathbf{0}_{n_x}\}$.

Note that both $\mathcal{X}^i$ and $\mathcal{Z}_s^m$ depend only on the base model, which makes them time-invariant and can therefore be computed offline.

Taking this into account, the invariant set Σ^t is described by:

$$\Sigma^t = \left\{ (x, \theta) : \begin{bmatrix} A_K & BL_\theta \\ 0 & I_{n_\theta} \end{bmatrix}^i \begin{bmatrix} x \\ \theta \end{bmatrix} \in \mathcal{X}^i, \quad \text{for all } i \geq 0 \right\}.$$

In order to reach the previous set no matter the mismatch, the safe terminal set is defined

$$\Omega^t \doteq \Sigma^t \ominus (\mathcal{H}(N) \times \{0\}).$$

As we will proof in section 6.5, the recursive feasibility of the ROL controller (6.10) is guaranteed despite the modification policy Λ^f used. Furthermore, it

converges to the optimal setpoint as long as the modifier term eventually converges and makes the base model match locally the real system's dynamics.

Besides, should one use a modifier term with a complex structure, the control problem would become very hard and finding a solution could become very time consuming, which would make it ill-suited for most real-time applications.

6.4 Modifier-adaptation update

Online updated models can achieve a good performance without requiring high complexity models. An example of simple and effective update policy are modifier-adaptation (MA) techniques, which use gradient information to update the model with affine terms. Provided that the gradients of the plant are estimated with enough accuracy, MA techniques are able to modify the RTO so that the optimal solution of the model-based problem matches the NCO of the true optimal solution.

MA techniques have been thoroughly studied during the last years, especially in the context of the classical two-layer control scheme. There, the MA modifiers are implemented in the RTO layer to lean its solution towards the optimal steady-state. Most of the research has been driven by the estimation of the real system gradients [132], which are vital in the success of these methods and much progress has been achieved in this area, mainly exploiting the information of past steady-states [157, 158, 159]. To a lesser extent, methods for estimating the gradients using transient measures [131, 160] and gradient-free modifiers [60] have also been proposed, but the application of the latter is still limited because they are either slow or require knowledge/tuning of many unknown parameters.

While the research of gradient estimation techniques is still of much interest, in this chapter we put the focus on the formulation of a robust controller which is able to control the system in a single control layer and converges to the optimal steady operation. The reader is referred to chapter 7 of this document as well as the literature cited in the previous paragraph to learn how to compute the gradient-based modifiers.

6.4.1 Robust one-layer modifier-adaptation (ROLMA)

We can use the MA scheme proposed in [161] as the modifier term in (6.10). The following controller, denoted as Robust One-Layer Modifier-Adaptation (ROLMA), is able to economically control the system to a setpoint that satisfies the NCO of the real economic problem and constitutes the main contribution of this chapter.

$$
\begin{aligned}
& \mathcal{P}_N(x_k, x_k^e, u_k^e, p_k) = && \text{(6.12a)}\\
\min_{\mathbf{c},\theta} \quad & V_N(\tilde{\mathbf{x}}, \mathbf{u}, \tilde{x}^s, u^s; x_k^e, u_k^e) && \text{(6.12b)}\\
\text{s.t.} \quad & \tilde{x}_{i+1} = A\tilde{x}_i + Bu_i + \lambda_k^x \tilde{x}_i + \lambda_k^u u_i + \epsilon_k^f && \text{(6.12c)}\\
& \tilde{x}^s = A\tilde{x}^s + Bu^s + \lambda_k^x \tilde{x}^s + \lambda_k^u u^s + \epsilon_k^f && \text{(6.12d)}\\
& \tilde{x}_0 = x_k && \text{(6.12e)}\\
& \bar{x}_{i+1} = A\bar{x}_i + Bu_i && \text{(6.12f)}\\
& (\bar{x}_i, u_i) \in \overline{\mathcal{Z}}_m \ominus (\mathcal{H}(i) \times K\mathcal{H}(i)) && \text{(6.12g)}\\
& (\bar{x}_N, \theta) \in \Omega^t && \text{(6.12h)}\\
& u_i = K\bar{x}_i + L_\theta \theta + c_i && \text{(6.12i)}\\
& \bar{x}_0 = x_k && \text{(6.12j)}\\
& \text{for all } i = 0, 1, \cdots, N-1.
\end{aligned}
$$

Denote $f_p(x_i, u_i) = f_m(x_i, u_i) + d^*(x_i, u_i)$ the real plant dynamics, then the modifiers λ_k^x, λ_k^u and ϵ_k^f are computed as follows:

$$
\begin{aligned}
\lambda_{k+1}^x &= \nabla_x f_p(x_k, u_k) - \nabla_x f_m(x_k, u_k)\\
\lambda_{k+1}^u &= \nabla_u f_p(x_k, u_k) - \nabla_u f_m(x_k, u_k)\\
\epsilon_{k+1}^f &= x_{k+1} - \left(Ax_k + Bu_k + \lambda_k^x x_k + \lambda_k^u u_k + \epsilon_k^f \right).
\end{aligned}
\tag{6.13}
$$

Assumption 6.3. *Both the real system f_p and the base model f_m are differentiable on $x \in \mathcal{X}$ and $u \in \mathcal{U}$.*

The modifiers can be filtered in order to make the convergence smoother. The following remarks highlight relevant information about the ROLMA controller.

Remark 6.1. *If both $\overline{\mathcal{Z}}_m$ and $\mathcal{W}_f$ are polytopes, problem $\mathcal{P}_N$ boils down to a quadratic programming (QP). This family of problems can be solved in a low computational time with many solvers, e.g. [162, 163].*

Remark 6.2. *The modifiers do not need to be computed at each sampling time. The only requirement required to converge to the NCO of the real problem is that the modifiers eventually converge and that the gradients are perfectly computed upon convergence. The modifiers can be e.g. computed in parallel and update the system with the gradient information from several time steps prior.*

In the next section, we will go through the robustness and the performance of the proposed controllers.

6.5 Robustness and performance

In this section we prove that, upon convergence, both ROL and ROLMA controllers proposed in sections 6.3.1 and 6.4.1 reach a setpoint that matches the NCO the real control problem. First, we prove that ROL and ROLMA controllers are both recursive feasible. Then, we define the optimal economic operation and show how the modification scheme leads to it upon convergence.

Given a feasible control sequence at any given time k, the following theorem shows that a feasible control law for the ROL controller (6.10) at time $k+1$ always exists.

Theorem 6.1 (Recursive feasibility)**.** *Let* $\mathbf{v}_k^o = \{\mathbf{c}_k^o, \theta_k^o\}$ *be the sequence that defines the optimal control law computed by the ROL controller at the current step* k*, then should one apply the control sequence defined by* $\bar{\mathbf{v}}_{k+1} = \{\bar{\mathbf{c}}_{k+1}, \bar{\theta}_{k+1}\}$*:*

$$\bar{\mathbf{c}}_{k+1} = \begin{bmatrix} c^o(1|k) & c^o(2|k) & \cdots & c^o(N-1|k) & \mathbf{0} \end{bmatrix} \tag{6.14a}$$

$$\bar{\theta}_{k+1} = \theta_k^o \tag{6.14b}$$

at the next step $(k+1)$*, the system would remain feasible.*

Proof.

The recursive feasibility of the ROL controller can be derived from lemmas 1-4 from [155]. Notice that, even though the ROL controller follows a one-layer scheme with dual trajectory (robust and performance), feasibility is only determined by the constraints (6.10e)-(6.10i), which are the ones considered in [155]. The additional constraints only affect the cost function and will therefore be considered in the convergence to the NCO of the real control problem.

□

Remark 6.3. *Since the ROLMA controller is a particular case of the ROL controller, the recursive feasibility will also be guaranteed for the ROLMA controller.*

Now, we define the optimal steady operation of system (6.1) subject to constraints (6.4).

Definition 6.1 (Optimal steady operation)**.** *The optimal steady economic operation of system* (6.1) *subject to constraints* $y \in \mathcal{Y}$ *is defined as:*

$$\begin{aligned} (x^*, u^*) = \arg\min_{x,u} \quad & J(x,u) \\ s.t. \quad & x = f_m(x,u) + d^*(x,u) \\ & h_m(x_i,u_i) + d_h^*(x_i,u_i) \in \mathcal{Y}. \end{aligned} \tag{6.15}$$

Assumption 6.4 (Unique optimal solution)**.** *The optimal steady operation* (x^*, u^*) *of the real system is unique.*

Assumption 6.5 (Feasible optimal solution)**.** *The optimal steady operation* (x^*, u^*) *is contained within the robust region, i.e.*

$$(x^*, u^*) \in \overline{\mathcal{Z}}_m.$$

Definition 6.2. *The optimal value of the modifier parameter* p^* *is defined such that, for the optimal operation point, the modification* Λ^f *matches locally the real mismatch between the base model and the real system* (d^*)*. That is,* p^* *satisfies the following equations:*

$$\Lambda^f(x^*, u^*, p^*) = d^*(x^*, u^*) \tag{6.16a}$$
$$\nabla_x \Lambda^f(x^*, u^*, p^*) = \nabla_x d^*(x^*, u^*) \tag{6.16b}$$
$$\nabla_u \Lambda^f(x^*, u^*, p^*) = \nabla_u d^*(x^*, u^*). \tag{6.16c}$$

Assumption 6.6. *Given the modifier function* Λ^f *and the optimal setpoint* (x^*, u^*)*, there exists a unique value* p^* *that satisfies equations* (6.16)*.*

Assumption 6.7. *Given the optimal value of the modifier parameter* p^* *as defined by Definition 6.2, then the solution to the following real-time optimization exists and is unique:*

$$\begin{aligned} \min_{x,u} \quad & J(x,u) \\ s.t. \quad & x = f_m(x,u) + \Lambda^f(x,u,p^*) \\ & (x,u) \in \overline{\mathcal{Z}}_m. \end{aligned} \tag{6.17}$$

Theorem 6.2. *Consider assumptions 6.1-6.5 meet, then if the states of ROL* (6.10) *converge to the artificial trajectory* (x^s, u^s) *and the modifier term* $\Lambda^f(x,u,p)$ *converges locally to the mismatch at the optimal operation* $d^*(x^*, u^*)$*, then the controlled system converges to the optimal steady operation.*

Proof. The predicted performance trajectory converges to the artificial reference (x^s, u^s), so the regulation part of the cost function vanishes and the cost function can be expressed as the second order approximation of the economic function

$$\begin{aligned} V_N =& J(x_k^e, u_k^e) + \nabla_x J(x_k^e, u_k^e)^T(\tilde{x}^s - x_k^e) + \nabla_u J(x_k^e, u_k^e)^T(u^s - u_k^e) + \\ &+ \frac{\rho_x}{2}\|\tilde{x}^s - x_k^e\|^2 + \frac{\rho_u}{2}\|u^s - u_k^e\|^2. \end{aligned} \tag{6.18}$$

Given that the linearization point (x_k^e, u_k^e) follows the update policy proposed in (6.7), then when the system converges to the artificial trajectory, we have that $(x_k^e, u_k^e) = (x^s, u^s)$ and the cost function (6.18) can be further simplified as

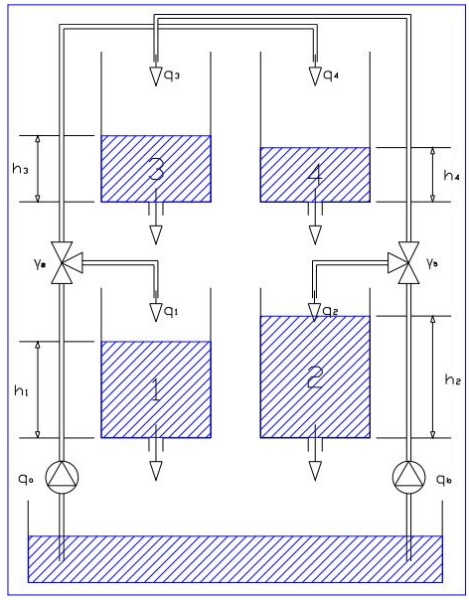


Figure 6.2: Quadruple tank system diagram, reproduced from [141].

$$V_N = J(x^s, u^s). \tag{6.19}$$

Therefore we can write problem (6.10) as

$$\begin{aligned} \min_{x^s, u^s} \quad & J(x^s, u^s) \\ \text{s.t.} \quad & x^s = f_m(x^s, u^s) + \Lambda^f(x^s, u^s, p^*) \\ & (x^s, u^s) \in \overline{\mathcal{Z}}_m. \end{aligned} \tag{6.20}$$

Taking into account assumption 6.5, the feasibility constraints have been substituted with $(x^s, u^s) \in \overline{\mathcal{Z}}_m$.

Notice that if Λ^f approximates locally the mismatch d^* at the optimal operation (x^*, u^*), then the solution of (6.20) matches that of (6.15).

□

6.6 Numerical example

In this section we revisit the quadruple tank process presented in chapter 5 to illustrate the ROLMA controller proposed in Section 6.4.1. This system consists of four interconected tanks (as shown in Figure 6.2) which share water according to the following first principle equations:

	Value	Unit	Description
S	0.06	m^2	Cross-section of the tanks
$\mathbf{a}$	$\begin{bmatrix} 1.34 \\ 1.53 \\ 0.932 \\ 0.906 \end{bmatrix} e^{-4}$	m^2	Discharge constants
$\mathbf{h}_{\max}$	$\begin{bmatrix} 1.36 \\ 1.36 \\ 1.30 \\ 1.30 \end{bmatrix}^T$	m	Maximum water level
$\mathbf{h}_{\min}$	$\begin{bmatrix} 0.3 \\ 0.3 \\ 0.3 \\ 0.3 \end{bmatrix}^T$	m	Minimum water level
$\mathbf{q}_{\max}$	$\begin{bmatrix} 3.4 \\ 3.8 \end{bmatrix}^T$	m^3/h	Maximum water flow
$\mathbf{q}_{\min}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}^T$	m^3/h	Minimum water flow
g	9.81	$\mathrm{m/s}^2$	Gravity acceleration

Table 6.1: Parameters of the quadruple tank process.

$$
\begin{aligned}
S\frac{dh_1}{dt} &= -a_1\sqrt{2gh_1} + a_3\sqrt{2gh_3} + \frac{\gamma_a q_a}{3600} \\
S\frac{dh_2}{dt} &= -a_2\sqrt{2gh_2} + a_4\sqrt{2gh_4} + \frac{\gamma_b q_b}{3600} \\
S\frac{dh_3}{dt} &= -a_3\sqrt{2gh_3} + (1-\gamma_b)\frac{q_b}{3600} \\
S\frac{dh_4}{dt} &= -a_4\sqrt{2gh_4} + (1-\gamma_a)\frac{q_a}{3600}.
\end{aligned}
$$

The parameters of the previous equations are given in a compact form by Table 6.1, where:

$$
\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix}, \mathbf{y} = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}, \mathbf{u} = \begin{bmatrix} q_a \\ q_b \end{bmatrix}, \boldsymbol{\gamma} = \begin{bmatrix} \gamma_a \\ \gamma_b \end{bmatrix}. \tag{6.21}
$$

The economic cost function to be minimized depends on the given cost parameters c and p and is given by the following formula:

$$
J(x,u) = 10(q_a^2 + cq_b^2) + p\frac{0.12}{S(h_1+h_2)}. \tag{6.22}
$$

In this example, we consider four different pairs of cost parameters to study how the ROLMA controller adapts to a changing operating point. These pairs are: $(c,p) = \{(1,35),(1.1,40),(0.8,30),(0.9,30)\}$ and will be changed each 1000 time steps. The aim of keeping the cost parameters constant during long periods of time is to bring it closer to reality, where systems usually spend the majority of the time in a steady-state rather than in a transient behaviour. Note that, due to the inaccurate nature of the base model, it can be impossible to optimize the transient behaviour.

The considered base model is a linearization of the system at the equilibrium point $x_0 = \begin{bmatrix}0.627 & 0.636 & 0.652 & 0.633\end{bmatrix}^T$, $u_0 = \begin{bmatrix}1.643 & 2\end{bmatrix}^T$ discretized with a sampling time of 10 seconds. The resulting base model is:

$$x_{k+1} = \begin{bmatrix}0.939 & 0 & 0.040 & 0\\ 0 & 0.932 & 0 & 0.040\\ 0 & 0 & 0.958 & 0\\ 0 & 0 & 0 & 0.959\end{bmatrix} x_k + \begin{bmatrix}0.0135 & 0.0006\\ 0.0007 & 0.0179\\ 0 & 0.0272\\ 0.0317 & 0\end{bmatrix} u_k.$$

As detailed in Section 6.4.1, the control actions are computed in a receding horizon fashion solving the ROLMA optimization problem (6.12) at each sampling time. The constraints of the system are box constraints given by $\mathbf{h}_{\max}, \mathbf{h}_{\min}, \mathbf{q}_{\max}$ and $\mathbf{q}_{\min}$ and the plant-model mismatch is given by

$$|W_f| \leq \begin{bmatrix}6.25 & 6.25 & 6.25 & 6.25\end{bmatrix}^T 10^{-3}.$$

This bound is a bit more conservative than the one computed in [164]. Since both $\overline{\mathcal{Z}}_m$ and $\mathcal{W}_f$ are polytopes, the ROLMA optimization problem is a QP according to Remark 6.1 and is solved using the Multi-Parametric Toolbox 3.0 [165]. The numerical integration of the real dynamics is done with the Sundials Suite [166].

The value of the gain K is:

$$K = \begin{bmatrix}-5.9997 & -18.7429 & 6.2544 & -37.0666\\ -20.6414 & -12.8487 & -29.7042 & -3.0337\end{bmatrix}, \tag{6.23}$$

and the control horizon is set to $N = 5$.

The response of the system controlled by both the ROLMA controller and the ROL controller without any modifier term is presented in Figure 6.3. Figure 6.4 shows a measure of the economic cost associated to both controllers. This measure stands for the cumulative economic cost of the controlled systems minus the economic cost of the optimal setpoint. Even though both controllers take into account the economic cost of the transient, the results show how, using ROLMA, the system converges to the economic optimum, while when the modifiers are taken out (non modified ROL), the system reaches a non optimal steady-state. This means that even if the transient cost may be lower, suboptimal steady behaviour turns into steady losses that accumulate through time.

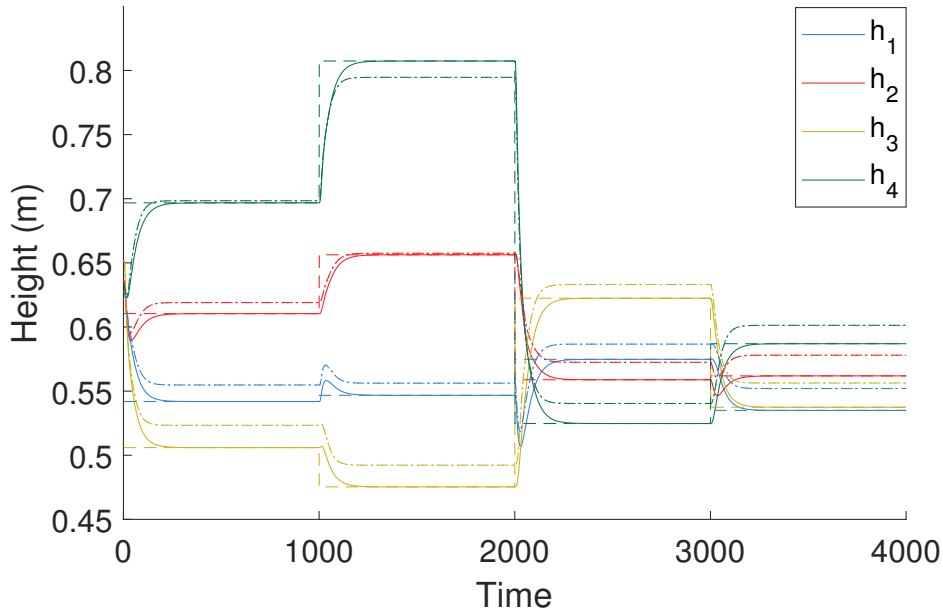


Figure 6.3: Response of the controlled systems. The optimal steady-state is represented with a dashed line, the ROLMA response with a solid line and the ROL without modifiers response with a dot-dash line. Each state is assigned a different colour.

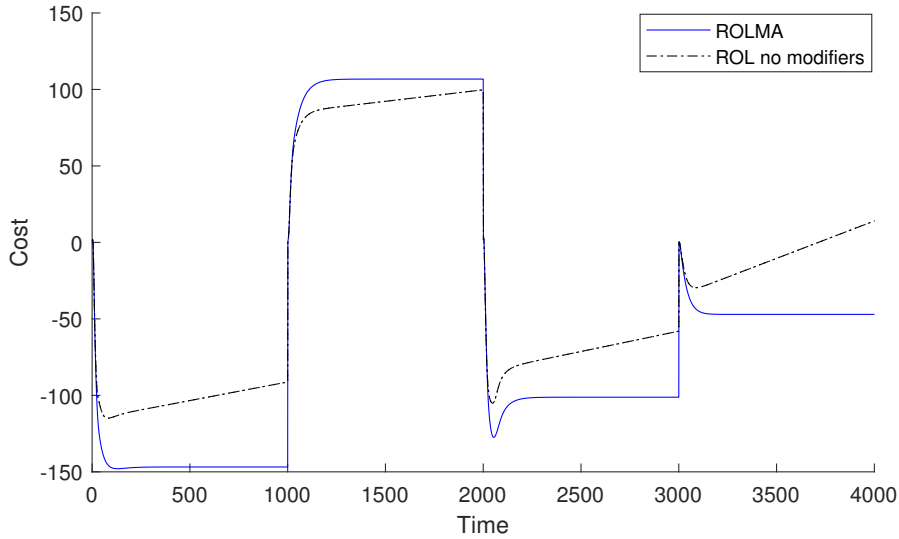


Figure 6.4: Cumulative economic cost of the controlled systems minus the cost of the optimal setpoint. The cumulative cost is reset each time the economic parameters are changed.

6.7 Conclusions

In this chapter we have presented an integrated approach that controls a system to its economic steady optimal operation. Given a (possibly inaccurate) dynamical model of the system and a method to compute gradients of the real plant dynamics, the proposed approach is able to control the system to its optimal steady operation by solving a single optimization problem (which boils down to a quadratic programming when affine modification terms are used). This opens the door to the economic control of quick systems without sacrificing the performance of the steady-state setpoint. The proposed controllers denoted ROL and ROLMA are proven to converge to the optimal steady operation and be recursive feasible under certain circumstances. Tested against the quadruple tank system, ROLMA control shows an improved performance compared to the non modified approach.

Chapter 7

Digital Twins in Modifier-Adaptation Schemes

7.1 Introduction

Digital twins have experienced a surge in popularity in recent years as an integral component of Industry 4.0. The number of scientific publications focusing on digital twins is exponentially rising, leaded by the growing interest and presence in the industry [167]. Digital twin is a framework which contains, among others elements, a digital replica of the real physical system. They can include measure information and are used for multiple purposes, e.g. design of the final system, anomaly detection, optimization... and in plenty of fields, e.g. biochemistry, aeronautical, mechanical... [168]

The core of digital twins lies in a digital framework embedded into the real process (physical entity) which takes continuous measures of the real system, and updates the virtual entity through process analytical technologies (PAT) and advanced mathematical modeling tools. Not only the physical entity has a virtual equivalent, but also does the physical environment. This guarantees that the virtual process describes perfectly the physical one.

In this chapter, we do not focus on building and updating the digital twins. Instead, we consider them available and we take advantage of their ability to accurately reflect the real system behaviour in a virtual process.

Modifier-adaptation (MA) schemes [9] arose to improve the calculation of the optimal steady operation. As commented in previous chapters, these methodologies update the RTO with affine terms, which generally include information of the dynamics and gradients of the real plant. Given that this information is accurate, modifier-adaptation schemes are able to converge to the optimal steady operation of the real plant. However, the practical unavailability of the real plant gradients usually makes MA methodologies struggle and resort to alternatives to approximate them. These approximations based on empirical data generally bring further issues. Reformulations of the real-time optimizer to excite the system and

estimate gradients can make the problem more complex and the control slower. Furthermore, under the presence of disturbances, gradient estimation often requires filtering and can be both slow and inaccurate. These issues exacerbate as the number of gradients to estimate increases.

The aim of this chapter is to study how modifier-adaptation schemes can take advantage of the digital twin framework and overcome their main challenges.

7.2 Modifier-Adaptation

As shown in previous chapters, modifier-adaptation schemes update the real-time optimization problem with affine terms in order to make its optimal solution converge to the necessary conditions of optimality of the real plant steady operation.

Over the years, multiple formulations based on modifier-adaptation have been proposed. In this chapter, we recapitulate three of them, which require modifiers with different plant and gradient information: standard MA, output MA and periodic MA. These RTO formulations are simple and do not include the gradient computation, so that modifiers can be computed independently.

Let the inputs, states and outputs of a system be given by $u \in \mathbb{R}^{n_u}, x \in \mathbb{R}^{n_x}$ and $y \in \mathbb{R}^{n_y}$ respectively. Consider the economic cost function $J : \mathbb{R}^{n_y \times n_u} \to \mathbb{R}$ and the constraint function $g : \mathbb{R}^{n_y \times n_u} \to \mathbb{R}$ such that every admissible pair (y, u) satisfies $g(y, u) \leq 0$. Let $f_p^s : \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$ be a function that maps every feasible steady input into its correspondent steady output, and define the steady cost function $\Phi_p(u) = J(f_p^s(u), u)$ and the steady constraint $G_p(u) = g(f_p^s(u), u)$. Then, the optimal steady input (u_{opt}^s) for the plant can be calculated as

$$\begin{aligned} u_{\text{opt}}^s = \arg\min_{u} \quad & \Phi_p(u) \\ \text{s.t.} \quad & G_p(u) \leq 0 \\ & u^L \leq u \leq u^U, \end{aligned} \tag{7.1}$$

where u^L and u^U are the lower and upper bounds respectively for the steady input.

Problem (7.1) involves the knowledge of f_p^s, which is usually unknown. Consequently, the cost and constraint functions Φ_p and G_p are also unknown. Modifier-adaptation schemes presented in the remainder of this section use a model of the real system and reformulate problem (7.1) with modifier terms built from information of the real system. Under certain assumptions, the model-based modified problem converges to the optimal steady operation.

7.2.1 Standard MA (MA)

First, we cover the standard formulation proposed in [9].

Let f_m^s be the model for f_p^s. Then, models for Φ_p and G_p can be calculated as

$$\begin{aligned} \Phi_m(u) &= J(f_m^s(u), u) \\ G_m(u) &= g(f_m^s(u), u) \end{aligned} \tag{7.2}$$

respectively. The standard MA RTO is formulated as follows

$$\begin{aligned} u_{\text{ma}}^s = \arg\min_u \quad & \Phi_m(u) + \lambda_k^\Phi u \\ \text{s.t.} \quad & G_m(u) + \lambda_k^G(u - u_k) + \epsilon_k^G \leq 0 \\ & u^L \leq u \leq u^U. \end{aligned} \tag{7.3}$$

At each time k, the modifiers $\lambda_k^\Phi, \lambda_k^G$ and ϵ_k^G change the problem in an adaptive fashion. It is proven that, under certain assumptions, if the modifiers are computed as

$$\begin{aligned} \epsilon_{k+1}^G &= G_p(u_k) - G_m(u_k) \\ \lambda_{k+1}^G &= \frac{\partial G_p}{\partial u}(u_k) - \frac{\partial G_m}{\partial u}(u_k) \\ \lambda_{k+1}^\Phi &= \frac{\partial \Phi_p}{\partial u}(u_k) - \frac{\partial \Phi_m}{\partial u}(u_k), \end{aligned} \tag{7.4}$$

then the optimal solution of (7.3) converges to the one of (7.1) as $k \to \infty$.

7.2.2 Output MA (MAy)

In MAy [9], the modifiers update the model of the static characteristic f_m^s instead of the cost and constraints (which are also updated, but in an indirect fashion). Therefore, at each time step k, we define

$$\begin{aligned} f_{m,k}^s(u) &= f_m^s(u) + \lambda_k^y(u - u_k) + \epsilon_k^y \\ \Phi_{m,k}(u) &= J(f_{m,k}^s(u), u) \\ G_{m,k}(u) &= g(f_{m,k}^s(u), u), \end{aligned} \tag{7.5}$$

where the modifiers are computed as

$$\begin{aligned} \epsilon_{k+1}^y &= f_p^s(u_k) - f_m^s(u_k) \\ \lambda_{k+1}^y &= \frac{\partial f_p^s}{\partial u}(u_k) - \frac{\partial f_m^s}{\partial u}(u_k), \end{aligned} \tag{7.6}$$

and the MAy RTO is formulated as

$$\begin{aligned} u_{\text{may}}^s = \arg\min_u \quad & \Phi_{m,k}(u) \\ \text{s.t.} \quad & G_{m,k}(u) \leq 0 \\ & u^L \leq u \leq u^U. \end{aligned} \tag{7.7}$$

Uncer certain assumptions, the optimal solution of (7.7) converges to that of (7.1) as $k \to \infty$.

7.2.3 Periodic MA (P-MA)

The last formulation of MA considers the case where the optimal steady operation is a periodic trajectory, covered in chapter 5. Given the current state x_k and the next input u_k, the unknown function $f_{p,k} : \mathbb{R}^{n_x \times n_u} \to \mathbb{R}^{n_x}$ defines the state of the system at step $k+1$ and $h : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ corresponds to the relation between the states x_k and the outputs y_k, which for the sake of simplicity we assume to be known.

$$\begin{aligned} x_{k+1} &= f_{p,k}(x_k, u_k) \\ y_k &= h(x_k). \end{aligned} \tag{7.8}$$

Let the system (7.8) be periodic with period T. Then, at $k = 0$, given the initial state x_0 and the sequence of T next inputs $\mathbf{u}_T = \begin{bmatrix} u_0^T & u_1^T & \cdots & u_{T-1}^T \end{bmatrix}^T$, the function $F_p : \mathbb{R}^{n_x \times T n_u} \to \mathbb{R}^{T n_x}$ describes the T future states of the system and is built so that

$$\mathbf{x}_T = \begin{bmatrix} x_1^T & x_2^T & \dots & x_T^T \end{bmatrix}^T = F_p(x_0, \mathbf{u}_T).$$

Similar to (7.1), let the optimal periodic trajectory of period T be calculated as

$$\begin{aligned} \{x_0^{\text{opt}}, \mathbf{x}_T^{\text{opt}}, \mathbf{u}_T^{\text{opt}}\} = & \\ \underset{x_0, \mathbf{x}_T, \mathbf{u}_T}{\arg\min} \quad & \sum_{i=0}^{T-1} J(y_i, u_i) \\ \text{s.t.} \quad & \mathbf{x}_T = F_p(x_0, \mathbf{u}_T) \\ & y_i = h(x_i), \quad \text{for all } i = 0, \cdots, T-1 \\ & g(y_i, u_i) \leq 0, \quad \text{for all } i = 0, \cdots, T-1 \\ & u^L \leq u_i \leq u^U, \quad \text{for all } i = 0, \cdots, T-1 \\ & x_0 = x_T. \end{aligned} \tag{7.9}$$

Let F_m be an available model of F_p, then the P-MA dynamic RTO is formulated as follows

$$\begin{aligned} \{x_{0,k}^{\text{pma}}, \mathbf{x}_{T,k}^{\text{pma}}, \mathbf{u}_{T,k}^{\text{pma}}\} = & \\ \underset{x_0, \mathbf{x}_T, \mathbf{u}_T}{\arg\min} \quad & \sum_{i=0}^{T-1} J(y_i, u_i) \\ \text{s.t.} \quad & \mathbf{x}_T = F_m(x_0, \mathbf{u}_T) + \lambda_k^x (x_0 - x_{0,k-1}^{\text{pma}}) + \lambda_k^u (\mathbf{u}_T - \mathbf{u}_{T,k-1}^{\text{pma}}) + \epsilon_k \\ & y_i = h(x_i), \quad \text{for all } i = 0, \cdots, T-1 \\ & g(y_i, u_i) \leq 0, \quad \text{for all } i = 0, \cdots, T-1 \\ & u^L \leq u_i \leq u^U, \quad \text{for all } i = 0, \cdots, T-1 \\ & x_0 = x_T, \end{aligned} \tag{7.10}$$

where the modifiers ϵ_k, λ_k^x and λ_k^u are computed at each iteration as

$$\begin{aligned}
\epsilon_{k+1} &= F_p(x_{0,k}^{\text{pma}}, \mathbf{u}_{T,k}^{\text{pma}}) - F_m(x_{0,k}^{\text{pma}}, \mathbf{u}_{T,k}^{\text{pma}}) \\
\lambda_{k+1}^x &= \frac{\partial F_p}{\partial x_0}(x_{0,k}^{\text{pma}}, \mathbf{u}_{T,k}^{\text{pma}}) - \frac{\partial F_m}{\partial x_0}(x_{0,k}^{\text{pma}}, \mathbf{u}_{T,k}^{\text{pma}}) \\
\lambda_{k+1}^u &= \frac{\partial F_p}{\partial \mathbf{u}_T}(x_{0,k}^{\text{pma}}, \mathbf{u}_{T,k}^{\text{pma}}) - \frac{\partial F_m}{\partial \mathbf{u}_T}(x_{0,k}^{\text{pma}}, \mathbf{u}_{T,k}^{\text{pma}}),
\end{aligned} \tag{7.11}$$

and $\{x_{0,k}^{\text{pma}}, \mathbf{x}_{T,k}^{\text{pma}}, \mathbf{u}_{T,k}^{\text{pma}}\}$ corresponds to the optimal solution of the dynamic RTO (7.10) at step k. Under certain assumptions, the optimal solution of (7.10) converges to the optimal steady behaviour from (7.9) as $k \to \infty$.

7.2.4 Issues

All the previous approaches involve the computation of gradients of both the real system and the model. On the one hand, obtaining gradients of the model is generally easy, since they can be computed by numerical or symbolic procedures. The model typically describes the nominal behaviour of the system, which means that noise is not present and unmeasured disturbances are estimated. Therefore, they do not interfere in the computation of the gradients. On the other hand, gradients of the real plant are usually estimated from online measurements, which brings some challenges. Approximating the real plant gradients and proposing alternative formulations to bypass gradient estimation constitute two major pillars of the research in modifier-adaptation [59, 159, 60, 10].

These lines of work ultimately solve the problem of estimating the modifiers. However, because of the fact that they are based on measures taken straight from the real system, they introduce some issues:

- Gradient approximation based on measured data is usually sensitive to noise.
- Some states of the system may not be measurable, so accurate estimation of their gradients is seldom possible.
- The computation of the gradients requires sufficient excitation of the system.
- To guarantee that the system is excited enough in the next step, usually more complex formulations of the control problem are presented [59].
- Gradient-free methodologies involve prior tuning of parameters which do not hold any physical meaning and are unknown to the user [60].
- The need of an estimation step may lead to a slower control.
- Gradient estimation based on data measurements does not scale well to large systems (with many inputs, states and outputs).

In the next section, we propose to use the digital twin framework to obtain gradient information and compute the modifiers without any of the aforementioned issues.

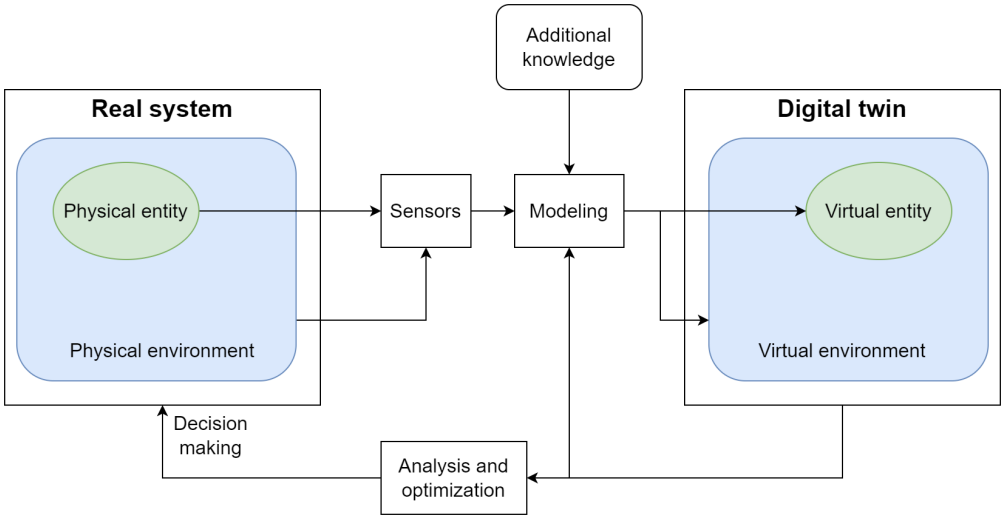


Figure 7.1: Scheme of the digital twin.

7.3 Digital Twin for gradient computation

The fourth industrial revolution, also known as Industry 4.0, describes the latest trend in manufacturing systems [169]. It takes advantage of the progress made in multiple fields like Cyber-Physical Systems (CPS), Artificial Intelligence (AI) or Internet of Things (IoT) and use it to optimize the design, production and technical support of the products. The digital transformation described in the Industry 4.0 paradigm has already made its way into the industry, and major companies like IBM, Airservices Australia or Bridgestone have embraced it and made substantial investments in it [170, 171].

One of the key components of Industry 4.0 is the digital twin framework, which is the focus of this chapter. First, we give a formal definition to the concept of digital twin based on the ones given in [167, 172].

Definition 7.1. *Digital twin is a framework in which a virtual entity is created as a replica of the physical system via process analytical technologies (PAT) and other advanced mathematical tools. In this framework, virtual and physical entities are linked together though the entire lifetime of the system and share data to ensure that the virtual process remains identical to the physical one, including the system's degradation.*

Figure 7.1 shows the main structure of the digital twin framework. In this scheme, the digital twin is constituted by the virtual entity and environment, which are modeled using: 1) information of the sensors installed in the real system, 2) information of the last known iteration of the digital twin and 3) additional knowledge of the system, such as first principle equations or physical bounds. The resulting digital twin can be used to analyze and test the system without

compromising the stability and performance of the real system. Therefore, it is possible to test on the digital twin changes (such as control formulations) prior to implementing them into the real system, since it acts as a perfect replica.

The digital twin framework has proliferated in the industry over the last years [171, 173]. In terms of revenue, Marketsandmarkets estimates that digital twin market was worth USD 6.9 billion in 2022 and is expected to reach 73.5 billion by 2027 [174]. Today, digital twins are present in many sectors (e.g. automotive, health care, aircraft, robotics...) and their applications are diverse, ranging from prototyping to interrogating the virtual processes about their current states.

Since the digital twin framework is already adopted in many industry applications, we can use the existing infrastructure and take advantage of its ability to reflect and predict the states and gradients of the physical system. Furthermore, in the context of automatic control, digital twins can also be useful in other tasks such as model identification or testing the controllers. In systems where a digital twin is available, neglecting its information and using alternative methods to estimate the gradients of the real system, identify models and test controllers may not only be less accurate, but also less efficient.

Contrary to state-of-the-art gradient estimation, obtaining the gradients from the digital twin can be considered exact given that the fidelity of the digital twin is high enough (Definition 7.1). Now, we introduce two assumptions required to use the digital twin in modifier-adaptation schemes.

Assumption 7.1. *The states, outputs and economic cost of the RTO model, as well as their gradients with respect of the model inputs are available on the virtual entity.*

Remark 7.1. *Assumption 7.1 meets e.g. when the states of the RTO model are autoregressive or have a physical meaning.*

Assumption 7.2. *Simulations on the virtual entity are made under nominal conditions.*

Note that no assumption on the physical system is made. While only assumption 7.1 is required for standard MA approaches, for more intricate schemes like the Periodic MA, the assumption 7.2 can also be necessary to guarantee that the prediction includes only the nominal behaviour and consequently the noise does not interfere with the computation of gradients.

The virtual process is parallel to the control scheme. Therefore, obtaining the gradient information from it does not result in any slowdown in the control. Furthermore, the gradient calculation task can be carried out in the background, while the virtual process remains linked to the operating physical one.

7.3.1 Benefits

Using the virtual process of the digital twin allows to obtain the gradients required for the modifier-adaptation schemes presented in this dissertation without any of

the issues presented in Section 7.2.4:

- Gradient information obtained from the virtual process is not sensitive to noise.
- The virtual process includes all the information from the physical process, so it is possible to obtain their associated gradients.
- The computation of the gradients is independent from the excitation of the system.
- There is no need to change the original formulation of the control problem.
- The modifiers are still based on gradients, so there is no need to tune gradient-free cumbersome parameters.
- Obtaining the gradients is done in parallel with the control scheme, which does not slow down the control.
- Consulting the real plant information through its virtual counterpart scales well to large systems.

In systems where a digital twin is available, it is straighforward to use it to calculate the gradients of the real system in modifier-adaptation schemes. On the other hand, building a digital twin from the ground up in systems where it is not yet implemented is also a reasonable alternative. Compared to state-of-the-art modifier-adaptation formulations, the digital twin framework does not only improve the performance and efficiency of gradient calculation, but it also serves for many different purpuses such as analysis or anomaly detection.

7.4 Conclusions

In this chapter we have reviewed three formulations of modifier-adaptation. While all of them can theoretically compute the optimal steady operation of a system, they require information about the gradients of the real systems, which gives rise to a number of problems.

We take advantage of the digital transformation of industry 4.0 and the digital twin framework, which is already present in many industries, and propose a workaround to gradient estimation, using the information available in the digital twins. This new paradigm applies directly to many modifier-adaptation schemes and replaces the gradient estimation step. As a result, it solves the main issues associated with modifier-adaptation techniques, leading to a more accurate and efficient gradient estimation.

Besides the gradient estimation in real-time optimization, the digital twin framework may also be used for a wide variety of applications, such as anomaly detection or system modification design. This versatility makes the digital twin

framework an appealing alternative to state-of-the-art modifier-adaptation approaches.

Chapter 8

Conclusions and Future Lines

This chapter provides a recapitulation the thesis contributions while also delineating potential future research direction to advance the work presented in this dissertation.

8.1 Summary of contributions

The contributions of this thesis can be categorized in two different fields: safe region estimation and automatic control under plant-model mismatch. These fields represent major pillars within the automatic control research, where extensive study has been carried out for decades, highlighting their foundational relevance. In what follows, the contributions of this thesis to both of these fields are stated:

- In chapter 2, a series of methods to bound the absolute value of the prediction error have been presented. They all share a probabilistic maximization scheme that uses available samples of the systems to set probabilistic bounds on the prediction error. The different methods presented in this chapter vary in the size of the bound, which can be fixed or varying, and the number of predictors taken into account. Contrary to other state-of-the-art methods, the ones presented in this chapter do not depend on the complexity of the prediction model.

- In chapter 3, a general sample-based approach to calculate inner approximations of the chance-constrained set has been presented. This approach called classical probabilistic scaling is able to scale a set of reduced complexity until it is contained in the desired chance-constrained set with a given confidence. The required number of samples depends only on the desired probabilistic guarantees, and not on the dimension of the problem or the probabilistic distribution of the uncertainty.

- In chapter 4, an extension to the classical probabilistic scaling called pack-based probabilistic scaling has been presented. This new approach is able to

not only calculate inner approximations of the chance-constrained set, but it also guarantees a degree of tightening. A branch-and-bound heuristic to compute the pack-based probabilistic scaling is also detailed.

- In chapter 5, a scheme that optimizes the steady behaviour of periodic systems under plant-model mismatch has been presented. This scheme is a generalization of modifier-adaptation schemes to the case of periodic optimal behaviour and uses affine periodic modifiers to update the optimization problem.

- In chapter 6, a robust integrated scheme that controls systems under plant-model mismatch to their optimal setpoints has been presented. Unlike other state-of-the-art solutions, the proposed approach integrates the control in a single layer that optimizes the target setpoint and computes the control sequence at the same time.

- In chapter 7, the use of the digital twin framework in modifier-adaptation schemes has been proposed. This framework makes it trivial to obtain the gradients of the real systems at any operating points and can therefore potentially solve the main challenges of modifier-adaptation schemes.

8.2 Future research

In the previous section we have reviewed the contributions of this thesis to the fields of safe region estimation and automatic control under plant-model mismatch. Now, we will go through what can be built upon the foundations presented in this dissertation.

The first major contribution of this work has been the proposal of a series of methods to probabilistically bound the prediction error, which can be used to make intervalar predictions. However, the classification problem has not been included in this work. Extending the prediction error quantification to the multiclass prediction problem is possible and rather straightforward and remains an interesting line to explore.

This work constitutes the birth of the probabilistic scaling methodology. Major advances have been made in this topic, yet, there is still much research to be done to exploit its full potential. As for the simple approximating sets (SAS), we have proposed a few families with simple shapes that should perform well in many problems. However, it would be interesting to study new custom initial geometries which may potentially outperform the proposed ones for specific problems.

In the probabilistic scaling, some other interesting research lines would be the extension to mixed-integer programming (MIP) and the extension of the pack-based probabilistic scaling heuristic to nonlinear setups. The improvement of the probabilistic scaling algorithm would also be very beneficial and could lead to better tightenings of the approximations of the chance-constrained sets.

As for modifier-adaptation, it has been proved that upon convergence, the optimal setpoint or steady periodic trajectory may be reached. Future work could focus on the convergence rate to that steady behaviour and the conditions under which the modified system converges. Furthermore, the proposal of digital twins to obtain gradient information useful for modifier-adaptation methodologies has been purely theoretical. Its use in real applications merits attention in future investigations.

Finally, future reseach could also explore the use of the advances in uncertainty quantification and safe sets approximations in robust schemes such as the ROLMA controller presented in chapter 6.

Bibliography

[1] Z.-H. Zhou, *Machine learning*. Springer Nature, 2021.

[2] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR).[Internet]*, vol. 9, no. 1, pp. 381–386, 2020.

[3] N. C. Giri, *Multivariate Statistical Analysis*. Marcel Dekker, second, revised and expanded ed., 2004.

[4] S. M. Idrees, M. A. Alam, and P. Agarwal, "A prediction approach for stock market volatility based on time series data," *IEEE Access*, vol. 7, pp. 17287–17298, 2019.

[5] G. Alfonso, A. D. Carnerero, D. R. Ramirez, and T. Alamo, "Stock forecasting using local data," *IEEE Access*, vol. 9, pp. 9334–9344, 2020.

[6] G. Falkovich, *Fluid Mechanics: A Short Course for Physicists*. Cambridge: Cambridge University Press, 2011.

[7] T. H. Broholt, M. D. Knudsen, and S. Petersen, "The robustness of black and grey-box models of thermal building behaviour against weather changes," *Energy and Buildings*, vol. 275, p. 112460, 2022.

[8] J. A. Borja-Conde, K. Witheephanich, J. Coronel, and D. Limon, "Thermal modeling of existing buildings in high-fidelity simulators: A novel, practical methodology," *Energy and Buildings*, vol. 292, p. 113127, 2023.

[9] A. Marchetti, B. Chachuat, and D. Bonvin, "Modifier-adaptation methodology for real-time optimization," *Industrial & engineering chemistry research*, vol. 48, no. 13, pp. 6022–6033, 2009.

[10] A. Marchetti, G. François, T. Faulwasser, and D. Bonvin, "Modifier adaptation for real-time optimization—methods and applications," *Processes*, vol. 4, no. 4, p. 55, 2016.

[11] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control*, pp. 207–226, Springer, 1999.

[12] D. Q. Mayne, M. M. Seron, and S. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.

[13] D. Q. Mayne, S. V. Raković, R. Findeisen, and F. Allgöwer, "Robust output feedback model predictive control of constrained linear systems," *Automatica*, vol. 42, no. 7, pp. 1217–1222, 2006.

[14] M. G. Safonov, "Origins of robust control: Early history and future speculations," *Annual Reviews in Control*, vol. 36, no. 2, pp. 173–181, 2012.

[15] D. Chatterjee, P. Hokayem, and J. Lygeros, "Stochastic receding horizon control with bounded control inputs: A vector space approach," *IEEE Transactions on Automatic Control*, vol. 56, no. 11, pp. 2704–2710, 2011.

[16] D. Chatterjee and J. Lygeros, "On stability and performance of stochastic predictive control techniques," *IEEE Transactions on Automatic Control*, vol. 60, no. 2, pp. 509–514, 2014.

[17] M. Lorenzen, F. Dabbene, R. Tempo, and F. Allgöwer, "Constraint-tightening and stability in stochastic model predictive control," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3165–3177, 2016.

[18] M. Farina, L. Giulioni, and R. Scattolini, "Stochastic linear model predictive control with chance constraints–a review," *Journal of Process Control*, vol. 44, pp. 53–67, 2016.

[19] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Systems Magazine*, vol. 36, no. 6, pp. 30–44, 2016.

[20] A. Charnes and W. Cooper, "Chance constraints and normal deviates," *Journal of the American statistical association*, vol. 57, no. 297, pp. 134–148, 1962.

[21] H. Wang, W. H. Lam, X. Zhang, and H. Shao, "Sustainable transportation network design with stochastic demands and chance constraints," *International Journal of Sustainable Transportation*, vol. 9, no. 2, pp. 126–144, 2015.

[22] D. S. Chang and C. S. Tsou, "A chance-constraints linear programming model on the economic evaluation of flexible manufacturing systems," *Production Planning & Control*, vol. 4, no. 2, pp. 159–165, 1993.

[23] X. Geng and L. Xie, "Data-driven decision making in power systems with probabilistic guarantees: Theory and applications of chance-constrained optimization," *Annual reviews in control*, vol. 47, pp. 341–363, 2019.

[24] B. L. Miller and H. M. Wagner, "Chance constrained programming with joint constraints," *Operations Research*, vol. 13, no. 6, pp. 930–945, 1965.

[25] S. Kataoka, "A stochastic programming model," *Econometrica: Journal of the Econometric Society*, pp. 181–196, 1963.

[26] A. Prékopa, *Stochastic programming*, vol. 324. Springer Science & Business Media, 2013.

[27] A. Prékopa, "Logarithmic concave measures with application to stochastic programming," *Acta Scientiarum Mathematicarum*, vol. 32, pp. 301–316, 1971.

[28] W. van Ackooij, "Eventual convexity of chance constrained feasible sets," *Optimization*, vol. 64, no. 5, pp. 1263–1284, 2015.

[29] C. M. Lagoa, "On the convexity of probabilistically constrained linear programs," in *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No. 99CH36304)*, vol. 1, pp. 516–521, IEEE, 1999.

[30] G. C. Calafiore and L. E. Ghaoui, "On distributionally robust chance-constrained linear programs," *Journal of Optimization Theory and Applications*, vol. 130, pp. 1–22, 2006.

[31] R. Henrion and C. Strugarek, "Convexity of chance constraints with independent random variables," *Computational Optimization and Applications*, vol. 41, no. 2, pp. 263–276, 2008.

[32] P. Beraldi and A. Ruszczyński, "A branch and bound method for stochastic integer problems under probabilistic constraints," *Optimization Methods and Software*, vol. 17, no. 3, pp. 359–382, 2002.

[33] D. Dentcheva, A. Prékopa, and A. Ruszczynski, "Concavity and efficient points of discrete distributions in probabilistic programming," *Mathematical programming*, vol. 89, pp. 55–77, 2000.

[34] A. Prékopa, "Dual method for the solution of a one-stage stochastic programming problem with random rhs obeying a discrete probability distribution," *Zeitschrift für Operations Research*, vol. 34, pp. 441–461, 1990.

[35] L. G. Khachiyan, "The problem of calculating the volume of a polyhedron is enumerably hard," *Russian Mathematical Surveys*, vol. 44, no. 3, p. 199, 1989.

[36] L. Hewing and M. N. Zeilinger, "Stochastic model predictive control for linear systems using probabilistic reachable sets," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 5182–5188, IEEE, 2018.

[37] S. Yan, P. Goulart, and M. Cannon, "Stochastic model predictive control with discounted probabilistic constraints," in *2018 European Control Conference (ECC)*, pp. 1003–1008, IEEE, 2018.

[38] A. Ben-Tal and A. Nemirovski, "Robust convex optimization," *Mathematics of operations research*, vol. 23, no. 4, pp. 769–805, 1998.

[39] A. Nemirovski and A. Shapiro, "Convex approximations of chance constrained programs," *SIAM Journal on Optimization*, vol. 17, no. 4, pp. 969–996, 2007.

[40] A. Nemirovski, "On safe tractable approximations of chance constraints," *European Journal of Operational Research*, vol. 219, no. 3, pp. 707–718, 2012.

[41] W. Chen, M. Sim, J. Sun, and C.-P. Teo, "From cvar to uncertainty set: Implications in joint chance-constrained optimization," *Operations research*, vol. 58, no. 2, pp. 470–485, 2010.

[42] I. Bremer, R. Henrion, and A. Möller, "Probabilistic constraints via sqp solver: Application to a renewable energy management problem," *Computational Management Science*, vol. 12, no. 3, pp. 435–459, 2015.

[43] A. M. Jasour, N. S. Aybat, and C. M. Lagoa, "Semidefinite programming for chance constrained optimization over semialgebraic sets," *SIAM Journal on Optimization*, vol. 25, no. 3, pp. 1411–1440, 2015.

[44] J. B. Lasserre, "Representation of chance-constraints with strong asymptotic guarantees," *IEEE Control Systems Letters*, vol. 1, no. 1, pp. 50–55, 2017.

[45] S. Ahmed and W. Xie, "Relaxations and approximations of chance constraints under finite distributions," *Mathematical Programming*, vol. 170, pp. 43–65, 2018.

[46] M. A. Lejeune, "Pattern-based modeling and solution of probabilistically constrained optimization problems," *Operations research*, vol. 60, no. 6, pp. 1356–1372, 2012.

[47] M. A. Lejeune and F. Margot, "Solving chance-constrained optimization problems with stochastic quadratic inequalities," *Operations Research*, vol. 64, no. 4, pp. 939–957, 2016.

[48] J. Luedtke and S. Ahmed, "A sample approximation approach for optimization with probabilistic constraints," *SIAM Journal on Optimization*, vol. 19, no. 2, pp. 674–699, 2008.

[49] T. Alamo, R. Tempo, and E. F. Camacho, "Randomized strategies for probabilistic solutions of uncertain feasibility and optimization problems," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2545–2559, 2009.

[50] G. C. Calafiore, F. Dabbene, and R. Tempo, "Research on probabilistic methods for control system design," *Automatica*, vol. 47, no. 7, pp. 1279–1293, 2011.

[51] R. Tempo, G. Calafiore, F. Dabbene, *et al.*, *Randomized algorithms for analysis and control of uncertain systems: with applications*, vol. 7. Springer, 2013.

[52] G. C. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Transactions on automatic control*, vol. 51, no. 5, pp. 742–753, 2006.

[53] M. C. Campi and S. Garatti, "The exact feasibility of randomized solutions of uncertain convex programs," *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1211–1230, 2008.

[54] G. C. Calafiore, "Random convex programs," *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 3427–3464, 2010.

[55] M. C. Campi and S. Garatti, "A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality," *Journal of optimization theory and applications*, vol. 148, no. 2, pp. 257–280, 2011.

[56] W. Findeisen, F. N. Bailey, M. Brdys, K. Malinowski, P. Tatjewski, and A. Wozniak, *Control and coordination in hierarchical systems*. John Wiley & Sons, 1980.

[57] P. Roberts, "An algorithm for steady-state system optimization and parameter estimation," *International Journal of Systems Science*, vol. 10, no. 7, pp. 719–734, 1979.

[58] A. Marchetti, B. Chachuat, and D. Bonvin, "A dual modifier-adaptation approach for real-time optimization," *Journal of Process Control*, vol. 20, no. 9, pp. 1027–1037, 2010.

[59] W. Gao, S. Wenzel, and S. Engell, "A reliable modifier-adaptation strategy for real-time optimization," *Computers & chemical engineering*, vol. 91, pp. 318–328, 2016.

[60] D. Navia, L. Briceño, G. Gutiérrez, and C. De Prada, "Modifier-adaptation methodology for real-time optimization reformulated as a nested optimization problem," *Industrial & engineering chemistry research*, vol. 54, no. 48, pp. 12054–12071, 2015.

[61] R. G. McClarren, P. McClarren, and R. Penrose, *Uncertainty quantification and predictive computational science*. Springer, 2018.

[62] A. F. Villaverde, E. Raimúndez, J. Hasenauer, and J. R. Banga, "A comparison of methods for quantifying prediction uncertainty in systems biology," *IFAC-PapersOnLine*, vol. 52, no. 26, pp. 45–51, 2019.

[63] P.-J. Meyer, A. Devonport, and M. Arcak, *Interval reachability analysis: Bounding trajectories of uncertain systems with boxes for control and verification*. Springer Nature, 2021.

[64] R. Koenker and K. F. Hallock, "Quantile regression," *Journal of economic perspectives*, vol. 15, no. 4, pp. 143–156, 2001.

[65] M. C. Campi, G. Calafiore, and S. Garatti, "Interval predictor models: Identification and reliability," *Automatica*, vol. 45, no. 2, pp. 382–392, 2009.

[66] S. Garatti, M. C. Campi, and A. Care, "On a class of interval predictor models with universal reliability," *Automatica*, vol. 110, p. 108542, 2019.

[67] V. Balasubramanian, S.-S. Ho, and V. Vovk, *Conformal prediction for reliable machine learning: theory, adaptations and applications*. Newnes, 2014.

[68] G. Shafer and V. Vovk, "A tutorial on conformal prediction.," *Journal of Machine Learning Research*, vol. 9, no. 3, 2008.

[69] T. Alamo, J. Manzano, and E. Camacho, "Robust design through probabilistic maximization," in *Uncertainty in Complex Networked Systems*, pp. 247–274, Springer, 2018.

[70] M. Mammarella, T. Alamo, F. Dabbene, and M. Lorenzen, "Computationally efficient stochastic MPC: A probabilistic scaling approach," in *2020 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 25–30, IEEE, 2020.

[71] V. Mirasierra, M. Mammarella, F. Dabbene, and T. Alamo, "Prediction error quantification through probabilistic scaling," *IEEE Control Systems Letters*, vol. 6, pp. 1118–1123, 2021.

[72] R. Tempo, E.-W. Bai, and F. Dabbene, "Probabilistic robustness analysis: Explicit bounds for the minimum number of samples," in *Proceedings of 35th IEEE Conference on Decision and Control*, vol. 3, pp. 3424–3428, IEEE, 1996.

[73] M. Ahsanullah, V. B. Nevzorov, and M. Shakil, *An introduction to order statistics*, vol. 8. Springer, 2013.

[74] T. Alamo, V. Mirasierra, F. Dabbene, and M. Lorenzen, "Safe approximations of chance constrained sets by probabilistic scaling," in *2019 18th European Control Conference (ECC)*, pp. 1380–1385, IEEE, 2019.

[75] M. Mammarella, V. Mirasierra, M. Lorenzen, T. Alamo, and F. Dabbene, "Chance constrained sets approximation: A probabilistic scaling approach–extended version," *arXiv preprint arXiv:2101.06052*, 2021.

[76] B. Karg, T. Alamo, and S. Lucia, "Probabilistic performance validation of deep learning-based robust nmpc controllers," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 18, pp. 8855–8876, 2021.

[77] M. Mammarella, T. Alamo, S. Lucia, and F. Dabbene, "A probabilistic validation approach for penalty function design in stochastic model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 11271–11276, 2020.

[78] T. Alamo, R. Tempo, A. Luque, and D. R. Ramirez, "Randomized methods for design of uncertain systems: Sample complexity and sequential algorithms," *Automatica*, vol. 52, pp. 160–172, 2015.

[79] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[80] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," 2008.

[81] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[82] B. Pelletier, "Kernel density estimation on Riemannian manifolds," *Statistics & probability letters*, vol. 73, no. 3, pp. 297–304, 2005.

[83] A. Papoulis and S. Unnikrishna Pillai, *Probability, random variables and stochastic processes*. 2002.

[84] M. Huber, "Halving the bounds for the Markov, Chebyshev, and Chernoff inequalities using smoothing," *The American Mathematical Monthly*, vol. 126, no. 10, pp. 915–927, 2019.

[85] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[86] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer, 2009.

[87] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[88] V. Vovk, "Kernel ridge regression," in *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, pp. 105–116, Springer, 2013.

[89] H. V. Henderson and S. R. Searle, "On deriving the inverse of a sum of matrices," *Siam Review*, vol. 23, no. 1, pp. 53–60, 1981.

[90] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[91] N. V. Sahinidis, "Optimization under uncertainty: state-of-the-art and opportunities," *Computers & chemical engineering*, vol. 28, no. 6-7, pp. 971–983, 2004.

[92] M. Mammarella, E. Capello, F. Dabbene, and G. Guglieri, "Sample-based SMPC for tracking control of fixed-wing UAV," *IEEE control systems letters*, vol. 2, no. 4, pp. 611–616, 2018.

[93] J. Li, W. Zhan, Y. Hu, and M. Tomizuka, "Generic tracking and probabilistic prediction framework and its application in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3634–3649, 2019.

[94] M. Chamanbaz, F. Dabbene, and C. Lagoa, "Algorithms for optimal ac power flow in the presence of renewable sources," *arXiv preprint arXiv:1811.05397*, 2018.

[95] M. Chamanbaz, F. Dabbene, and C. M. Lagoa, "Probabilistically robust ac optimal power flow," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 3, pp. 1135–1147, 2019.

[96] M. Lorenzen, F. Dabbene, R. Tempo, and F. Allgöwer, "Stochastic MPC with offline uncertainty sampling," *Automatica*, vol. 81, pp. 176–183, 2017.

[97] M. Mammarella, M. Lorenzen, E. Capello, H. Park, F. Dabbene, G. Guglieri, M. Romano, and F. Allgöwer, "An offline-sampling SMPC framework with application to autonomous space maneuvers," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 2, pp. 388–402, 2018.

[98] K. Margellos, P. Goulart, and J. Lygeros, "On the road between robust optimization and the scenario approach for chance constrained optimization problems," *IEEE Transactions on Automatic Control*, vol. 59, no. 8, pp. 2258–2263, 2014.

[99] M. Mammarella, V. Mirasierra, M. Lorenzen, T. Alamo, and F. Dabbene, "Chance-constrained sets approximation: A probabilistic scaling approach," *Automatica*, vol. 137, p. 110108, 2022.

[100] L. Hewing, A. Carron, K. P. Wabersich, and M. N. Zeilinger, "On a correspondence between probabilistic and robust invariant sets for linear systems," in *2018 European Control Conference (ECC)*, pp. 1642–1647, IEEE, 2018.

[101] A. Shapiro, D. Dentcheva, and A. Ruszczynski, *Lectures on stochastic programming: modeling and theory*. SIAM, 2021.

[102] W. Van Ackooij and J. Malick, "Eventual convexity of probability constraints with elliptical distributions," *Mathematical Programming*, vol. 175, no. 1-2, pp. 1–27, 2019.

[103] A. Prékopa, T. Rapcsák, and I. Zsuffa, "Serially linked reservoir system design using stochastic programing," *Water Resources Research*, vol. 14, no. 4, pp. 672–678, 1978.

[104] D. Dentcheva, B. Lai, and A. Ruszczyński, "Dual methods for probabilistic optimization problems," *Mathematical methods of operations research*, vol. 60, pp. 331–346, 2004.

[105] A. Vicino and G. Zappa, "Sequential approximation of feasible parameter sets for identification with set membership uncertainty," *IEEE Transactions on Automatic Control*, vol. 41, no. 6, pp. 774–785, 1996.

[106] J. M. Bravo, T. Alamo, and E. F. Camacho, "Bounded error identification of systems with time-varying parameters," *IEEE Transactions on Automatic Control*, vol. 51, no. 7, pp. 1144–1150, 2006.

[107] V. Puig, "Fault diagnosis and fault tolerant control using set-membership approaches: Application to real case studies," 2010.

[108] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 1999.

[109] C. A. Hoare, "Quicksort," *The computer journal*, vol. 5, no. 1, pp. 10–16, 1962.

[110] R. Cole, "Parallel merge sort," *SIAM Journal on Computing*, vol. 17, no. 4, pp. 770–785, 1988.

[111] C. Martınez, "Partial quicksort," in *Proc. 6th ACMSIAM Workshop on Algorithm Engineering and Experiments and 1st ACM-SIAM Workshop on Analytic Algorithmics and Combinatorics*, pp. 224–228, 2004.

[112] D. Dominique and F. Dominique, "An efficient partial sort algorithm for real time applications," in *2020 International Conference on Electrical Engineering (ICEE)*, pp. 1–5, IEEE, 2020.

[113] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[114] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari, "Multi-parametric toolbox 3.0," in *2013 European control conference (ECC)*, pp. 502–510, IEEE, 2013.

[115] F. Dabbene, C. Lagoa, and P. Shcherbakov, "On the complexity of randomized approximations of nonconvex sets," in *2010 IEEE international symposium on computer-aided control system design*, pp. 1564–1569, IEEE, 2010.

[116] V. T. H. Le, C. Stoica, T. Alamo, E. F. Camacho, and D. Dumur, *Zonotopes: From guaranteed state-estimation to control*. John Wiley & Sons, 2013.

[117] F. Dabbene, D. Henrion, C. Lagoa, and P. Shcherbakov, "Randomized approximations of the image set of nonlinear mappings with applications to filtering," *IFAC-PapersOnLine*, vol. 48, no. 14, pp. 37–42, 2015.

[118] T. Alamo, J. M. Bravo, and E. F. Camacho, "Guaranteed state estimation by zonotopes," *Automatica*, vol. 41, no. 6, pp. 1035–1043, 2005.

[119] E. Gover and N. Krikorian, "Determinants and the volumes of parallelotopes and zonotopes," *Linear Algebra and its Applications*, vol. 433, no. 1, pp. 28–40, 2010.

[120] M. D. Buhmann, "Radial basis functions," *Acta numerica*, vol. 9, pp. 1–38, 2000.

[121] J.-L. Goffin and J.-P. Vial, "On the computation of weighted analytic centers and dual ellipsoids with the projective algorithm," *Mathematical Programming*, vol. 60, no. 1, pp. 81–92, 1993.

[122] Y. Nesterov, "Cutting plane algorithms from analytic centers: efficiency estimates," *Mathematical Programming*, vol. 69, no. 1, pp. 149–176, 1995.

[123] A. Singh, "An overview of the optimization modelling applications," *Journal of Hydrology*, vol. 466, pp. 167–182, 2012.

[124] K. R. Muske, "Steady-state target optimization in linear model predictive control," in *Proceedings of the 1997 American control conference (Cat. No. 97CH36041)*, vol. 6, pp. 3597–3601, IEEE, 1997.

[125] V. Mirasierra, J. D. Vergara-Dietrich, and D. Limon, "Real-time optimization of periodic systems: A modifier-adaptation approach," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1690–1695, 2020.

[126] T. Rodríguez-Blanco, D. Sarabia, and C. de Prada, "Optimización en tiempo real utilizando la metodología de adaptación de modificadores," *Revista Iberoamericana de Automática e Informática industrial*, vol. 15, no. 2, pp. 133–144, 2018.

[127] J. D. Vergara-Dietrich, V. Mirasierra, A. Ferramosca, J. E. Normey-Rico, and D. Limón, "A modifier-adaptation approach to the one-layer economic MPC," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6957–6962, 2020.

[128] S. Costello, G. François, B. Srinivasan, and D. Bonvin, "Modifier adaptation for run-to-run optimization of transient processes," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 11471–11476, 2011.

[129] V. Mirasierra and D. Limon, "Modifier-adaptation for real-time optimal periodic operation," *arXiv preprint arXiv:2309.09680*, 2023.

[130] D. Limon, M. Pereira, D. M. De La Peña, T. Alamo, and J. M. Grosso, "Single-layer economic model predictive control for periodic operation," *Journal of Process Control*, vol. 24, no. 8, pp. 1207–1224, 2014.

[131] G. François and D. Bonvin, "Use of transient measurements for the optimization of steady-state performance via modifier adaptation," *Industrial & Engineering Chemistry Research*, vol. 53, no. 13, pp. 5148–5159, 2014.

[132] M. Vaccari, D. Bonvin, F. Pelagagge, and G. Pannocchia, "Offset-free economic MPC based on modifier adaptation: Investigation of several gradient-estimation techniques," *Processes*, vol. 9, no. 5, p. 901, 2021.

[133] K. R. Muske and T. A. Badgwell, "Disturbance modeling for offset-free linear model predictive control," *Journal of Process Control*, vol. 12, no. 5, pp. 617–632, 2002.

[134] G. Pannocchia and J. B. Rawlings, "Disturbance models for offset-free model-predictive control," *AIChE journal*, vol. 49, no. 2, pp. 426–437, 2003.

[135] D. Limón, I. Alvarado, T. Alamo, and E. F. Camacho, "Robust tube-based MPC for tracking of constrained linear systems with additive disturbances," *Journal of Process Control*, vol. 20, no. 3, pp. 248–260, 2010.

[136] D. Limon, I. Alvarado, T. Alamo, and E. Camacho, "On the design of robust tube-based MPC for tracking," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 15333–15338, 2008.

[137] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control*, pp. 207–226, Springer, 2007.

[138] K. H. Johansson, "The quadruple-tank process: A multivariable laboratory process with an adjustable zero," *IEEE Transactions on control systems technology*, vol. 8, no. 3, pp. 456–465, 2000.

[139] I. Alvarado, D. Limon, W. Garcia-Gabin, T. Alamo, and E. F. Camacho, "An educational plant based on the quadruple-tank process," *IFAC Proceedings Volumes*, vol. 39, no. 6, pp. 82–87, 2006.

[140] T. Raff, S. Huber, Z. K. Nagy, and F. Allgower, "Nonlinear model predictive control of a four tank system: An experimental stability study," in *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, pp. 237–242, IEEE, 2006.

[141] I. Alvarado, D. Limon, D. M. De La Peña, J. Maestre, M. Ridao, H. Scheu, W. Marquardt, R. Negenborn, B. De Schutter, F. Valencia, *et al.*, "A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark," *Journal of Process Control*, vol. 21, no. 5, pp. 800–815, 2011.

[142] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, In Press, 2018.

[143] A. Ferramosca, J. B. Rawlings, D. Limón, and E. F. Camacho, "Economic MPC for a changing economic criterion," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 6131–6136, IEEE, 2010.

[144] A. Ferramosca, D. Limon, and E. F. Camacho, "Economic MPC for a changing economic criterion for linear systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2657–2667, 2014.

[145] D. Limon, T. Alamo, M. Pereira, A. Ferramosca, A. H. González, and D. Odloak, "Integrating the RTO in the MPC: an adaptive gradient-based approach," in *2013 European Control Conference (ECC)*, pp. 7–12, IEEE, 2013.

[146] T. E. Marlin, A. N. Hrymak, *et al.*, "Real-time operations optimization of continuous processes," in *AIChE Symposium Series*, vol. 93, pp. 156–164, New York, NY: American Institute of Chemical Engineers, 1971-c2002., 1997.

[147] C. Y. Chen and B. Joseph, "On-line optimization using a two-phase approach: An application study," *Industrial & engineering chemistry research*, vol. 26, no. 9, pp. 1924–1930, 1987.

[148] S. Skogestad, "Self-optimizing control: The missing link between steady-state optimization and control," *Computers & Chemical Engineering*, vol. 24, no. 2-7, pp. 569–575, 2000.

[149] B. Srinivasan, C. Primus, D. Bonvin, and N. Ricker, "Run-to-run optimization via constraint control," *IFAC Proceedings Volumes*, vol. 33, no. 10, pp. 773–778, 2000.

[150] B. Chachuat, B. Srinivasan, and D. Bonvin, "Adaptation strategies for real-time optimization," *Computers & Chemical Engineering*, vol. 33, no. 10, pp. 1557–1567, 2009.

[151] M. R. Rajamani, J. B. Rawlings, and S. J. Qin, "Achieving state estimation equivalence for misassigned disturbances in offset-free model predictive control," *AIChE Journal*, vol. 55, no. 2, pp. 396–407, 2009.

[152] G. Pannocchia, M. Gabiccini, and A. Artoni, "Offset-free MPC explained: novelties, subtleties, and applications," *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 342–351, 2015.

[153] G. Pannocchia and J. B. Rawlings, "The velocity algorithm LQR: a survey," in *Technical Report 2001-01, TWMCC*, Department of Chemical Engineering, University of Wisconsin-Madison, 2001.

[154] L. Chisci, J. A. Rossiter, and G. Zappa, "Systems with persistent disturbances: predictive control with restricted constraints," *Automatica*, vol. 37, no. 7, pp. 1019–1028, 2001.

[155] A. Ferramosca, D. Limon, A. H. González, I. Alvarado, and E. F. Camacho, "Robust MPC for tracking zone regions based on nominal predictions," *Journal of Process Control*, vol. 22, no. 10, pp. 1966–1974, 2012.

[156] I. Alvarado, P. Krupa, D. Limon, and T. Alamo, "Tractable robust MPC design based on nominal predictions," *Journal of Process Control*, vol. 111, pp. 75–85, 2022.

[157] A. G. Marchetti, "A new dual modifier-adaptation approach for iterative process optimization with inaccurate models," *Computers & Chemical Engineering*, vol. 59, pp. 89–100, 2013.

[158] W. Gao and S. Engell, "Iterative set-point optimization of batch chromatography," *Computers & Chemical Engineering*, vol. 29, no. 6, pp. 1401–1409, 2005.

[159] S. Costello, G. François, and D. Bonvin, "A directional modifier-adaptation algorithm for real-time optimization," *Journal of Process Control*, vol. 39, pp. 64–76, 2016.

[160] T. de Avila Ferreira, G. François, A. G. Marchetti, and D. Bonvin, "Use of transient measurements for static real-time optimization," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5737–5742, 2017.

[161] M. Vaccari and G. Pannocchia, "A modifier-adaptation strategy towards offset-free economic MPC," *Processes*, vol. 5, no. 1, p. 2, 2016.

[162] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical programming*, vol. 27, no. 1, pp. 1–33, 1983.

[163] B. Gärtner and S. Schönherr, "An efficient, exact, and generic quadratic programming solver for geometric optimization," in *Proceedings of the sixteenth annual symposium on Computational geometry*, pp. 110–118, 2000.

[164] I. Alvarado, "Model predictive control for tracking constrained linear systems," *Doctor of Philosophy, Department of Systems Engineering and Automation, University of Sevilla Sevilla*, 2007.

[165] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. of the European Control Conference*, (Zürich, Switzerland), pp. 502–510, July 17–19 2013. http://control.ee.ethz.ch/ mpt.

[166] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward, "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers," *ACM Transactions on Mathematical Software (TOMS)*, vol. 31, no. 3, pp. 363–396, 2005.

[167] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the digital twin: A systematic literature review," *CIRP Journal of Manufacturing Science and Technology*, vol. 29, pp. 36–52, 2020.

[168] C. Herwig, R. Pörtner, and J. Möller, *Digital Twins*. Springer, 2021.

[169] F. Pires, A. Cachada, J. Barbosa, A. P. Moreira, and P. Leitão, "Digital twin in industry 4.0: Technologies, applications and challenges," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, vol. 1, pp. 721–726, IEEE, 2019.

[170] IBM, "What is a digital twin?." https://www.ibm.com/es-es/topics/what-is-a-digital-twin What is a digital twin?, Accessed: May 16, 2023.

[171] Deloitte, "Digital twin applications: Bridging the physical and digital." https://www2.deloitte.com/us/en/insights/focus/tech-trends/2020/digital-twin-applications-bridging-the-physical-and-digital.html Deloitte - Digital Twin Applications, 2020. Accessed: May 16, 2023.

[172] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," *Transdisciplinary perspectives on complex systems: New findings and approaches*, pp. 85–113, 2017.

[173] X. Fang, H. Wang, G. Liu, X. Tian, G. Ding, and H. Zhang, "Industry application of digital twin: From concept to implementation," *The International Journal of Advanced Manufacturing Technology*, vol. 121, no. 7-8, pp. 4289–4312, 2022.

[174] MarketsandMarkets, "Digital twin market revenue trends and growth drivers." https://www.marketsandmarkets.com/Market-Reports/digital-twin-market-225269522.html MarketsandMarkets - Digital Twin Market Revenue. Accessed: May 16, 2023.