Robust Multi-stage Nonlinear Model Predictive Control

Zur Erlangung des akademischen Grades eines

Dr.-Ing.

von der Fakultät Bio- und Chemieingenieurwesen
der Technischen Universität Dortmund
genehmigte Dissertation

vorgelegt von

Sergio Lucia, M.Sc.

aus

Zaragoza, Spanien

Tag der mündlichen Prüfung: 10.12.2014

1. Gutachter: Prof. Dr. Sebastian Engell

2. Gutachter: Prof. Dr. Lorenz T. Biegler

Dortmund 2015

Acknowledgements

First and foremost I would like to thank Prof. Sebastian Engell for giving me the opportunity of writing my PhD thesis under his supervision. It has been for me a very motivating and enriching experience and I am deeply grateful for his support and discussions during these four years, including the pressure that helped me to improve and try my best. I am also very grateful to Prof. Lorenz T. Biegler for agreeing to be the external referee of my thesis and for his very valuable feedback.

The work presented in this thesis would not have been possible without the help and the collaboration of my colleagues at the Group of Process Dynamics and Operations of the TU Dortmund. I would like to thank all the colleagues, secretaries and also the students that I supervised. Especially, I would like to thank Radoslav Paulen for all the work that we have done together, which has been very motivating for me, for proof-reading this thesis, and for teaching me my first words in Slovak while spending long hours of research together. I would like to thank also Daniel Haßkerl mainly for all the time that we have spent together during the organization of the International Summer Program of the TU Dortmund. I am also very grateful to all the colleagues which I have collaborated with. Many thanks especially to Sankaranarayanan Subramanian, Tiago Finkler, Ehsan Nabati, Christian Schoppmeyer, Lars Simora and Malte Behrens.

During my time in Dortmund I had the opportunity to participate in several projects and I would like to thank my colleagues from other universities for the productive and motivating collaborations. I am especially grateful to Joel Andersson for all the work together regarding CasADi and also to Daniel Limon for the joint work on the stability of the multi-stage approach.

I am deeply grateful to Alba for happily sharing with me these years in Germany, and for her continuous support, love and understanding. Finally, I would like to thank all my family, especially my parents and my brothers for being an endless source of support and love.

Abstract

Model Predictive Control (MPC) has become one of the most popular control techniques in the process industry mainly because of its ability to deal with multiple-input-multiple-output plants and with constraints. However, its performance can deteriorate in the presence of model uncertainties and disturbances. In the last years, the development of robust MPC techniques has been widely discussed, but these were rarely applied in practice due to their conservativeness or their computational complexity.

This thesis presents multi-stage nonlinear model predictive control (multi-stage NMPC) as a promising non-conservative robust NMPC control scheme, which is applicable in real-time. The approach is based on the representation of the evolution of the uncertainty by a scenario tree. It leads to non-conservative robust control of the plant because it takes into account explicitly that new information (usually present as measurements) will become available at future time steps and that the future control inputs can be adapted accordingly, acting as recourse variables.

Different aspects of the proposed multi-stage NMPC scheme are studied in detail in this thesis. Firstly, the approach is analyzed from a control theory point of view, including a formulation that guarantees stability and constraint satisfaction. Secondly, an efficient implementation is described, which is necessary to deal with one of the challenges of the presented method: The size of the resulting optimization problems. Thirdly, novel algorithms and modifications are proposed to enhance its performance and capabilities.

The method is evaluated using examples from the chemical engineering field. Several simulations and real experiments presented in this thesis show that multi-stage NMPC is a promising strategy for the optimizing control of uncertain nonlinear systems subject to hard constraints. It is also shown that multi-stage NMPC performs better than standard NMPC and better than other robust NMPC approaches presented in the literature while still being implementable in real-time.

Zusammenfassung

Modellprädiktive Regelung (kurz: MPC) ist eine der populärsten Methoden zur Regelung von Anlagen in der chemischen Industrie. Dies beruht auf der Möglichkeit, Mehrgrößensysteme mit Beschränkungen zu behandeln. Ungenauigkeiten im mathematischen Modell und äußere Einflüsse führen allerdings zu einer Verschlechterung der Regelgüte. Um die genannte Probleme zu vermeiden, wurden in den letzten Jahren verschiedene robuste MPC-Ansätze untersucht. Diese Ansätze finden aber in der Praxis selten Anwendung, weil sie sehr konservative Lösungen liefern oder ihre numerische Komplexität zu hoch ist.

In dieser Dissertation wird die Methode des mehrstufigen nichtlinearen MPC als ein vielversprechender robuster und nicht konservativer MPC-Ansatz vorgestellt. Die Methode basiert auf der Beschreibung der Unsicherheiten als Szenarienbaum. Dies führt zu einer nicht konservativen und robusten Regelung der Strecke, weil explizit berücksichtigt wird, dass in der Zukunft neue Informationen (oft in der Form von Messungen) zur Verfügung stehen. Die zukünftigen Stellgrößen können dann entsprechend angepasst werden und haben die Wirkung von *recourse* Variablen.

Verschiedene Aspekte des vorgestellten mehrstufigen nichtlinearen MPC-Ansatzes werden in dieser Dissertation ausführlich untersucht. Zunächst wird eine Analyse aus der Perspektive der Regelungstheorie vorgestellt, einschließlich einer Formulierung, welche die Stabilität und die Einhaltung der Nebenbedingungen garantieren kann. Im nächsten Schritt wird eine effiziente Implementierung beschrieben, die notwendig ist, um die wichtigste Herausforderung des Ansatzes zu bewältigen: die Größe des zu lösenden Optimierungsproblems. Im Anschluss daran werden neue Algorithmen und Ergänzungen vorgeschlagen, um das Potenzial des Ansatzes zu erhöhen.

Die vorgestellte Methode wird mit Hilfe von Beispielen aus der Verfahrenstechnik evaluiert. Mehrere Simulationen und Experimente werden in dieser Dissertation gezeigt. Die Ergebnisse deuten darauf hin, dass die Methode des mehrstufigen nichtlinearen MPC eine vielversprechende Strategie für die optimierungsbasierte Regelung unsicherer Systeme mit Beschränkungen ist. Im Vergleich zu Standard MPC oder anderen robusten MPC Methoden, weist der vorgestellte Ansatz eine bessere Regelgüte auf, während die gleichzeitig in Echtzeit realisierbar ist.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In times when energy consumption, emissions, and production volumes of industrial plants become ever more important, the optimization of the processes is a main goal of all engineers in charge. This objective has to be achieved taking into account constraints arising from quality, safety and legal requirements. To successfully realize this task in an uncertain environment and under the presence of disturbances is one of the main goals of advanced process control.

Beyond classical linear control theory, model-based control strategies have been established as a standard technique to cope with this problem, mainly because they can deal with coupled multivariable plants subject to constraints on the control inputs as well as on the process variables. However, the use of a model introduces uncertainty into the control problem in the form of plant-model mismatch, thus making robust control a fundamental tool.

One of the most popular model-based control strategies is model predictive control (MPC). MPC uses a mathematical model to predict the future behavior of a system and to compute a sequence of future control inputs by solving an optimization problem that minimizes a pre-defined objective. The accuracy of the model and the presence of disturbances have a strong influence on the performance and on the stability of such a controller. This is one of the main reasons that prevents MPC, especially its nonlinear version, from being widely used in the industry.

Dealing with uncertainty is one of the major challenges of model predictive control and therefore this subject has been widely studied during the last decades. While for the linear case useful results which can be also applied in practice have been developed, for the nonlinear case there is still a large gap between the problems that are solved in academia and the problems that arise from real applications. To bridge this gap is the central motivation of this thesis.

1.2 Scope of the Thesis

The main goal of this thesis is to investigate a new approach for the explicit consideration of uncertainty within the framework of model predictive control. This approach should extend the current state of the art by providing a method that achieves robust control of a system under the presence of uncertainties with a better performance than existing methods. The proposed approach should be applicable to complex problems and implementable in real-time. Furthermore, it should guarantee the stability and the constraint satisfaction of the controlled system as well as the recursive feasibility of the optimization problems.

These goals are accomplished in this thesis by means of **multi-stage nonlinear model predictive control**, which uses ideas of multi-stage stochastic programming to formulate the robust NMPC problem as a large optimization problem.

The performance of the proposed strategy will be evaluated with the help of challenging case studies from the chemical engineering field that show the advantages of the approach with respect to other methods presented in the literature.

1.3 Structure and Contribution of the Thesis

Fig. 1.1 shows the structure of this thesis. It consists of four main parts which deal with the different aspects necessary to develop the multi-stage NMPC approach. In order to achieve the main goals of this thesis, a multi-disciplinary approach is necessary which leads to different contributions of each part of the thesis. In the remainder of this section, the structure of the thesis is explained in detail together with the main contributions which have led to several publications.

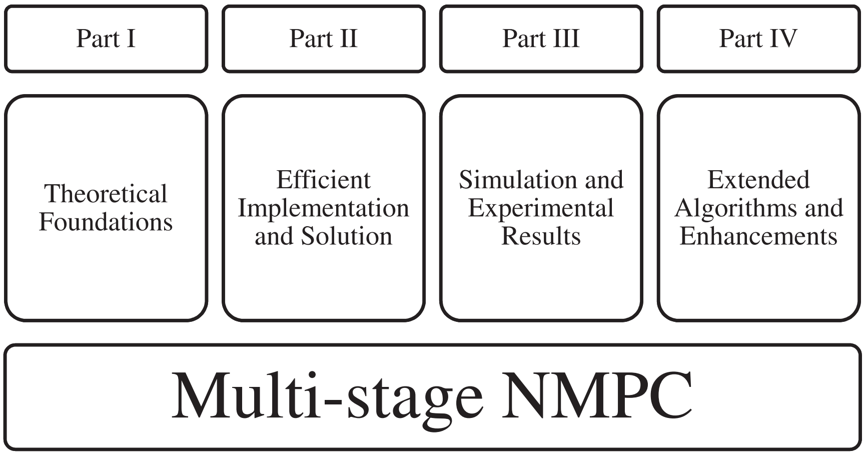
Part I
Part II
Part III
Part IV
Theoretical Foundations
Efficient Implementation and Solution
Simulation and Experimental Results
Extended Algorithms and Enhancements
Multi-stage NMPC

Figure 1.1: Structure of the thesis.

Part I: Theoretical Foundations

The first part presents the theoretical foundations of the approach on which the rest of the thesis is based. In particular, Chapter 2 presents a literature review of the existing methods for robust optimal control and for robust model predictive control.

Chapter 3 introduces the proposed multi-stage NMPC approach as a general framework for robust NMPC, which includes several other robust approaches. This chapter is mainly based on the publications (Lucia et al., 2013a) and (Lucia et al., 2014b).

Chapter 4 presents a formulation of the multi-stage approach that guarantees stability and recursive feasibility of the optimization problem by extending the proofs available for existing methods. The results presented in this chapter have been published in (Lucia et al., 2014c).

Part II: Efficient Implementation and Solution

The use of multi-stage NMPC results in the formulation of multi-stage stochastic optimization problems. These optimization problems are large and an efficient implementation is necessary to realize a real-time solution. This part of the thesis presents an efficient implementation and introduces a tool called DO-MPC for the simple, modular and efficient implementation of multi-stage NMPC.

Chapter 5 presents the different discretization methods used for the discretization of the nonlinear models that are used in this thesis, together with efficient methods for the computation of the derivatives and their use via the optimization framework CasADi.

Chapter 6 presents the DO-MPC tool that has been developed in cooperation with coworkers of the department for the efficient, simple and modular use of multi-stage NMPC. This efficient implementation has been published in (Lucia et al., 2014e).

Chapter 7 presents an approach to decompose the large-scale optimization problems that result from the multi-stage NMPC formulation into smaller subproblems which are computationally less expensive. This chapter is based on the results presented in (Lucia et al., 2013b).

Part III: Simulation and Experimental Results

The third part presents simulation and experimental results that illustrate the advantages of the proposed approach.

Chapter 8 includes results achieved for polymerization processes under strong uncertainties. In particular, results of the Chylla-Haase polymerization reactor, which are published in (Lucia et al., 2013a), and results of an industrial batch polymerization reactor provided by BASF SE (published in (Lucia et al., 2014b) and (Lucia et al., 2014a)) are presented to show the advantages of the multi-stage approach compared to existing methods. Other results that illustrate different aspects and applications of multi-stage NMPC that are not included in this thesis can be found in (Lucia and Engell, 2012), (Lucia and Engell, 2013) or in (Lucia and Engell, 2014).

Chapter 9 presents the results for a stability guaranteeing formulation of multi-stage NMPC applied to a simple example, as published in (Lucia et al., 2014c).

The efficient implementation of this approach enables its implementation on real plants, which is demonstrated in Chapter 10 by its application to a laboratory process.

Part IV: Extended Algorithms and Enhancements

The last part of the thesis describes different algorithms that are based on the aforementioned theory and methods, and can be used to enhance the performance of the proposed approach in different manners.

Chapter 11 describes an extension of the multi-stage approach to reduce the variability of the system under uncertainty. It is shown that enforcing a low variability of a system under uncertainty may lead to a very poor performance, especially in the case of NMPC with an economic cost function. These results have been published in (Lucia et al., 2014b).

Chapter 12 presents a modification of the multi-stage NMPC that makes it possible to guarantee robust constraint satisfaction for the uncertainty values that are not included in the scenario tree on which the proposed approach is based. This chapter presents the results published in (Lucia et al., 2014d).

Chapter 13 integrates the multi-stage approach with the use of optimal experiment design which reduces the uncertainty range leading to an improved performance. These results appeared in (Lucia and Paulen, 2014).

Part V: Summary, Conclusions and Future Work

The thesis is finalized in Chapter 14 where the main conclusions and directions for future work are presented. Chapter 14 also includes some guidelines for the use of multi-stage NMPC based on the results developed throughout this thesis and Appendix A presents results of multi-stage NMPC for the problem of setpoint tracking.

The work presented in this thesis is the result of the research performed at TU Dortmund with the financial support of the EU in the framework of the FP7 Project EMBOCON (248940) and the ERC Advanced Grant MOBOCON (291458), as well as with the support of the Deutsche Forschungsgemeinschaft (German Research Council) in the framework of the research cluster optimization-based control of uncertain systems (EN 152/39-1). This support is gratefully acknowledged and appreciated.

Part I

Theoretical Foundations

Chapter 2

Optimization-based Robust Control

2.1 Optimal Control

Optimal control theory was born in the form of calculus of variations in 1697 when Johann Bernoulli published his solution and the solution that different contemporaries (Newton, Leibniz, l'Hopital, Tschirnbaus and his brother Jakob Bernoulli) had proposed to solve the *brachystochrone problem*. After important contributions from Euler and Lagrange among others (see (Sargent, 2000) for a more detailed historical perspective), two of the major contributions to modern optimal control theory as it is known today were provided by Pontryagin in (Pontryagin et al., 1962) where Pontryagin's minimum principle was presented and by the work of Bellman (Bellman, 1957) which resulted in the Hamilton-Jacobi-Bellman equation. The contributions from Pontryagin and Bellman made it possible to solve optimal control problems analytically.

After that, many works were focused on the linear, unconstrained optimal control problems, in particular on the linear quadratic (LQ) optimal control problem and on the linear quadratic estimation (LQE) problem. In the LQ problem one tries to find the optimal inputs that minimize a quadratic cost function for an infinite horizon and an analytic solution can be obtained by solving the Ricatti differential equation. The LQE problem consists in finding the optimal estimates using a model affected by process noise and noisy measurements. Kalman proposed in (Kalman, 1960) a recursive formulation which gives the optimal solution to this problem and it is known as the Kalman Filter. The solution to these two important problems was one of the major achievements in control theory during the 20th century. The combination of both problems (the linear quadratic regulator and the Kalman Filter) is known as the linear quadratic Gaussian (LQG) problem. Doyle showed in (Doyle, 1978), that the LQG solution could exhibit robustness problems even for arbitrarily small perturbations. This led to the development of robust linear control theory. Most of the approaches for robust linear control theory are focused on $\mathcal{H}_\infty$ control (Zames, 1981), on Kharitonov's methods (Kharitonov, 1978), and on set-theoretic methods e.g. with bounding ellipsoids as presented in (Glover and Schweppe, 1971).

Finding an analytic solution for the nonlinear case of the optimal control problem (and of the robust optimal control problem) is in general not possible. It is then necessary to employ numerical methods to solve general optimal control problems. The most used numerical methods are usually classified as *indirect* methods and *direct* methods. Indirect methods formulate the optimality conditions (obtained e.g. applying Pontryagin's minimum principle) and then discretize them to obtain a solution numerically. For this reason, indirect methods are also called *first optimize then discretize* methods, see e.g. (Bryson and Ho, 1975). Direct methods for the solution of optimal control problems were proposed by (Bosarge and Johnson, 1970) among others leading to the control vector parametrization approach in (Sargent and Sullivan, 1978), the collocation approach in (Tsang et al., 1975), and (Cuthrell and Biegler, 1987) and the multiple shooting method in (Bock and Plitt, 1984). These methods are also referred to as *first discretize then optimize* methods because the dynamics of the system and the control inputs are first discretized to formulate a nonlinear programming problem, which is then solved numerically. These methods form the basis of most applications of dynamic optimization.

Another major advantage of transforming the optimal control problem into an optimization problem is that different techniques from the field of optimization under uncertainty can be used to address the issue of robustness of the optimal control solutions. The field of optimization under uncertainty has experienced a tremendous progress in the last century and some of the most relevant approaches are reviewed in the next section.

2.2 Optimization Under Uncertainty

Over the past 70 years, many techniques have been proposed to tackle the problem of optimization under uncertainty including stochastic programming, dynamic programming, and robust (worst-case, min-max) optimization.

One of the first efforts in the field of optimization under uncertainty was the work of Dantzig (Dantzig, 1955) and Beale (Beale, 1955) for linear stochastic programming. A huge amount of works followed this research making stochastic programming a very active field with applications in many different disciplines, including operations research, finances, and engineering.

Most of the formulations in stochastic programming consider a two- or multi-stage problem based on different scenarios of the uncertainty (Birge, 1997). The decision maker can adapt the future decisions to the future observations, which is usually illustrated in the separation of the decisions between *here-and-now* decisions that have to be fixed at a certain time instant and the *wait-and-see* decisions which can be adapted according to future observations. This concept – called *recourse* – plays a major role in optimization under uncertainty and it is also a central element of this thesis. The typical formulation of a two-stage linear stochastic programming problem can be written as:

$$\min_{x,y_1,\ldots,y_N} \quad c^T x + \sum_{i=1}^{N} \omega_i \, q_i^T \, y_i \tag{2.1a}$$

subject to:

$$Ax \leq b \tag{2.1b}$$

$$W_i y_i \leq h_i - T_i x, \quad i = 1, ..., N, \tag{2.1c}$$

where the vector x denotes the *here-and-now* (or first-stage) decisions, and the vectors y_i are the *wait-and-see* (or second-stage) decisions for each one of the N different possible scenarios, which have a probability ω_i to occur. The cost is a linear function of the first-stage and the second-stage variables with coefficients c and q_i. The constraints (2.1b) (with A and b being a matrix and a vector of appropriate dimensions) represent linear constraints on the first-stage variables and (2.1c) (with W_i, T_i and h_i matrices and vector of appropriate dimensions) denotes the constraints on the second-stage variables. The key of this formulation is that the problem (2.1) models the fact that at the current time a decision has to be made (*here-and-now* decision) in the presence of the uncertainty, but this decision is taken assuming that at the next sampling time the decision maker will take the optimal decision for each realization of the uncertainty. Thanks to this recourse, the *here-and-now* decision can be less conservative because it relies on the possibility of adapting the future decisions to the observation of the uncertainty. Two-stage problems are based on the optimistic assumption that the uncertainty will be completely revealed at the next time stage, and therefore an optimal decision for each scenario can be taken. If this assumption is not valid for the problem under study, the formulation has to be extended to a multi-stage problem. In this case decisions are taken at each stage as a response to the new information that becomes available as time progresses. A good overview of the methods and theoretical basis of stochastic programming can be found in (Birge, 1997) and in (Shapiro, 2009).

A different possibility to tackle the problem of uncertainty in optimization is to use dynamic programming. This approach tries to find the optimal policies π_k that minimize the expected cost over a predefined planning horizon for all the possible values of the uncertainty. Its formulation is based on the definition of the state s_k of a dynamic system at time k, to which decisions x_k are applied. Then it searches for the policies $\pi_k(s_k) = x_k$ that will result in a minimization of the expectation of the cost $C_k(x_k)$ over a predefined planning horizon. Because of the presence of uncertainty, the state at the next stage is a function of the previous state, the control action, and the uncertainty ($s_{k+1} = f(s_k, x_k, d_k)$). A general formulation of the dynamic programming formulation can be written as the following optimization problem:

$$\min_{\pi_k \in \Pi} \quad \mathbb{E}\left\{\sum_{k=0}^{K} C_k(x_k))\right\}, \tag{2.2}$$

where K is the planning horizon and Π is the set of allowed policies. The work from Bellman (Bellman, 1957) shows that an optimal policy that satisfies (2.2) also satisfies the so-called Bellman equation, which defines the cost-to-go or value function at state s_k, denoted as $V_k(s_k)$, as the optimal cost from stage k until the end of the planning horizon and can be written as:

$$V_k(s_k) = \min_{\pi_k \in \Pi} \quad C_k(x_k) + \mathbb{E}\left\{V_{k+1}(f(s_k, x_k, d_k)|s_k\right\}. \tag{2.3}$$

The Bellman equation is based on the so-called Bellman's principle of optimality (Bellman, 1957), which says that for any optimal policy, regardless of the previous decisions, the remaining decisions must be an optimal policy with respect to the state that resulted from the previous decisions, i.e., every terminal part of an optimal policy is optimal. However, solving (2.3) involves in general solving a recursive optimization problem which is very difficult especially for high dimensions of the state and some approximations are necessary (Powell, 2007). In particular the use of approximate dynamic programming (ADP) has reported interesting results (see (Lee, 2014) for an overview). Dynamic programming shares many common elements with stochastic programming, including the concept of recourse, but as recently discussed in (Powell, 2012), due to the use of a different formalism and different tools some of the parallelisms cannot be recognized easily.

A different approach to optimization under uncertainty is to use the *robust counterpart* of the optimization problem. This approach started with the work of Soyster (Soyster, 1973) on robust linear programming. An important number of works have been devoted to this topic in the last decades under the name of Robust Optimization. Most of the approaches are focused on a convex formulation because in some cases it is possible to reformulate the problem to a tractable convex optimization problem as shown in (Ben-Tal and Nemirovski, 1998), which can be easily solved.

The simplest formulation addresses (in the linear case) problems of the following type:

$$\min_{x} \quad \tilde{c}^T x, \tag{2.4a}$$

subject to:

$$\tilde{A}x \leq \tilde{b}, \tag{2.4b}$$

where x is the vector of decision variables, $\tilde{c}$ is the cost vector and $\tilde{A}$ and $\tilde{b}$ denote the constraints on the decision variables. In this framework, it is considered that $\tilde{c}$, $\tilde{A}$ and $\tilde{b}$ are all uncertain but known to be contained in a given uncertainty set. The optimal solution of such a problem in the framework of robust optimization is the solution that minimizes the value of the cost function for the worst case realization of the uncertainty while satisfying the constraints for all the possible values of the uncertainty. For this reason, these methods are also referred as min-max formulations. All the decision variables are *here-and-now* decisions, i.e., there is no possibility of recourse, and the fact that future decisions can be adapted to the future observations is ignored. This can result in conservative solutions compared to approaches that take recourse into account. In order to overcome this drawback, some extensions of robust optimization have been recently proposed such as the so-called adjustable robust counterpart (Ben-Tal et al., 2004) which includes the notion of recourse in the robust optimization framework, borrowing ideas from dynamic and stochastic programming. A wider overview of these methods is presented in the recent book (Ben-Tal et al., 2009).

2.3 Robust Model Predictive Control

Model Predictive Control, which was initially proposed in (Richalet et al., 1978) and (Cutler and Ramaker, 1980), calculates at each sampling time of the controller a sequence of optimal control inputs by solving an optimization problem that minimizes a pre-defined cost function over a finite prediction horizon subject to input and state constraints. This is illustrated in Fig. 2.1. Here the red line represents the control input trajectory $u(t)$, which is calculated until the prediction horizon N_P. In a typical setting the objective is to minimize the distance between a pre-defined reference ($w(t)$) and the predicted trajectory of the plant ($y(t)$). As shown in Fig. 2.1, the control inputs are usually considered as piece-wise constant. In order to reduce the computational complexity of the resulting optimization problems it is common to reduce its degrees of freedom by considering the control input trajectory to be constant after the so-called control horizon (N_u). Only the first control input of this sequence is applied to the plant, and at the next sampling time the procedure is repeated using the new measurements of the states as the initial condition for the next optimization problem. Usually steady-state accuracy is achieved in the presence of modeling errors and disturbances via a bias correction term that is computed from the last output measurements. In this case, the difference between the last measurement and the prediction obtained at the previous step is added to the predicted output along the prediction horizon, i.e., it is assumed that this difference will remain constant in the future. As a result, feedback information – which is necessary to control an uncertain system – enters the control loop only by the re-initialization of the optimization problem and by the bias term. For this reason, the performance and the stability of NMPC controllers rely strongly on the accuracy of the model that is used for the optimization.

Robust model predictive control methods aim at overcoming the limitations of conventional MPC with respect to the influence of model errors and disturbances, and have attracted the attention of many researchers during the last years. Although it is known that the standard MPC approach – ignoring uncertainties – has an inherent robustness under quite strong assumptions (see (Grimm et al., 2004), (Limon et al., 2009)), this is usually not sufficient for general nonlinear constrained systems and therefore the ideas of optimization under uncertainty presented in the previous section were also used for the robustification of MPC controllers. The first efforts in robust MPC were focused on the so-called min-max MPC (Campo and Morari, 1987) using ideas of robust optimization

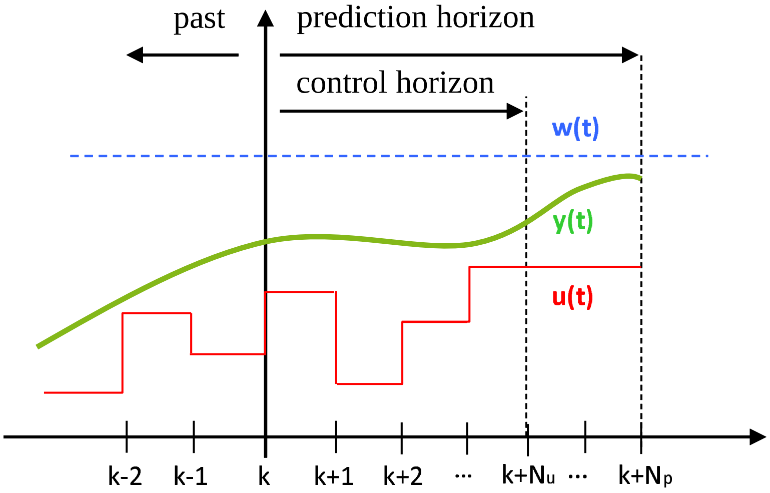

Figure 2.1: Model Predictive Control strategy.

which were already presented in (Witsenhausen, 1968). This approach obtains a sequence of control inputs that minimizes the cost of the worst-case realization of the uncertainty while satisfying the constraints also for all the cases of the uncertainty. Min-max MPC however does not take into account the fact that new information will be available in the future and therefore the result may be overly conservative and may lead to infeasible optimization problems, as illustrated in (Scokaert and Mayne, 1998).

In order to overcome this problem, closed-loop (or feedback) min-max NMPC was proposed in (Lee and Yu, 1997), and (Mayne, 2001) using ideas from robust dynamic programming. In this method, the cost function is minimized over a sequence of control policies (in contrast to a sequence of control inputs). This solves the problems of conservativeness and feasibility because feedback is taken into account also within the optimization problem and not only by its re-initialization. However, the resulting problem is of infinite dimension and therefore difficult to solve. A solution presented in the literature is to restrict the possible policies to apply, for instance, only affine feedback control policies as in (Bemporad, 1998), (Löfberg, 2003) or in (Goulart et al., 2006). This leads to a certain degree of suboptimality that can be estimated for time-varying linear discrete systems, as shown in (Hadjiyiannis et al., 2011). Other works in this direction tried to apply approximate dynamic programming (Bjornberg and Diehl, 2006) in the context of MPC. The complexity of this formulation limits its application to linear systems or more recently to nonlinear systems without state constraints (Summers et al., 2013).

Interestingly, less attention has been paid to the integration of the stochastic programming paradigm in the framework of model predictive control. Although some works have been presented in the linear case ((Scokaert and Mayne, 1998), (Muñoz de la Peña et al., 2005b), (Bernardini and Bemporad, 2009)), the use of stochastic programming in robust MPC has not been extensively studied in the case of general nonlinear systems. This is the main topic of this work. In this thesis, multi-stage NMPC is presented as a general framework that includes standard and min-max NMPC. Under the assumption that the uncertainty can be perfectly modeled by a scenario tree, multi-stage NMPC provides the best possible solution for the robust NMPC problem since it computes the optimal closed-loop feedback policy over a finite prediction horizon. If this assumption does not hold, i.e. if the uncertainty does not take only discrete values, multi-stage NMPC computes an approximation of the optimal feedback policy that can be close to the optimal one if a suitable scenario tree is chosen as will be discussed in Chapter 8.

The main drawback of the approach is that the size of the resulting optimization problem grows exponentially with the length of the prediction horizon, and with the number of uncertainties as well as with the number of different values of each uncertainty that is considered in the design of the scenario tree. A detailed description of this approach is provided in the next chapter.

A different approach for robust MPC which is not based on optimization under uncertainty techniques but on set-theoretic methods is tube-based MPC. This approach was presented for linear systems in (Mayne et al., 2005) and extended to the nonlinear case in (Mayne and Kerrigan, 2007). It recently received attention as an alternative to min-max approaches for the formulation of a robust NMPC scheme with guaranteed stability and recursive feasibility. Tube-based MPC is based on the solution of the nominal control problem and the inclusion of a so-called ancillary controller that ensures that the evolution of the real uncertain system stays within a tube around the nominal trajectory. The cross-section of this tube is a robust positive invariant set, centered around the nominal trajectory.

Different modifications and improvements have been reported in the literature (see (Rawlings and Mayne, 2009), (Mayne et al., 2011), (Rakovic et al., 2011), (Yu et al., 2011)). They mainly differ in the way how the cross-sections and the ancillary controller are calculated, leading to different computation complexities and degrees of conservativeness. In order to ensure that the constraints are robustly satisfied, the admissible set for the nominal problem has to be tightened in all cases, which can be difficult and conservative for nonlinear systems if the constraints are active, as indicated in (Rawlings and Mayne, 2009). Tube-based control can be seen as the formalization of the standard engineering approach to separate between trajectory or set-point generation and local feedback control to reduce the influence of disturbances. It can guarantee stability and constraint satisfaction in the presence of bounded but unknown influences but does not address the issue of optimal performance in the presence of uncertainties.

Other methods try to remove (completely or at least partially) the uncertainty from the problem by estimating it using methods of experimental design (Qian et al., 2013) or dual control (Adetola et al., 2009). The robustness with respect to the remaining uncertainty can be achieved by means of min-max methods, or by computing a backoff with respect to the active constraints as done e.g. in (Arellano-Garcia et al., 2005).

All the approaches discussed in this chapter try to guarantee stability or robust constraint satisfaction for a given uncertainty range. There are other recent methods which enforce the satisfaction of the constraints with a given probability (also called chance constraints), given a probability distribution of the uncertainty, or a certain number of samples. These methods are usually classified as stochastic model predictive control approaches. One possibility is to use the scenario approach presented in (Calafiore and Campi, 2006) which can be used for convex problems or special cases of nonlinear problems as shown in (Grammatico et al., 2014). Other approaches use a problem formulation based on polynomial chaos expansions e.g. in (Fagiano and Khammash, 2012), (Mesbah et al., 2014) for parametric time-invariant uncertainty, or the works in (Farina et al., 2013), (Cannon et al., 2011), which assume additive time-varying uncertainties.

2.4 Real-Time Optimization

Traditionally in the control of chemical processes under uncertainty, linear or nonlinear MPC controllers are used to track set-points calculated by an upper layer which uses steady-state optimization. Different methods have been presented in the literature to solve the optimization problems in the upper layer explicitly, considering uncertainties. These methods are usually classified under the name of Real-Time Optimization (RTO).

The main idea of RTO is to use measurements of the real system in order to achieve –

usually in an iterative manner – optimal performance and constraint satisfaction in the presence of uncertainties. As classified in (Chachuat et al., 2009), the RTO schemes can be divided into different categories depending on how the measurement information is used. If this information is used to update the parameters of the model, the two-step approach presented in (Chen and Joseph, 1987), or in (Marlin and Hrymak, 1997) is obtained. The information of the measurements can be used for the direct adaptation of the inputs using different methods as for example the self optimizing control presented in (Skogestad, 2000) or the tracking of the necessary conditions of optimality (NCO tracking, (Srinivasan et al., 2002)). The main idea of the direct adaptation of the inputs is to avoid the use of numerical optimization by finding some measured variables that, if maintained constant (using e.g. a simple controller), enforce optimal performance. The measurement information can also be used to modify directly the optimization problem by updating the cost function and/or the constraints. This category of RTO schemes is usually referred to as modifier adaptation. The basic idea of these methods is to use measurements of the real plant to estimate an experimental gradient of the plant, which is added to the cost function achieving robust optimality as presented in (Roberts, 1979). An update of the constraints can also be performed as shown in (Gao and Engell, 2005) so that the gradients of the constraints also match the plant gradients and upon convergence the Karush-Kuhn-Tucker conditions of optimality are satisfied for the real plant (Marchetti et al., 2009).

In a classical setup of a process control hierarchy, the RTO techniques are used to compute set-points that have to be tracked using a lower-level MPC controller. Most of the literature in control theory and applications until the recent years has focused on achieving optimal and of course stable reference tracking. However, the actual goal of process control is not to track as well as possible a certain set-point that has been previously generated, but to maximize the profit (or minimize the costs) of a process as discussed in (Engell, 2007). For this reason, in the last years the traditional two-layer framework with a tracking cost function of the MPC scheme has frequently been replaced (at least in the academic community) by a one-layer scheme in which an NMPC controller is used with a general cost function that can represent the plant economics. Different results have been reported recently usually referring to this approach as economic MPC in e.g. (Rawlings and Amrit, 2009), (Idris and Engell, 2012), (Ellis et al., 2013), and in (Gopalakrishnan and Biegler, 2013). An important implication of economic MPC is that the economic operation of a system usually drives the system to its constraints and for this reason the use of robust schemes for economic MPC is crucial to guarantee that the constraints will be respected also in the presence of model uncertainty and disturbances.

2.5 Discussion

This thesis uses ideas from the four research fields presented in this chapter: Optimal control, optimization under uncertainty, robust model predictive control and real time optimization.

Optimal control ideas are used in this thesis to transform the original optimal control problem into a nonlinear programming problem using direct methods. This transformation makes it possible to use optimization under uncertainty, in particular multi-stage stochastic programming, in the framework of robust NMPC to achieve a robust non-conservative control of systems under uncertainty in which the control task is not to track a predefined set-point, but to maximize the profit of a process using the ideas of real-time optimization that led to economic NMPC formulations.

This chapter presented a short overview of some of the methods that are widely used for the optimization-based control of uncertain systems. Due to the great interest that this subject has attracted in the last decades, it is not possible to cover all the different methods. For a more detailed description of classical robust optimal control methods the reader is referred to (Zhou et al., 1996). The review in (Sahinidis, 2004) presents different alternatives for optimization under uncertainty and the recent book (Ben-Tal et al., 2009) presents robust optimization in detail. The main ideas of approximate dynamic programming are discussed in (Powell, 2007) and some approaches for robust MPC, especially tube-based methods, are discussed in more detail in (Rawlings and Mayne, 2009). More information about RTO schemes can be found in (Srinivasan et al., 2002), (Engell, 2009) and (Chachuat et al., 2009).

Chapter 3

Multi-stage NMPC as a General Framework for Robust NMPC

This chapter presents the framework of multi-stage nonlinear model predictive control, which is the central idea of the thesis. It is shown that this framework includes other approaches such as a standard NMPC (which ignores the uncertainties), an open-loop approach where no recourse is introduced and other robust approaches that parametrize the future control inputs as affine policies. The resulting optimization problems and the notation which will be used for the rest of the thesis are also defined here.

3.1 Multi-stage Nonlinear Model Predictive Control

Multi-stage NMPC is a robust NMPC approach that is based on describing the evolution of the uncertainty by a scenario tree (see Fig. 3.1). Each branching at a node represents the effect of an unknown uncertain influence (disturbance and model error) together with the chosen control input. The tree structure represents how future control inputs can depend on the previous values of the uncertainty if full state measurement or perfect estimation is assumed. This means that future control inputs can act as recourse variables that counteract the effect of the future uncertainties, i.e., when the scenario tree is formulated at a given time it is considered explicitly that when a new decision has to be taken in the future, new information (usually in the form of measurements) will be available and hence the decision can be adjusted accordingly. Thus, multi-stage NMPC is a closed-

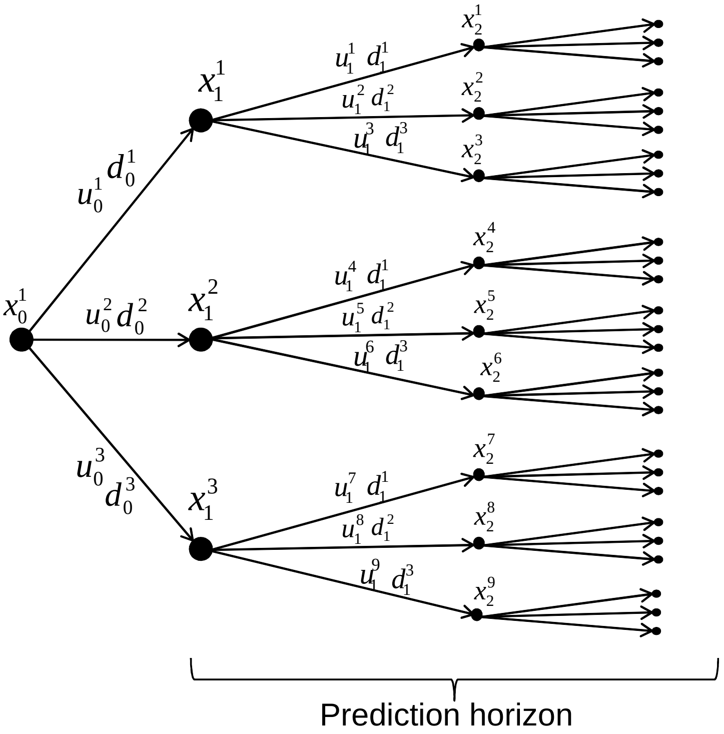

Figure 3.1: Scenario tree representation of the uncertainty evolution for multi-stage NMPC.

loop robust NMPC approach that exhibits a lower degree of conservativeness compared to other approaches, such as open-loop min-max NMPC, multi-model or multi-scenario approaches without recourse (as the one in (Huang et al., 2009a)). The basic idea of the use of a scenario tree for MPC was suggested in (Scokaert and Mayne, 1998) and some results for linear MPC have been reported in (Muñoz de la Peña et al., 2005b) and in (Bernardini and Bemporad, 2009). The same approach has been applied in planning and scheduling, see e.g. (Sand and Engell, 2004) and (Cui and Engell, 2010). The first steps in the direction of nonlinear systems were given in (Dadhe and Engell, 2008).

It is important to note that the tree structure does not necessarily represent time-varying uncertainties or disturbances, but it reflects the fact that if the uncertainty is not known at one sampling time, it will remain unknown at the next sampling time when a new tree (shifted in time) will have to be considered.

Since the constraints are directly enforced in the optimization problem, their satisfaction is guaranteed for all the values of the uncertainty that are considered in the scenario tree.

For general nonlinear systems, robust constraint satisfaction cannot be guaranteed for values that are not represented explicitly in the tree. However, the values of the parameters that produce the worst-case scenario are often on the boundaries of the considered parameter interval (Srinivasan et al., 2002) and therefore a suitable scenario tree should include the combinations of the extreme values of the parameters. In the remainder of the thesis, an NMPC scheme is considered to be robust if it satisfies the constraints for a given (properly chosen) scenario tree. If a suitable scenario is generated by the combination of the different possible extreme values of the uncertainties, the size of the tree grows exponentially with the number of uncertainties, as well as with the length of the prediction horizon. This constitutes the main drawback of the approach.

In order to represent the real-time decision problem correctly, the control inputs cannot anticipate the values of the uncertainty that are realized after the corresponding decision point. This is enforced by the non-anticipativity constraints that require all the control inputs that branch at the same node to be equal (for example in Fig. 3.1, $u_0^1 = u_0^2 = u_0^3$; $u_1^1 = u_1^2 = u_1^3$; ...).

The scenario tree setting assumes a discrete-time formulation of an uncertain nonlinear dynamic system described by $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \to \mathbb{R}^{n_x}$ that can be written as:

$$x_{k+1}^j = f(x_k^{p(j)}, u_k^j, d_k^{r(j)}), \tag{3.1}$$

where each state $x_{k+1}^j \in \mathbb{R}^{n_x}$ is a function of the previous state $x_k^{p(j)}$ and n_x is the number of states. The superscript $^{p(j)}$ denotes the index of the previous (also called parent) node in the tree, which is a function of its position j and of the stage k. The explicit dependence on k is dropped from the notation for simplicity. The control input vector is denoted as $u_k^j \in \mathbb{R}^{n_u}$ and n_u is the number of control inputs. The realization r of the uncertainty at stage k, which for each stage is a function of the position j in the scenario tree, is denoted as $d_k^{r(j)} \in \mathbb{R}^{n_d}$, where n_d is the dimension of the uncertainty vector (for example in Fig. 3.1, $x_2^7 = f(x_1^3, u_1^7, d_1^1)$). For simplicity of the presentation, it is considered that the tree has the same number of branches at all nodes, given by $d_k^{r(j)} \in \{d_k^1, d_k^2, ..., d_k^s\}$ at stage k for s different possible values of the uncertainty. In order to make the notation clear, the index set of all occurring indices (j, k) is denoted by I. Each path from the root node x_0 to a leaf node $x_{N_P}^i$ is called a scenario and is denoted by S_i, which contains all the states x_k^j and control inputs u_k^j that belong to scenario i:

$$S_i = \{x_{N_P}^i, x_{N_P-1}^{p(i)}, x_{N_P-2}^{p(p(i))}, ..., x_0^1, u_{N_P-1}^i, u_{N_P-2}^{p(i)}, u_{N_P-2}^{p(p(i))}, ..., u_0^i\}, \;\; \forall\, i = 1, ..., N, \tag{3.2}$$

where N is the number of scenarios (or leaf nodes) and N_P is the prediction horizon. In a similar way the set of states that belong to scenario i is denoted as:

$$X_i = \{x^i_{N_P}, x^{p(i)}_{N_P-1}, x^{p(p(i))}_{N_P-2}, ..., x^1_0\}, \;\; \forall\, i = 1, ..., N, \tag{3.3}$$

and the set of control inputs that belong to scenario i can be written as:

$$U_i = \{u^i_{N_P-1}, u^{p(i)}_{N_P-2}, u^{p(p(i))}_{N_P-2}, ..., u^i_0\}, \;\; \forall\, i = 1, ..., N, \tag{3.4}$$

The optimization problem resulting from the multi-stage formulation in a scenario-based setting can be written as:

$$\min_{x^j_k, u^j_k \forall (j,k) \in I} \quad \sum_{i=1}^{N} \omega_i J_i(X_i, U_i), \tag{3.5a}$$

subject to:

$$x^j_{k+1} = f(x^{p(j)}_k, u^j_k, d^{r(j)}_k), \qquad \forall\, (j, k+1) \in I, \tag{3.5b}$$

$$g(x^j_{k+1}, u^j_k) \leq 0\ , \qquad \forall\ (j, k+1) \in I, \tag{3.5c}$$

$$u^j_k = u^l_k \text{ if } x^{p(j)}_k = x^{p(l)}_k \qquad \forall\ (j,k), (l,k) \in I, \tag{3.5d}$$

where $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_g}$ represents general and possibly nonlinear constraints on the states and the inputs of the control problem evaluated at each node of the tree. The number of constraints is determined by n_g. The cost of each scenario S_i with weight ω_i is denoted by $J_i : \mathbb{R}^{n_x \times N_P+1} \times \mathbb{R}^{n_u \times N_P} \rightarrow \mathbb{R}$ defined as:

$$J_i(X_i, U_i) = \sum_{k=0}^{N_P-1} L(x^j_{k+1}, u^j_k), \;\; \forall\, x^j_{k+1} \in X_i,\, u^j_k \in U_i, \tag{3.6}$$

$L : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}$ is the stage cost, which represents a general cost function. The non-anticipativity constraints in (3.5d) enforce that the decisions u^j_k with the same parent node $x^{p(j)}_k$ must be the same. If the summation in (3.5a) is replaced by the max operator, a closed-loop min-max approach is obtained, in which feedback is taken explicitly into account. A comparison of both approaches will be presented in Chapter 8. If the number of scenarios is $N = 1$, the problem is reduced to standard NMPC. The weights $\omega_i \in \mathbb{R}^N_+$ can be adapted according to parameter estimation or stochastic information if it is available or chosen to be identical if no information is available.

3.1.1 The Robust Horizon Assumption

From a computational point of view, the main challenge of the multi-stage approach is the rapid growth of the size of the scenario tree for increasing length of the prediction horizon, increasing number of uncertainties and increasing number of branches considered for each uncertainty. This problem is commonly known in the literature of stochastic programming and dynamic programming as the curse of dimensionality. This can be seen in Table 3.1 in which the number of nodes of a scenario tree resulting from a multi-stage NMPC formulation is shown for different lengths of the prediction horizon N_P and number of uncertainties. It is assumed that the scenario tree is generated by combining the maximum, minimum and nominal value of the uncertainties. It is clear that the size of the problem becomes rapidly intractable.

Table 3.1: Number of nodes of a scenario tree for multi-stage NMPC for different lengths of the prediction horizon (N_p) and different number of uncertainties, assuming three branches per uncertainty.

	# of uncertainties			
N$_\mathbf{P}$	**1**	**2**	**3**	**4**
1	3	9	27	81
2	9	81	729	6,561
3	27	729	19,683	531,441
4	81	6,561	531,441	43,046,721
5	243	59,049	14,348,907	3,486,784,401
6	729	531,441	387,420,489	28,242,953,648

A simple strategy to deal with the growth of the tree with the prediction horizon is to assume that the uncertainty remains constant after a certain point in time (called the robust horizon). The main idea of this simplification is that due to the receding horizon nature of NMPC, modeling the far future very accurately is not critical because all the control inputs will be recomputed at the next sampling time anyway. An example of a scenario tree illustrating this simplification can be seen in Fig. 3.2. This simplification has been shown to provide very good results in practice as can be seen in (Lucia et al., 2013a) and it will be used in the remainder of this thesis.

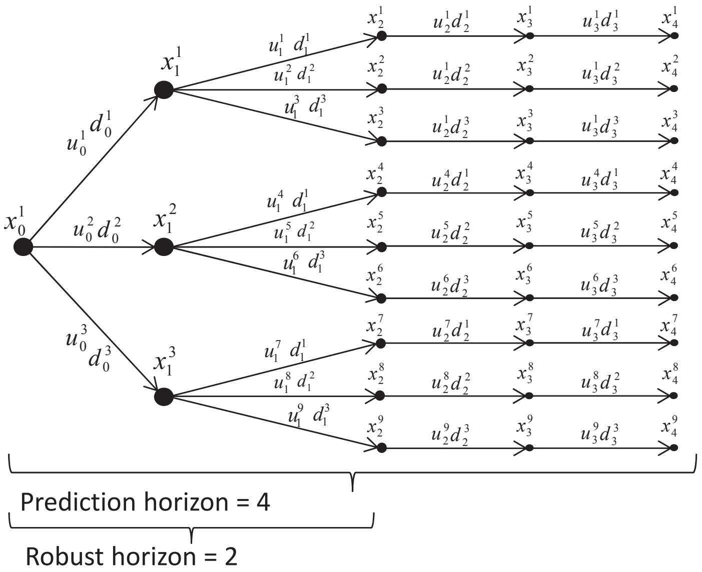

Figure 3.2: Scenario tree representation of the uncertainty evolution for multi-stage NMPC with robust horizon.

As pointed out above, the branching of the scenario tree does not only model the fact that the uncertainties may be time-varying, but it also models the fact that at the next sampling time a new tree will have to be considered. For this reason, even if the uncertainties are truly constant (but unknown), the performance of multi-stage NMPC with robust horizon $N_R = 1$ does not necessarily have to be better than the performance of multi-stage NMPC with $N_R = N_P$.

3.2 Open-loop Robust Nonlinear Model Predictive Control

The formulation of multi-stage NMPC presented in (3.5) can be modified to represent the open-loop case in which no feedback is incorporated in the prediction of the NMPC controller. For this purpose the non-anticipativity constraints are modified such that they force all control inputs to be equal at each time stage. Thus, a sequence of control inputs satisfying the constraints for all the cases of the uncertainty in the scenario tree has to be computed. The optimization problem that has to be solved at each sampling time reads as:

$$\min_{x_k^j, u_k^j \forall (j,k) \in I} \quad \sum_{i=1}^{N} \omega_i J_i(X_i, U_i), \tag{3.7a}$$

subject to:

$$x_{k+1}^j = f(x_k^{p(j)}, u_k^j, d_k^{r(j)}), \qquad \forall\, (j, k+1) \in I, \tag{3.7b}$$

$$g(x_{k+1}^j, u_k^j) \leq 0\ , \qquad \forall\ (j, k+1) \in I, \tag{3.7c}$$

$$u_k^j = u_k^l\ , \qquad \forall\ (j,k), (l,k) \in I, \tag{3.7d}$$

here the only difference with respect to the multi-stage formulation (3.5) is the modification of the non-anticipativity constraints in (3.7d).

3.3 The Importance of Recourse

This section shows a very simple example where it is easy to understand why the introduction of recourse, which is realized by the formulation of a scenario tree in multi-stage NMPC and ignored in an open-loop approach, can greatly improve the performance and the feasible set of the controller. This example has been adapted from (Scokaert and Mayne, 1998).

The following discrete-time system is considered:

$$x_{k+1} = x_k + u_k + d_k. \tag{3.8}$$

The disturbance d_k can take values at each sampling time from the interval $d_k \in [-1, 1]$. A scenario tree is generated using the maximum and the minimum possible values of d_k as it can be seen in Fig. 3.3 with a prediction horizon and a robust horizon $N_P = N_R = 3$. The stage cost is the tracking of the origin, i.e. $L = (x_k^j - 0)^2$.

First an open-loop robust MPC approach is formulated, in which all the control inputs at each stage have to be equal. The optimization problem that has to be solved at each sampling time is:

$$\min_{x_k^j, u_k^j \forall (j,k) \in I} \quad \sum_{i=1}^{N} (\omega_i J_i(X_i, U_i)), \tag{3.9a}$$

subject to:

$$x_{k+1}^j = x_k^{p(j)} + u_k^j + d_k^{r(j)}, \qquad \forall\ (j, k+1) \in I, \tag{3.9b}$$

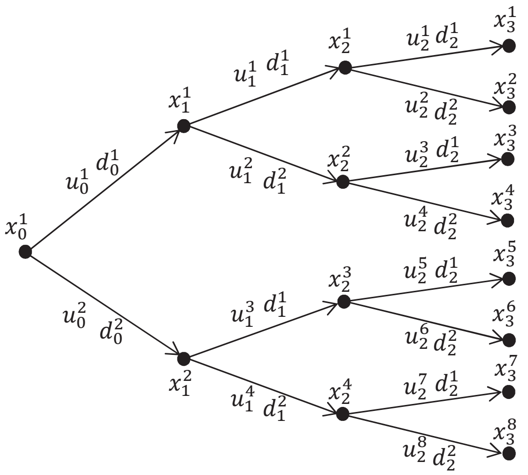

Figure 3.3: Scenario tree representation of the uncertainty with prediction horizon and robust horizon $N_P = N_R = 3$ with two realizations of the uncertainty.

$$x_0^1 = 2 \tag{3.9c}$$

$$-2.0 \leq x_k^j \leq 2.0, \qquad \forall\ (j,k) \in I, \tag{3.9d}$$

$$-1.0 \leq u_k^j \leq 1.0, \qquad \forall\ (j,k) \in I, \tag{3.9e}$$

$$u_k^j = u_k^l, \qquad \forall\ (j,k),(l,k) \in I, \tag{3.9f}$$

Using the two extreme scenarios (realization with $d_k^{r(j)} = -1$ and $d_k^{r(j)} = 1$ for $k = 0,..,N_P - 1$), it is easy to conclude that the problem (3.9) is infeasible as it is shown below. For those scenarios the state at the end of the prediction horizon can be calculated as:

$$\begin{aligned}\text{Scenario 1: } x_3^1 &= x_0^1 + u_0^1 + u_1^1 + u_2^1 + d_0^1 + d_1^1 + d_2^1 = 2 + u_0^1 + u_1^1 + u_2^1 - 1 - 1 - 1 \\ &= -1 + u_0^1 + u_1^1 + u_2^1.\end{aligned} \tag{3.10}$$

$$\begin{aligned}\text{Scenario 8: } x_3^8 &= x_0^1 + u_0^2 + u_4^1 + u_2^8 + d_0^2 + d_1^2 + d_2^2 = 2 + u_0^2 + u_1^4 + u_2^8 + 1 + 1 + 1 \\ &= 5 + u_0^2 + u_1^4 + u_2^8.\end{aligned} \tag{3.11}$$

From the equation defining the final state of the scenario 8 it is possible to infer that the control inputs have to be chosen $u_0^2 = u_1^4 = u_2^8 = -1$ in order to satisfy the state constraints. Due to the open-loop nature of this formulation, the same control inputs have to be used for all the cases of the uncertainties (imposed in (3.9f)) and this control inputs result in constraint violations for scenario 1. Then this optimization problem is infeasible and does not have a solution.

However, if a multi-stage formulation is used, which can be written as:

$$\min_{x_k^j, u_k^j \forall (j,k) \in I} \quad \sum_{i=1}^{N} \omega_i J_i(X_i, U_i)) \tag{3.12a}$$

$$\text{subject to:} \tag{3.12b}$$

$$x_{k+1}^j = x_k^{p(j)} + u_k^j + d_k^{r(j)}, \qquad \forall\, (j, k+1) \in I, \tag{3.12c}$$

$$x_0^1 = 2 \tag{3.12d}$$

$$-2.0 \leq x_k^j \leq 2.0, \qquad \forall\, (j,k) \in I, \tag{3.12e}$$

$$-1.0 \leq u_k^j \leq 1.0, \qquad \forall\, (j,k) \in I, \tag{3.12f}$$

$$u_k^j = u_k^l \text{ if } x_k^{p(j)} = x_k^{p(l)}, \qquad \forall\, (j,k), (l,k) \in I, \tag{3.12g}$$

the scenario tree structure makes it possible to adapt the control inputs (respecting the non-anticipativity constraints in (3.12g)) and it is possible to find a feasible solution. For example in this case the control inputs $u_0^2 = u_1^4 = u_2^8 = -1$ and $u_0^1 = -1, u_1^4 = u_2^8 = +1$ provide a feasible solution for the scenarios 1 and 8. It is trivial to find feasible control inputs for the rest of the scenarios.

This simple example shows the importance of considering recourse if it exists the possibility of taking new decisions (control inputs) using measurements about the system that become available in the future, as it is the case in model predictive control.

3.4 Robust Model Predictive Control with Affine Policies

The computation of open-loop control policies can lead to very conservative solutions or even to infeasible optimization problems as has been illustrated in the previous section. A traditional way to introduce feedback in the predictions to achieve a closed-loop formulation is to include a feedback matrix which depends on the future predicted state as an additional degree of freedom in the optimization problem. This was introduced in the context of MPC in (Bemporad, 1998). In this case, the control input that is calculated at each sampling time in the prediction is $u_k^j = v_k^j + Kx_k^{p(j)}$ where $K \in \mathbb{R}^{n_x \times n_x}$ is the new optimization variable. Other parametrizations, including time-varying affine (state feedback) policies or affine policies parametrized on the uncertainty instead of on the states, have also been proposed in the literature (see e.g. (Löfberg, 2003) or (Goulart et al., 2006)).

The constraints have to be satisfied for all values of the uncertainty and in order to have a direct comparison with the other presented approaches this is enforced by applying the constraints to all the nodes of the tree. Following (Goulart et al., 2006), only the nominal scenario is considered in the cost function, which can easily be extended to the average over all the scenarios if wanted. The optimization problem that is solved at each sampling time in the case of constant affine policies can be written as:

$$\min_{x_k^j, v_k^j, K \ \forall (j,k) \in I} \quad \sum_{i=1}^{1} \omega_i J_i(X_i, U_i), \tag{3.13a}$$

subject to:

$$x_{k+1}^j = f(x_k^{p(j)}, v_k^j + K x_k^{p(j)}, d_k^{r(j)}), \qquad \forall \, (j, k+1) \in I, \tag{3.13b}$$

$$g(x_{k+1}^j, v_k^j + K x_k^{p(j)}) \leq 0 \, , \qquad \forall \, (j, k+1) \in I, \tag{3.13c}$$

$$v_k^j = v_k^l \, , \qquad \forall \, (j, k), (l, k) \in I. \tag{3.13d}$$

For the case of time varying state feedback policies, $u_k^j = v_k^j + K x_k^{p(j)}$ with $K_k \in \mathbb{R}^{n_x \times n_x}$, the optimization problem is formulated as:

$$\min_{x_k^j, v_k^j, K_k \ \forall (j,k) \in I} \quad \sum_{i=1}^{1} \omega_i J_i(X_i, U_i), \tag{3.14a}$$

subject to:

$$x_{k+1}^j = f(x_k^{p(j)}, v_k^j + K_k x_k^{p(j)}, d_k^{r(j)}), \qquad \forall \, (j, k+1) \in I, \tag{3.14b}$$

$$g(x_{k+1}^j, v_k^j + K_k x_k^{p(j)}) \leq 0 \, , \qquad \forall \, (j, k+1) \in I, \tag{3.14c}$$

$$v_k^j = v_k^l \, , \qquad \forall \, (j, k), (l, k) \in I. \tag{3.14d}$$

This chapter presented four different designs for robust NMPC which will be used in the remainder of this thesis. Multi-stage NMPC is described in (3.5), which also includes standard NMPC if the number of scenarios is $N = 1$, and min-max NMPC if the maximization operator is used instead of the summation in the cost. An open-loop robust NMPC implementation in which a sequence of control inputs has to satisfy the constraints for all the cases of the uncertainty can be found in (3.7). Robust NMPC with affine constant policies is defined in (3.13) and robust NMPC with affine time-varying policies is described in (3.14). These names will be used for referring to the different approaches in the remainder of the thesis.

Chapter 4

Stability Properties of Multi-stage NMPC

Since the very first paper on model predictive control in the work of (Richalet et al., 1978), it was shown that MPC had significant advantages over other control techniques, which led to it being taken up quickly in the process industry. However, it was later realized that the simple solution of a finite horizon problem is not enough to achieve a guarantee for the stability of the closed-loop system when such a receding horizon strategy is applied, as shown e.g. in (Bitmead et al., 1990). During the decade of the 1990s, many different modifications of the MPC formulation were proposed to ensure the closed-loop stability of a system controlled using MPC. Most of these results are summarized and condensed in the important publication by (Mayne et al., 2000). The different approaches are all based on modifications of the original MPC problem. These modifications usually include a so-called terminal penalty term that penalizes the states at the last stage of the prediction horizon and a terminal constraint which forces the state at the last stage of the prediction horizon to lie within a pre-computed terminal set.

In this chapter, these ideas are extended to the multi-stage NMPC case using the concept of Input-to-State Stability (ISS) (Jiang and Wang, 2001). ISS was recently presented as an unifying framework for the stability analysis of robust NMPC (see (Magni and Scattolini, 2007), (Limon et al., 2009)). The work presented in this chapter has been done in collaboration with Daniel Limon and it is largely identical to the work published in (Lucia et al., 2014c).

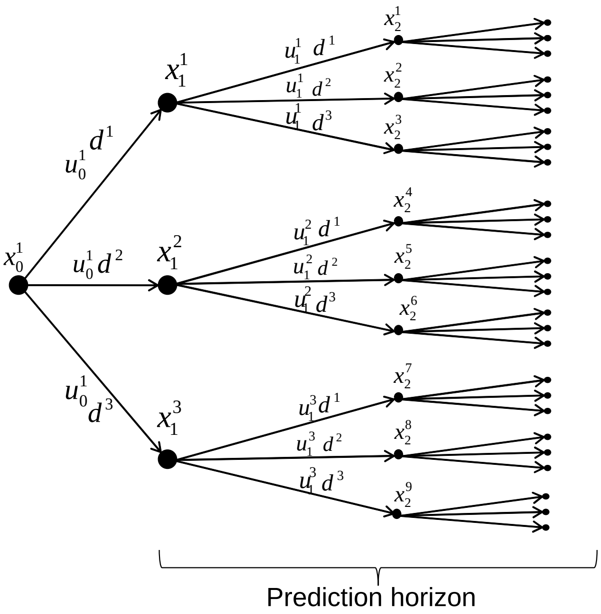

Figure 4.1: Scenario tree representation of the uncertainty with embedded non-anticipativity constraints.

In order to facilitate the notation used during the stability proofs presented in this chapter, the non-anticipativity constraints are embedded in the formulation by reducing the number of different control inputs that are available at each stage so that the same control input variable is used at each node as can be seen in Fig. 4.1. Then the indices for the description of the uncertain nonlinear system are used as:

$$x_{k+1}^c = f(x_k^j, u_k^j, d^r), \tag{4.1}$$

where $x_k^j \in \mathbb{R}^{n_x}$ denotes the state vector at stage k. The *child* node $x_{k+1}^c \in \mathbb{R}^{n_x}$ is obtained when the control input $u_k^j \in \mathbb{R}^{n_u}$ is applied at the parent node x_k^j and the realization of the uncertainty is $d^r \in \mathbb{R}^{n_d}$. Since the same values of the uncertainty are used at each time stage, the subindex k is dropped for d^r for the formulations of the proofs. For a uniform scenario tree, the index of the children node can be calculated as a function of the parent node and the realization of the uncertainty $c = s(j-1)+r$, where s denotes the number of realizations of the uncertainty (or branches of the scenario tree at each node). Note that it is also possible to determine the realization r of the uncertainty that leads to a node x_k^j using the modulo operator $r = \text{mod}(j, s) + 1$. Without loss of generality, it is considered that the scenario tree is generated at each sampling instance starting from stage $k = 0$. For the analysis of the multi-stage NMPC, some basic assumptions are necessary.

Assumption 1 (Basic Assumptions)**.**
A. The system (4.1) *is assumed to be uniformly continuous in d^r and to have an equilibrium point at the origin, i.e., $f(0,0,0) = 0$.*
B. Control and states are required to satisfy constraints (imposed in the optimization problem), $x_k^j \in \mathbb{X} \subseteq \mathbb{R}^n$ and $u_k^j \in \mathbb{U} \subseteq \mathbb{R}^m$, $\forall (j,k) \in I$, where $\mathbb{X}$ and $\mathbb{U}$ are compact sets.
C. The uncertainty is assumed to take one of the different s possible realizations at each node, i.e, $d^r \in \mathbb{D} = \{d^1, ..., d^s\}$ with probability $\{\pi^1, ..., \pi^s\}$ respectively and the set $\mathbb{D}$ is the same for all stages.
D. Perfect measurement information is available, that is, the states are available before a new decision has to be computed each sampling time.

In order to guarantee the closed-loop stability of multi-stage NMPC it is necessary to modify the optimization problem (3.5) presented in the previous chapter by adding a terminal penalty term and terminal constraints. In addition, it is easier to prove the stability of the multi-stage NMPC approach if the cost function is formulated stage-wise instead of scenario-wise, such that the cost for each node in the tree is summed at each stage. Then, the optimization problem at each sampling instance is formulated as:

$$\min_{x_k^j, u_k^j \forall (j,k) \in I} \quad \sum_{k=0}^{N_P-1} \sum_{j=1}^{s^{k+1}} \pi_k^j \ell(x_k^j, u_k^j) + \sum_{j=1}^{s^{N_P}} \pi_{N_P}^j V_f^r(x_{N_P}^j), \tag{4.2a}$$

subject to:

$$x_{k+1}^c = f(x_k^j, u_k^j, d^r), \qquad \forall\, (j, k+1) \in I, \tag{4.2b}$$

$$x_k^j \in \mathbb{X}\ , \qquad \forall\ (j,k) \in I, \tag{4.2c}$$

$$u_k^j \in \mathbb{U}\ , \qquad \forall\ (j,k) \in I, \tag{4.2d}$$

$$x_{N_P}^j \in \mathbb{X}_f\ , \qquad \forall\ (j, N_P) \in I, \tag{4.2e}$$

where in the stage cost $\ell : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$, x_k^j represents the parent node from which with the control input u_k^j and the realization of the uncertainty d^r the children node x_{k+1}^c is obtained. Since s denotes the number of realizations of the uncertainty, s^k is the number of nodes that the scenario tree has at stage k. $V_f^r : \mathbb{R}^{n_x} \to \mathbb{R}$ are possibly different terminal penalty terms, depending on which realization d^r led to the corresponding leaf node. Each node x_k^j has a certain probability of being reached from its parent node that is equal to the probability $\{\pi^1, ..., \pi^s\}$ of the corresponding realization of the uncertainty.

The accumulated probability π_k^j of the node x_k^j is defined as the probability of this node to be reached from the root node x_0^1 and it can be calculated as the product of the probabilities of the realizations of the uncertainty that occur along the path from x_0^1 until x_k^j. For instance, the probability associated to node x_2^3 in Fig. 4.1 is calculated as $\pi_2^3 = \pi^1 \cdot \pi^3$. In this way, the sum of the accumulated probabilities of all the nodes in each stage is equal to one. The accumulated probabilities π_k^j are in general different from the probabilities of each scenario ω_i defined in (3.5). The accumulated probabilities at the last stage $\pi_{N_P}^j$ coincide with the probabilities of each scenario ω_i. The set of states for which a solution of problem (4.2) exists is denoted as $\mathbb{X}_A(N_P)$ and it is called feasibility region.

Remark 1. *Note that there is only one terminal $\mathbb{X}_f$ set but there are different terminal penalty terms V_f^r, depending on the realization of the uncertainty in the last stage. By adapting these penalties, the guarantee of stability can be facilitated.*

The value of the cost function defined in (4.2a) for a given initial state x_0^1, a given prediction horizon N_P and a given set of control inputs $\tilde{u}$ is denoted as $J(x_0^1, \tilde{u}, N_P)$. The set of optimal control inputs that solve problem (4.2) are denoted as $\tilde{u}^*$ and as usual only the first stage control input is used (u_0^{1*}). This is repeated in a receding horizon fashion obtaining implicitly the NMPC control law κ_{N_P}. The optimal value of the cost function with prediction horizon N_P at each sampling instance is denoted as

$$V_{N_P}(x_0^1) = J(x_k^j, \tilde{u}^*, N_P). \tag{4.3}$$

4.1 Input-to-State Stability of Multi-stage NMPC

4.1.1 Notation and Basic Definitions

The open-loop system (4.1) is assumed to be controlled by a certain control law $u = \kappa(x)$, so that the closed-loop system is expressed as:

$$x_{k+1} = F(x_k, d_k), \tag{4.4}$$

where d_k represents the actual realization of the disturbance and x_k is the current state of the system. The following definitions are necessary for the analysis of the stability of the origin of multi-stage NMPC. They can be found in the literature (Khalil, 2002;

Magni and Scattolini, 2007; Limon et al., 2009) or (Rawlings and Mayne, 2009) and they are shortly presented here for the sake of completeness.

Definition 1 (Robust positively invariant (RPI) set). *A set $\Omega \subseteq \mathbb{X}^n$ that contains the origin is said to be a robust positively invariant set for system (4.4) with respect to $\mathbb{D}$ if for all $x \in \Omega$ it holds that $F(x, d) \in \Omega$ for all $d \in \mathbb{D}$.*

Definition 2 (Class $\mathcal{K}$-functions). *A function γ: $\mathbb{R}_+ \to \mathbb{R}_+$ is said to belong to class $\mathcal{K}$ if it is strictly increasing and $\gamma(0) = 0$. It is said to belong to class $\mathcal{K}_\infty$ if $\gamma(a) \to \infty$ as $a \to \infty$.*

Definition 3 (Class $\mathcal{KL}$-functions). *A function β: $\mathbb{R}_+ \times \mathbb{Z}_+ \to \mathbb{R}_+$ is said to belong to class $\mathcal{KL}$ if for each fixed k, $\beta(\cdot, k)$ belongs to class $\mathcal{K}$ and for each fixed $s \geq 0$, $\beta(s, \cdot)$ is decreasing and $\beta(s, k) \to 0$ as $k \to \infty$.*

Definition 4 (Regional Input-to-State Practical Stability). *Given a RPI set $\Gamma \in \mathbb{R}^n$, with the origin in its interior, the closed-loop system (4.4) is said to be ISpS in Γ with respect to d if there exist a $\mathcal{KL}$-function β, a $\mathcal{K}$-function γ, and a constant $c \geq 0$ such that*

$$|x_k| \leq \beta(|x_0|, k) + \gamma(|d_{[0,k-1]}|) + c, \tag{4.5}$$

for all $k \geq 0$, $x_0 \in \Gamma$, $d \in \mathbb{D}$ where $d_{[0,k-1]}$ indicates the sequence of realizations of the uncertainty from time 0 to time $k-1$. If it holds for $c = 0$, then the system 4.1 is said to be ISS in Γ.

The concept of Input-to-State stability implies asymptotic stability of the nominal system and that the effect of the uncertainty on the states is bounded (Limon et al., 2009). Also, the real uncertain system will converge asymptotically to the origin if the uncertainty vanishes. This is the main difference to the Input-to-State practical stability property, in which it is not guaranteed that the system will converge to the origin for the case of vanishing uncertainty, but it will converge to a compact set around the origin.

The following definition was introduced in (Limon et al., 2006) and describes the existence of a Lyapunov function that can be used for the analysis of the ISpS property.

Definition 5 (ISpS Lyapunov function in Γ). *Assuming that Γ is an RPI set with the origin in its interior and that there exists a compact set $\Theta \subseteq \Gamma$ with the origin as an interior point. A function $V : \mathbb{R}^n \to \mathbb{R}_+$ is called an ISpS-Lyapunov function in Γ for* *the closed loop system (4.4) with respect to d, if there exist $\mathcal{K}_\infty$-functions α_1, α_2, α_3, a $\mathcal{K}$-function σ and constants c_1, c_2 such that:*

$$V(x) \geq \alpha_1(|x|),\ \forall x \in \Gamma, \tag{4.6a}$$

$$V(x) \leq \alpha_2(|x|) + c_1,\ \forall x \in \Theta, \tag{4.6b}$$

$$\Delta V(x,d) \leq -\alpha_3(|x|) + \sigma(|d|) + c_2,\ \forall x \in \Gamma,\ d \in \mathbb{D}, \tag{4.6c}$$

where $\Delta V(x,d) = V(F(x,d)) - V(x)$.

After introducing these definitions, it is possible to state the following theorem, which has been presented in (Limon et al., 2006) and in (Limon et al., 2009).

Theorem 1. *Consider that the system (4.4) fulfills Assumption 1. If this system admits an ISpS-Lyapunov function in Γ with respect to d, then it is ISpS in Γ with respect to d.*

Proof. See (Limon et al., 2006) or (Limon et al., 2009). □

4.1.2 Input-to-State Practical Stability of Multi-stage NMPC

In this subsection the extension of the Input-to-State practical stability property to the case of multi-stage NMPC is presented. In order to derive this result, it is necessary to introduce the following assumptions.

Assumption 2 (Assumptions on the stage cost)**.** *There exist a constant $c_3 \geq 0$ such that: $\gamma_1(|x|) \leq \ell(x,u) \leq \gamma_2(|x|) + c_3$, where γ_1, γ_2 are $\mathcal{K}_\infty$ functions.*

Assumption 3 (Assumptions on the terminal ingredients)**.**
A. $\mathbb{X}_f \subseteq \mathbb{X}$, $0 \in \mathbb{X}_f$ and $\mathbb{X}_f$ is closed.
There exists a feedback policy $\kappa_f(x)$ inside $\mathbb{X}_f$ such that:
B. $\kappa_f(x) \in \mathbb{U}$, $\forall x \in \mathbb{X}_f$.
C. $f(x, \kappa_f(x), d) \in \mathbb{X}_f$, $\forall x \in \mathbb{X}_f$, $\forall d \in \mathbb{D}$.
There exist a set of terminal cost functions $V_f^r(x)$ such that:
D. $\gamma_3(|x|) \leq V_f^r(x) \leq \gamma_4(|x|)$, $\forall x \in \mathbb{X}_f$, $\forall V_f^r$ with $r \in \{1, ..., s\}$ and γ_3, γ_4 are $\mathcal{K}_\infty$-functions.
E. $\sum_{r=1}^{s} \pi^r V_f^r(f(x, \kappa_f(x), d^r)) - V_f^l(x) \leq -\ell(x, \kappa_f(x)) + \gamma_d(|d|)$, $\forall x \in \mathbb{X}_f$, $\forall d \in \mathbb{D}$, with $l \in \{1, ..., s\}$ and γ_d is a $\mathcal{K}_\infty$-function.

Remark 2. *The only nonstandard assumption is 3.E. As long as assumption 3.E is satisfied, the probabilities π^r can be chosen to enhance the performance of the controller. For instance, if the uncertainty can be estimated, the probabilities can be adjusted according to this estimation, achieving a better performance, but still guaranteeing robust stability and robust constraint satisfaction.*

One important ingredient for stability proofs of NMPC schemes is the monotonicity of the cost function. In the next lemma, we show that the cost function of the multi-stage NMPC scheme has this property.

Lemma 1 (Monotonicity of the cost function)**.** *Suppose that assumptions 1, 2 and 3 hold. Then, there exists a $\mathcal{K}_\infty$-function γ_5 such that*

$$V_{k+1}(x_0^1) \leq V_k(x_0^1) + \gamma_5(|\bar{d}|),\ \forall k \in \mathbb{Z}_{[0,N_P-1]},\ \forall \bar{d} \in \mathbb{D},\ \forall x_0^1 \in \mathbb{X}_A(N_P). \tag{4.7}$$

Proof. Let $\tilde{u}^+$ be a feasible, but not necessarily optimized, set of control inputs of the problem with horizon $N_P + 1$. It contains the optimized control inputs of the problem with horizon N_P, denoted as $\tilde{u}^*$, and the not necessarily optimized control inputs for stage $N_P + 1$, denoted as $u_{N_P}^j$. Provided that $x_{N_P}^j \in \mathbb{X}_f\ \forall (j, N_P) \in I$ (imposed in (3.5)), the inputs $u_{N_P}^j$ are calculated using the auxiliary control law $\kappa_f(x)$, so that $u_{N_P}^j = \kappa_f(x_{N_P}^j)$. In this manner, it is guaranteed that the system at time $N_P + 1$ satisfies both input and state constraints because of Assumptions 3.B and 3.C.

Then, the not necessarily optimal cost function of the problem with horizon $N_P + 1$ can be written as:

$$\begin{aligned} J(x_0^1, \tilde{u}^+, N_P + 1) &= V_{N_P}(x_0^1) + \sum_{j=1}^{s^{N_P}} \pi_{N_P}^j \ell(x_{N_P}^j, u_{N_P}^j) \\ - \sum_{j=1}^{s^{N_P}} \pi_{N_P}^j V_f^r(x_{N_P}^j) &+ \sum_{j=1}^{s^{N_P+1}} \pi_{N_P+1}^j V_f^r(f(x_{N_P}^j, u_{N_P}^j, d^r)). \end{aligned} \tag{4.8}$$

Making use of Assumption 3.E it is possible to prove that

$$\begin{aligned} &\sum_{j=1}^{s^{N_P+1}} \pi_{N_P+1}^j V_f^r(f(x_{N_P}^j, u_{N_P}^j, d_{N_P}^r)) - \sum_{j=1}^{s^{N_P}} \pi_{N_P}^j V_f^r(x^j) \\ &\leq - \sum_{j=1}^{s^{N_P}} \pi_{N_P}^j \ell(x_{N_P}^j, u_{N_P}^j) + \gamma_5(|\bar{d}|). \end{aligned} \tag{4.9}$$

This follows from applying Assumption 3.E to each of the s^{N_P} leaf nodes of the problem with horizon N_P. Then, it is clear that $V_{N_P+1}(x_0^1)$, the optimal cost of the problem with horizon $N_P + 1$, fulfills

$$V_{N_P+1}(x_0^1) \leq J(x_0^1, \tilde{u}^+, N_P + 1) \leq V_{N_P}(x_0^1) + \gamma_5(|\bar{d}|), \tag{4.10}$$

for all $x \in \mathbb{X}_A(N_P)$ and for all $\bar{d} \in \mathbb{D}$. This is true for all N_P and thus (4.7) is proved. □

Now we prove Input-to-State practical stability of multi-stage NMPC extending existing results as the ones presented for nominal (Rawlings and Mayne, 2009) and min-max NMPC (Raimondo et al., 2007).

Theorem 2 (ISpS of multi-stage NMPC)**.** *Provided that Assumptions 1, 2 and 3 hold, the system (4.1) with the receding horizon control law κ_{N_P} is ISpS with robust invariant region $\mathbb{X}_A(N_P)$.*

Proof. The theorem can be proved by showing that the optimal value of the cost function (4.3) is an ISpS Lyapunov function for the closed-loop system in $\mathbb{X}_A(N_P)$. It is easy to prove that under the given assumptions $\mathbb{X}_A(N_P)$ is robust positively invariant (Raimondo et al., 2007). Then, we establish the three conditions that $V_{N_P}(x_0^1)$ has to fulfill to be an ISpS Lyapunov function.

(i) First, the lower bound (4.6a) will be proved, which follows from Assumption 2 and

$$\begin{aligned} V_{N_P}(x_0^1) &\geq \sum_{k=0}^{N_P-1} \sum_{j=1}^{s^k} \pi_k^j \ell(x_k^j, u_k^j) \\ &\geq \min_{(j,k)\in I} \pi_k^j \ell(x_k^j, u_k^j) \geq \gamma_1(|x|), \end{aligned} \tag{4.11}$$

for all $x \in \mathbb{X}_A(N_P)$, where the last inequality holds because of Assumption 2.

(ii) The upper bound (4.6b) can be proved by using the monotonocity property derived in Lemma 1. By applying it recursively

$$\begin{aligned} &V_{N_P}(x_0^1) \leq V_{N_P-1}(x_0^1) + \gamma_5(|d|) \leq ... \leq V_0(x_0^1) \\ &+ N_P\gamma_5(|\bar{d}|) = V_f^r(x_0^1) + N_P\gamma_5(|d|) \leq \gamma_4(|x_0^1|) + N_P\gamma_5(|\bar{d}|), \end{aligned} \tag{4.12}$$

for all $x_0^1 \in \mathbb{X}_A(N_P)$, where the last inequality follows from Assumption 3.D.

(iii) The key element of MPC stability proofs is the descent property (4.6c). In order to show that this property holds, we write the optimal value of the cost function of the problem with horizon N_P as:

$$\begin{aligned}
&V_{N_P}(x_0^1) = \\
&\sum_{r=1}^{s} \pi^r V_{N_P-1}(f(x_0^1, u_0^1, d^r)) + \ell(x_0^1, u_0^1) \\
&= \ell(x_0^1, u_0^1) + \sum_{r=1}^{s} \pi^r V_{N_P}(f(x_0^1, u_0^1, d^r)) \\
&+ \sum_{r=1}^{s} \pi^r V_{N_P-1}(f(x_0^1, u_0^1, d^r)) - \sum_{r=1}^{s} \pi^r V_{N_P}(f(x_0^1, u_0^1, d^r))
\end{aligned}$$

Reordering and grouping summations we have that:

$$\begin{aligned}
&- V_{N_P}(x_0^1) + \ell(x_0^1, u_0^1) + \sum_{r=1}^{s} \pi^r V_{N_P}(f(x_0^1, u_0^1, d^r)) \\
&= \sum_{r=1}^{s} \pi^r [V_{N_P}(f(x_0^1, u_0^1, d^r)) - V_{N_P-1}(f(x_0^1, u_0^1, d^r))] \\
&\leq \gamma_5(|\bar{d}|),
\end{aligned}$$

where the inequality holds because of the monotonicity property derived in Lemma 1. Then:

$$\ell(x_0^1, u_0^1) + \sum_{r=1}^{s} \pi^r V_{N_P}(f(x_0^1, u_0^1, d^r)) \leq V_{N_P}(x_0^1) + \gamma_5(|\bar{d}|), \tag{4.13}$$

which implies

$$\sum_{r=1}^{s} \pi^r V_{N_P}(f(x_0^1, u_0^1, d^r)) - V_{N_P}(x_0^1) \leq -\gamma_1(|x_0^1|) + \gamma_5(|\bar{d}|), \tag{4.14}$$

for all $x_0^1 \in \mathbb{X}_A(N_P)$ and for all $d^r \in \mathbb{D}$.

The inequality in (4.14) shows an average descent in the optimal value of the cost function over all the possible realizations of the uncertainty. In order to prove ISpS of the closed loop system, a decrease for each realization is needed as shown in (4.6c). To achieve this condition, it is possible to prove that for all $x_0^1 \in \mathbb{X}_A(N_P)$, it holds for any $d^l \in \mathbb{D}$ that the difference between the individual costs $V_{N_P}(f(x_0^1, u_0^1, d^l))$ and the average cost over all $l = 1, ..., s$ is bounded:

$$V_{N_P}(f(x_0^1, u_0^1, d^l)) \leq \sum_{r=1}^{s} \pi^r V_{N_P}(f(x_0^1, u_0^1, d^r)) + \gamma_6(|\bar{d}|) + \rho, \tag{4.15}$$

This can be derived taking into account that V_N is such that for all $d^l, d^r \in \mathbb{D}$ $|V_{N_P}(f(x, u, d^r)) - V_{N_P}(f(x, u, d^l))| \leq \gamma_V(|\bar{d}|) + \rho_V$ where $\gamma_V(|\bar{d}|)$ is a $\mathcal{K}_\infty$-function and ρ_V is a constant which is nonzero only in the case of discontinuity of V_{N_P} and always exists because V_{N_P} is bounded. The dependence on $\bar{d} \in \mathbb{D}$ of the bound given by $\gamma_V(|\bar{d}|)$ follows from the the uniform continuity of $f(x, u, d^r)$ in d^r given by Assumption 1. By bounding each one of the cost functions inside the sum, then (4.15) holds with $\gamma_6(|\bar{d}|) = \gamma_V(|\bar{d}|)$ and $\rho = \rho_V$. Then for all $d^r \in \mathbb{D}$ and for all $x_0^1 \in \mathbb{X}_A(N_P)$ we have that:

$$\begin{aligned} V_{N_P}(f(x_0^1, u_0^1, d^r)) - V_{N_P}(x_0^1) &\leq -\gamma_1(|x_0^1|) + \gamma_5(|\bar{d}|) + \gamma_6(|\bar{d}|) + \rho \\ &= -\gamma_1(|x_0^1|) + \gamma_7(|\bar{d}|) + \rho. \end{aligned} \tag{4.16}$$

This proves condition (4.6c) and therefore Theorem 2 is also proved. □

4.1.3 Achieving Input-to-State Stability of Multi-stage NMPC

Note that the terms $\gamma_7(|\bar{d}|)$ and ρ that appear in (4.16) only allow for a convergence of the system to a neighborhood of the origin even if there is no disturbance. The reason for this stems from the design of the controller, which always takes into account all the possible scenarios. In a very similar manner as demonstrated in (Lazar et al., 2008), (Raimondo et al., 2009), and (Limon et al., 2009) for min-max NMPC, it is possible to prove the Input-to-State stability of the multi-stage NMPC by using a *dual-mode* controller. That is, the implicit control law derived from the solution of the NMPC problem is used until the system reaches the terminal set $\mathbb{X}_f$. Once the system is in this region the terminal control law κ_f is applied. ISS can be proved using the fact that the terminal set $\mathbb{X}_f$ can be reached in finite time, and imposing that $V_f^T = \sum_{r=1}^{s} \pi^r V_f^r$ is an ISS-Lyapunov function for the system (4.1) with the terminal control law κ_f in the set $\mathbb{X}_f$.

4.2 Multi-stage NMPC with Robust Horizon

In this section the stability proof is extended for the case of a robust horizon N_R smaller than the prediction horizon N_P. The optimization problem solved at each sampling time for multi-stage NMPC with prediction horizon N_P and robust horizon N_R is:

$$\min_{x_k^j, u_k^j \forall (j,k) \in I} \quad \sum_{k=0}^{N_R-1} \sum_{j=1}^{s^{k+1}} \pi_k^j \ell(x_k^j, u_k^j) + \sum_{j=1}^{s^{N_R}} \pi_{N_R}^j \underbrace{\left(\sum_{k=N_R}^{N_P-1} \ell(x_k^j, \kappa_f(x_k^j)) + V_f^r(x_{N_P}^j) \right)}_{W_f^r(x_{N_R}^j, \kappa_f(x_{N_R}^j), d^r)}, \tag{4.17a}$$

subject to:

$$
\begin{aligned}
x^c_{k+1} &= f(x^j_k, u^j_k, d^r), & \forall\,(j,k+1) \in I \text{ with } k = 0, ..., N_R - 1, \qquad &(4.17\text{b})\\
x^j_{k+1} &= f(x^j_k, \kappa_f(x^j_k), d^r), & \forall\,(j,k+1) \in I \text{ with } k = N_R, ..., N_P - 1, \qquad &(4.17\text{c})\\
x^j_k &\in \mathbb{X}\,, & \forall\,(j,k) \in I,\ \text{ with } k = 0, ..., N_R - 1, \qquad &(4.17\text{d})\\
u^j_k &\in \mathbb{U}\,, & \forall\,(j,k) \in I,\ \text{ with } k = 0, ..., N_R - 1, \qquad &(4.17\text{e})\\
x^j_{N_R} &\in \mathbb{X}_f\,, & \forall\,(j,N_R) \in I, \qquad &(4.17\text{f})
\end{aligned}
$$

The new terminal cost $W^r_f(\cdot)$ includes the original stage cost $\ell(\cdot)$ applied from the robust horizon N_R until $N_P - 1$ using the terminal control law as control input and the original terminal penalty term $V^r_f(\cdot)$ applied to the leaf nodes at stage N_P. Note that the accumulated probabilities remain constant after the robust horizon, and therefore the same probability $\pi^j_{N_R}$ can be used in (4.17a) for the stage and terminal cost after the robust horizon. For simplicity in the notation we denote the new terminal cost $W^r_f(x^j_{N_R}, \kappa_f(x^j_{N_R}), d^r)$ as $W^r_f(x^j_{N_R})$. The stability proof can be done in a similar form as in the case of the complete tree. Assumption 3.E is substituted by the following assumption.

Assumption 4 (Robust horizon assumption)**.**
A prediction horizon N_P and a robust horizon N_R with $N_P \geq N_R$ are chosen such that Assumption 3.E holds with $W^r_f(\cdot)$ as terminal cost function. That is, $\sum_{r=1}^{s} \pi^r W^r_f(f(x, \kappa_f(x), d^r)) - W^l_f(x) \leq -\ell(x, \kappa_f(x)) + \gamma_e(|d|)$, $\forall x \in \mathbb{X}_f$, $\forall d \in \mathbb{D}$, with $l \in \{1, ..., s\}$, γ_e is a $\mathcal{K}_\infty$-function.

Remark 3. *Denoting $f(x^j_k, \kappa_f(x^j_k), d^r)$ as $f^r_{\kappa_f}(x^j_k)$ for simplicity, it is possible to write the left hand side of Assumption 4 as*

$$
\begin{aligned}
&\sum_{r=1}^{s} \pi^r W^r_f(f^r_{\kappa_f}(x^j_k)) - W^l_f(x^j_k) =\\
&\sum_{r=1}^{s} \pi^r \left(\sum_{k=N_R}^{N_P-1} \ell(f^r_{\kappa_f}(x^j_k)), \kappa_f(f^r_{\kappa_f}(x^j_k))) + V^r_f(f^r_{\kappa_f}(x^j_k)) \right) - \sum_{k=N_R}^{N_P-1} \ell(x^j_k, \kappa_f(x^j_k)) + V^l_f(x^j_{N_P}) =\\
&\sum_{r=1}^{s} \pi^r V^r_f(f^r_{\kappa_f}(x^j_k))) - V^l_f(x^j_{N_P}) + \sum_{k=N_R}^{N_P-1} \left(\sum_{r=1}^{s} \pi^r \ell(f^r_{\kappa_f}(x^j_k), \kappa_f(f^r_{\kappa_f}(x^j_k))) - \ell(x^j_k, \kappa_f(x^j_k)) \right)\\
&\leq -\sum_{r=1}^{s} \pi^r \ell(x^j_k, \kappa_f(x^j_k)) + \gamma_e(|d^r|).
\end{aligned}
$$

As it can be seen from the expression above, If $N_R = N_P$, the second term of the left hand side of the inequality disappears and the assumption is reduced to Assumption 3.E. For $N_R < N_P$, and if this term is negative, Assumption 4 is a relaxed assumption with *respect to Assumption 3.E and therefore the domain of attraction may be increased as will be illustrated in the next section. Note that for this term to be negative it is only required that the average stage cost decreases in the terminal set along the time steps* $k = N_R ... N_P - 1$.

The optimal value of the problem with robust horizon N_R and prediction horizon N_P can be written as:

$$V_{N_P,N_R}(x_0^1) = J(x_k^j, \tilde{u}^*, N_P, N_R), \tag{4.18}$$

where $\tilde{u}^*$ is the set of optimal control inputs that solve problem (4.17). Then, it is possible to proof ISpS of the multi-stage approach with robust horizon making use of the following lemma.

Lemma 2. *If Assumptions 1, 2, 3.A-D and 4 hold, then there exists a* $\mathcal{K}_\infty$*-function* γ_8 *such that*

$$\begin{aligned} &V_{j+1,l+1}(x_0) \leq V_{j,l}(x_0) + \gamma_8(|\bar{d}|), \\ &\forall j \in \mathbb{Z}_{[0,N_P-1]}, \forall l \in \mathbb{Z}_{[0,N_R-1]} \text{ with } j \geq l, \ \forall \bar{d} \in \mathbb{D}. \end{aligned}$$

Proof. The not necessarily optimal cost function of the problem with prediction horizon $N_P + 1$ and robust horizon $N_R + 1$ can be written as:

$$\begin{aligned} J(x_k^j, \tilde{u}^+, N_P+1, N_R+1) = V_{N_P,N_R}(x_0^1) + \sum_{j=1}^{s^{N_R+1}} \pi_{N_R}^j \ell(x_{N_R}^j, u_{N_R}^j) \\ - \sum_{j=1}^{s^{N_R}} \pi_{N_R}^j W_f^r(x_{N_R}^j) + \sum_{j=1}^{s^{N_R+1}} \pi_{N_R+1}^j W_f^r(f(x_{N_R}^j, \kappa(x_{N_R}^j), d^r)). \end{aligned}$$

By applying Assumption 4 to all the nodes at stage N_R it is possible to prove that:

$$\begin{aligned} &- \sum_{j=1}^{s^{N_R}} \pi_{N_R}^i W_f^r(x_{N_R}^j) + \sum_{j=1}^{s^{N_R+1}} \pi_{N_R+1}^j W_f^r(f(x_{N_R}^j, \kappa(x_{N_R}^j), d^r)) \\ &\leq - \sum_{j=1}^{s^{N_R+1}} \pi_{N_R}^j \ell(x_{N_R}^j, u_{N_R}^j) + \gamma_8(|\bar{d}|). \end{aligned}$$

Then, the optimal cost of the problem with prediction horizon $N_P + 1$ and robust horizon $N_R + 1$ fulfills

$$\begin{aligned} &V_{N_P+1,N_R+1}(x_0^1) \leq J(x_k^j, \tilde{u}^+, N_P+1, N_R+1) \\ &\leq V_{N_P,N_R}(x_0^1) + \gamma_8(|\bar{d}|), \end{aligned}$$

and Lemma 2 is proved. □

Now we state in the following theorem the Input-to-State practical stability of multi-stage NMPC with robust horizon.

Theorem 3 (ISpS of multi-stage NMPC with robust horizon)**.** *Provided that assumptions 1, 2, 3.A-D and assumption 4 hold, the system (4.1) with the receding horizon control law* κ_{N_P,N_R} *is ISpS with robust invariant region* $\mathbb{X}_A(N_P, N_R)$*.*

Proof. The proof is analogous to the one presented for Theorem 2 and is omitted here for brevity. □

4.3 Discussion

This chapter presented a stability guaranteeing formulation of multi-stage nonlinear model predictive control based on the addition of a terminal cost and a terminal constraint to the original formulation presented in Chapter 3.

The necessary assumptions to guarantee stability of the multi-stage approach in particular and of any NMPC approach in general are not always fulfilled. It is very difficult to find in practice a terminal set which is a robust positively invariant set for a general nonlinear system, as well as to find its associated terminal control law. Another challenge is that the optimal cost of the MPC problem does not have to be a Lyapunov function of the closed-loop system if an economic cost function is used. In that case some additional assumptions are required (such as dissipativity of the system) to establish the stability of an NMPC scheme with an economic cost function, as recently studied in (Angeli et al., 2012).

For these reasons, in the rest of the thesis, the original formulation of multi-stage NMPC presented in (3.5) will be considered, and the stability of the controller is analyzed by simulation studies. In order to illustrate the stability guaranteeing formulation presented in this chapter, an example is provided in Chapter 9 in which multi-stage NMPC with an a priori guarantee for stability and recursive feasibility is applied.

Part II

Efficient Implementation and Solution

Chapter 5

Formulation and Solution of Dynamic Optimization Problems

5.1 Discretization Methods

Solving an optimal control problem involves the solution of an infinite-dimensional optimization problem, because it contains the dynamics of a system, which is typically described by Ordinary Differential Equations (ODEs) or Differential Algebraic Equations (DAEs), and the computation of control actions for all time instants within a given time interval. The exact solution of these type of problems is difficult and therefore only a numerical approximate solution of the problems is usually obtained. There are two main approaches to solve the resulting optimization problems numerically. The first type of methods, called indirect methods, apply the Pontryagin minimum principle and then discretize the resulting problem to obtain a solution. In contrast, this thesis – as well most of the modern dynamic optimization solvers – focuses on the application of direct methods. These approaches perform a discretization in order to convert the original infinite-dimensional optimal control problem into a finite-dimensional nonlinear programming (NLP) problem which can be solved using standard methods.

Direct methods can be classified as sequential if only a discretization of the control inputs is performed or simultaneous if both the control inputs and the state variables are discretized. These methods are briefly reviewed in the remainder of this section.

5.1.1 Sequential Approach

Sequential methods, also known as Control Vector Parametrization (CVP) (Sargent and Sullivan, 1978), consider only the control inputs to the system to be optimization variables. The control inputs are discretized in time, and this discretization can be performed in several manners. The most common discretization – especially in the case of MPC – is to consider a control input which is piecewise constant over a fixed time interval. This time interval is usually chosen equal to the sampling time of the controller.

In a sequential approach an initial guess for the control inputs is given to the optimizer. Then the system model is simulated by means of an integrator (ODE solver) by using the given controls as inputs to the system. In the multi-stage approach, the integration of the system has to be performed for each scenario. Then the objective function, the constraints and the derivative information are evaluated, and the conditions for optimality are checked. If they are satisfied, an optimal solution has been found, and if not, the NLP solver provides a new guess for the control inputs. Most of the computation time necessary to solve the optimization problem using a sequential approach is taken by the solution of the sequence of initial value problems that the ODE solver has to solve. In the case of multi-stage NMPC, this effort is multiplied by the number of scenarios that the scenario tree contains.

5.1.2 Simultaneous Approach

Although sequential methods are simple to implement, in the last years the use of simultaneous approaches has been preferred by a number of researchers because it can provide better treatment of path constraints and faster solutions if the right tools are used. Furthermore, simultaneous approaches (multiple shooting and full discretization) have the dichotomy property, i.e., by imposing bounds on the state variables along the time, the unstable modes can be *pinned down* avoiding that a state gets unbounded before the final prediction time. This can happen for the sequential approach when integrating unstable systems. A more detailed description of these concepts can be found in (Biegler, 2010). The disadvantage of simultaneous methods is that they result in larger optimization problems. In the simultaneous approach, both the control inputs and the states are discretized and are used as decision variables. There are two main simultaneous methods: multiple shooting and full discretization.

Multiple Shooting

In multiple shooting, presented in (Bock and Plitt, 1984), the states are also discretized and added as additional optimization variables to the original optimization problem. The integration of the system over the prediction horizon is divided into stages, that usually coincide with the sampling time. In order to enforce continuity of the solution, continuity constraints are added to the optimization problem, which enforce that the final point of the integration within one stage is equal to the initial point of the following stage. For a more detailed description of the multiple-shooting approach the reader is referred to (Bock and Plitt, 1984) or (Diehl et al., 2001). The extension of the multiple-shooting approach to the multi-stage NMPC framework is straightforward. The integration of the system over the prediction horizon is divided into stages, which can coincide with the different branches of the tree, as illustrated in Fig 5.1. From each node $x_k^{p(j)}$ in the tree, the system model is solved using an ODE solver for the corresponding control input and value of the uncertain parameters. At the end of each sampling time, an auxiliary variable $\tilde{s}_{k+1}^j$ is obtained as a result of the integration. This is done for all the branches of the scenario tree. In order to enforce continuity on the solution, new constraints are added so that the resulting optimization problem can be written as:

$$\min_{x_k^j, u_k^j, \tilde{s}_k^j \forall (j,k) \in I} \quad \sum_{i=1}^{N} \omega_i J_i(X_i, U_i) \tag{5.1a}$$

subject to:

$$\begin{aligned}
&\tilde{s}_{k+1}^j = \tilde{F}(x_k^{p(j)}, u_k^j, d_k^{r(j)}), && \forall\, (j, k+1) \in I, && (5.1\text{b})\\
&\tilde{s}_{k+1}^j - x_{k+1}^j = 0 && \forall\, (j,k) \in I, && (5.1\text{c})\\
&g(x_k^{p(j)}, u_k^j) \leq 0\ , && \forall\, (j,k) \in I, && (5.1\text{d})\\
&u_k^j = u_k^l \text{ if } x_k^{p(j)} = x_k^{p(l)}, && \forall\, (j,k), (l,k) \in I, && (5.1\text{e})
\end{aligned}$$

where $\tilde{F} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \rightarrow \mathbb{R}^{n_x}$ denotes the ODE solver operator and (5.1c) are the constraints that enforce the continuity of the solution. All the integrations at each branch of the tree can be performed in parallel because they are completely independent. Since the intermediate states are also optimization variables, the multiple shooting formulation results in a larger optimization problem compared to the sequential approach, but it is

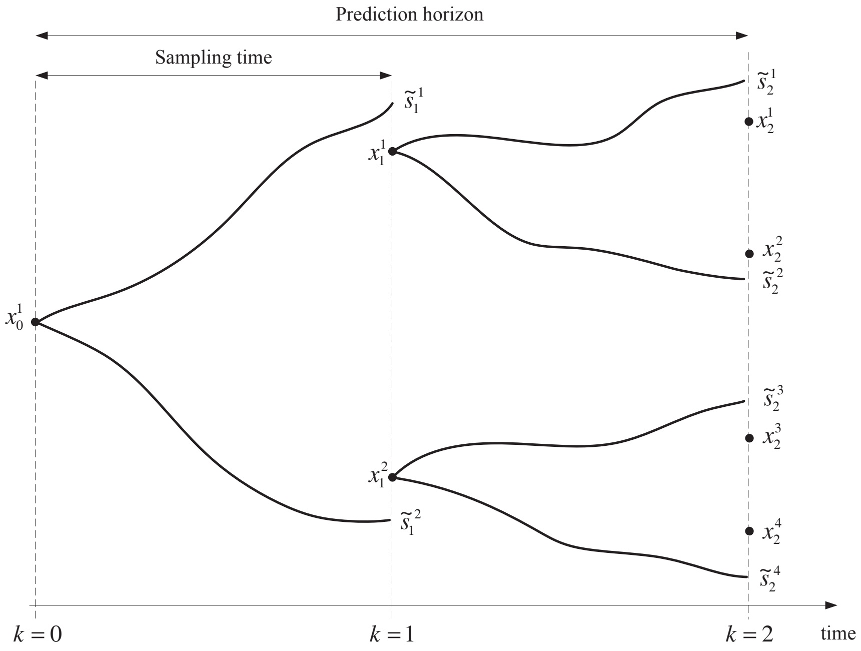

Figure 5.1: Multiple shooting for multi-stage NMPC for a scenario tree with two realizations of the uncertainty and a prediction horizon and robust horizon $N_P = N_R = 2$.

highly structured and sparse, which can be exploited by modern NLP solvers to achieve faster solutions as explained e.g. in (Diehl et al., 2001).

Full Discretization

A further step in the discretization of the dynamics of the system is performed when a full discretization approach is used, which is usually implemented using orthogonal collocation on finite elements. In this method, also the states and control inputs are discretized and added as optimization variables. Unlike in multiple shooting, the dynamics of the system are approximated between the sampling times so that the use of an ODE solver is not necessary. Each control interval is divided into finite elements, on which the state trajectory is parametrized using Lagrange polynomials such that the states are represented by:

$$x^j_{k,\gamma}(t) = \sum_{\theta=0}^{n_\theta} L_\theta(t) x^j_{k,\gamma,\theta}, \tag{5.2}$$

where $x^j_{k,\gamma}(t)$ are the state variables at position (j,k) of the tree, at time t for the finite element γ, and $x^j_{k,\gamma,\theta}$ are the state variables at stage k for the finite element γ at the collocation point θ. The Lagrange polynomials are defined as:

$$L_\theta(\tau) = \prod_{l=0,l\neq\theta}^{n_\theta} \frac{\tau - \tau_l}{\tau_\theta - \tau_l}, \tag{5.3}$$

where $\tau_0, ..., \tau_{n_\theta}$ are the collocation points chosen as the roots of the Gauss-Jacobi polynomials. The values of these coefficients can be found in the literature as for example in (Biegler, 2010). The time $t_{k,\gamma,\theta}$ denotes the time associated with the collocation point θ of the finite element γ at stage k in the scenario tree and $t^{\text{end}}_{k,\gamma}$ is the time at the end of finite element γ at stage k. These times only depend on the choice of the polynomials, number of finite elements n_γ and degree of polynomial n_θ and for this reason they are the same for all the scenarios at the same stage (they do not depend on the uncertainty realization). Note that due to the Lagrangian basis, $x^j_{k,\gamma,n_\theta} = x^j_k(t_{k,\gamma,\theta})$ is satisfied at the collocation points. Fig. 5.2 illustrates the collocation approach applied to multi-stage NMPC with a total number of finite elements per sampling time $n_\gamma = 2$ and for a polynomial of degree $n_\theta = 2$.

By differentiating (5.2) for the states along each finite element $x^j_{k,\gamma}$, the collocation algebraic equations are obtained and added as nonlinear constraints in (5.4b), where

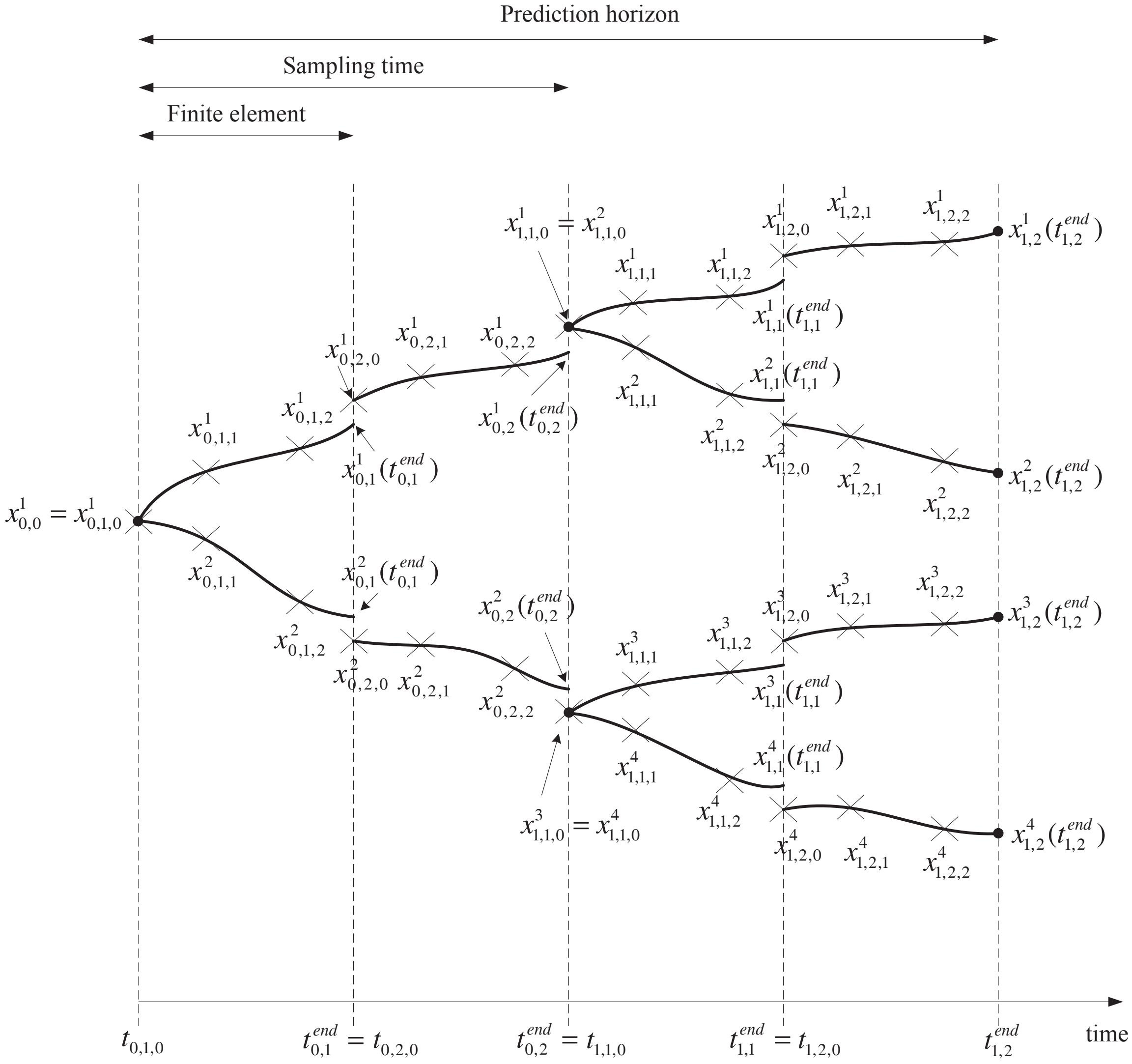
Prediction horizon
Sampling time
Finite element
$x^1_{0,0} = x^1_{0,1,0}$
$x^1_{0,1,1}$
$x^1_{0,1,2}$
$x^1_{0,1}(t^{end}_{0,1})$
$x^1_{0,2,0}$
$x^1_{0,2,1}$
$x^1_{0,2,2}$
$x^1_{0,2}(t^{end}_{0,2})$
$x^1_{1,1,0} = x^2_{1,1,0}$
$x^1_{1,1,1}$
$x^1_{1,1,2}$
$x^1_{1,1}(t^{end}_{1,1})$
$x^1_{1,2,0}$
$x^1_{1,2,1}$
$x^1_{1,2,2}$
$x^1_{1,2}(t^{end}_{1,2})$
$x^2_{1,1,1}$
$x^2_{1,1,2}$
$x^2_{1,1}(t^{end}_{1,1})$
$x^2_{1,2,0}$
$x^2_{1,2,1}$
$x^2_{1,2,2}$
$x^2_{1,2}(t^{end}_{1,2})$
$x^2_{0,1,1}$
$x^2_{0,1,2}$
$x^2_{0,1}(t^{end}_{0,1})$
$x^2_{0,2,0}$
$x^2_{0,2,1}$
$x^2_{0,2,2}$
$x^2_{0,2}(t^{end}_{0,2})$
$x^3_{1,1,0} = x^4_{1,1,0}$
$x^3_{1,1,1}$
$x^3_{1,1,2}$
$x^3_{1,1}(t^{end}_{1,1})$
$x^3_{1,2,0}$
$x^3_{1,2,1}$
$x^3_{1,2,2}$
$x^3_{1,2}(t^{end}_{1,2})$
$x^4_{1,1,1}$
$x^4_{1,1,2}$
$x^4_{1,1}(t^{end}_{1,1})$
$x^4_{1,2,0}$
$x^4_{1,2,1}$
$x^4_{1,2,2}$
$x^4_{1,2}(t^{end}_{1,2})$
$t_{0,1,0}$
$t^{end}_{0,1} = t_{0,2,0}$
$t^{end}_{0,2} = t_{1,1,0}$
$t^{end}_{1,1} = t_{1,2,0}$
$t^{end}_{1,2}$
time

Figure 5.2: Orthogonal collocation on finite elements for multi-stage NMPC with $n_\gamma = 2$ finite elements and polynomials of degree $n_\theta = 2$ for a scenario tree with two realizations of the uncertainty and a prediction horizon and robust horizon $N_P = N_R = 2$.

$\dot{x}^j_{k,\gamma} = \frac{1}{h_\gamma}\sum_{\theta=0}^{n_\theta} \dot{L}_\theta(t_{k,\gamma,\theta})x^j_{k,\gamma,\theta}$, where the coefficients $\dot{L}_\theta(t_{k,\gamma,\theta})$ can be calculated beforehand based on the choice of the collocation points. The length of the finite element γ is denoted by h_γ. The continuity equations at the end of each finite element (5.4c) and at the end of each control interval (5.4d) are also added as constraints. The constraints of the original dynamic optimization problem are checked for all the collocation points as can be seen in (5.4e). Thus the optimization problem for multi-stage NMPC with orthogonal collocation on finite elements can be written as:

$$\min_{x^j_{k,\gamma}, u^j_k, x^j_{k,\gamma,\theta} \forall (j,k)\in I,} \quad \sum_{i=1}^{N} \omega_i J_i(X_i, U_i) \tag{5.4a}$$

subject to:

$$\dot{x}^j_{k,\gamma} = f(x^j_{k,\gamma,\theta}, u^j_k, d^{r(j)}_k), \qquad \forall\,(j,k+1)\in I, \gamma\in\{1,...,n_\gamma\}, \theta\in\{0,...,n_\theta\} \tag{5.4b}$$

$$x^j_{k,\gamma}(t^{\text{end}}_{k,\gamma}) = x^j_{k,\gamma}(t_{k,\gamma+1,0}) \qquad \forall\,(j,k+1)\in I, \gamma\in\{1,...,n_\gamma-1\}, \tag{5.4c}$$

$$x^{p(j)}_{k,\gamma}(t^{\text{end}}_{k,n_\gamma}) = x^j_{k+1,\gamma}(t_{k+1,1,0}) \qquad \forall\,(j,k+1)\in I, \tag{5.4d}$$

$$g(x^j_{k,\gamma,\theta}, u^j_k) \le 0\ , \qquad \forall\,(j,k)\in I, \gamma\in\{1,...,n_\gamma\}, \theta\in\{0,...,n_\theta\} \tag{5.4e}$$

$$u^j_k = u^l_k \text{ if } x^{p(j)}_{k,0} = x^{p(l)}_{k,0}, \qquad \forall\,(j,k),(l,k)\in I, \tag{5.4f}$$

A more detailed description of orthogonal collocation on finite elements for a general setting can be found in the book (Biegler, 2010).

A simple collocation approach consists in applying an implicit Euler discretization in which the next node in the tree is calculated as:

$$x^j_{k+1} = x^{p(j)}_k + t_{\text{step}}\,(f(x^j_{k+1}, u^j_k, d^{r(j)}_k)), \tag{5.5}$$

where t_{step} is the sampling time. This discretization is very easy to implement, but it requires small sampling times t_{step} to obtain a good accuracy in the discretization.

In this thesis, only simultaneous methods are used. The optimization problems for the different examples discussed in the following chapters are kept in the general form presented in (3.5) for simplicity.

5.2 Calculation of the Derivatives

Regardless of the methods used for the numerical solution of the resulting multi-stage NMPC problem, an NLP solver is necessary to calculate the optimal control inputs. Modern NLP solvers apply different algorithms, such as Sequential Quadratic Programming

(SQP) or Interior Point (IP) methods, which need derivative information of both, the constraints and the cost function. The efficiency and accuracy of the calculation of the derivative information has a crucial influence on the performance of the optimizer. For this reason the computation of derivatives is a topic which has received great attention in the literature in the last years.

There are three main approaches to calculate derivatives: finite differences, symbolic differentiation and automatic differentiation.

Calculation by Finite Differences

The finite differences method is based on the application of Taylor's theorem to compute the derivative of a smooth function f with respect to the variable x, which in the scalar case can be written as:

$$\frac{\mathrm{d}f}{\mathrm{d}x} \approx \frac{f(x+h) - f(x)}{h}, \tag{5.6}$$

where h is a small number that has to be chosen. If h is too small, numerical (round-off) errors occur and if h is too large, the first order Taylor approximation will not be accurate resulting in a poor approximation of the derivative. These problems become more important when higher order derivatives are needed. The computation of the necessary gradients, Jacobians and Hessians can be done in a similar manner as explained in (Nocedal and Wright, 2006).

Symbolic differentiation

A symbolic calculation of the derivatives makes use of computer algebra tools that are able to recursively manipulate an algebraic expression. They result in exact computation of the derivatives up to machine precision, but for large expressions, they usually result in very long expressions that are expensive to evaluate. In addition, the code has to be provided in form of functions, so that the derivatives of algorithms cannot be provided. In contrast, these can be obtained by using automatic differentiation.

Automatic Differentiation

Automatic differentiation, also called algorithmic differentiation, is a technique for the evaluation of derivatives of functions that are represented in a computer. The main idea of automatic differentiation is the exploitation of the fact that any differentiable function can be represented as a combination of elementary operations such as addition, multiplication, division, trigonometric and exponential functions among others. Using this decomposition and applying the chain rule from elementary calculus, automatic differentiation provides derivatives that are exact up to machine precision with a computational cost that is of the same order of magnitude than the cost of evaluating the original function. Automatic differentiation is different from symbolic differentiation, mainly because the symbolic expression of the derivative is never generated but only evaluated which can result in a much faster calculation of the derivatives.

In order to illustrate the basic ideas of automatic differentiation, a simple example adapted from (Griewank and Walther, 2008) and (Nocedal and Wright, 2006) is used.

Consider the function:

$$f(x) = x_1 x_2 e^{x_1} + \cos(x_1/x_2). \tag{5.7}$$

The evaluation of this function can be decomposed into 6 elementary operations as follows:

$$v_1 = x_1 x_2, \tag{5.8a}$$

$$v_2 = e^{x_1}, \tag{5.8b}$$

$$v_3 = x_1/x_2, \tag{5.8c}$$

$$v_4 = v_1 v_2, \tag{5.8d}$$

$$v_5 = \cos(v_3), \tag{5.8e}$$

$$v_6 = v_4 + v_5. \tag{5.8f}$$

Usually, in the framework of automatic differentiation the sequence of expressions for the evaluation of the function is substituted by a computational graph that defines the function f as shown in Fig. 5.3. This also helps to have a better representation of functions that contain nested loops or other procedures, which would result in a huge amount of lines of elementary operations as the ones shown above. In automatic differentiation, the elementary operations are differentiated using the chain rule. There are two different approaches (also called modes) for automatic differentiation which are explained in the remainder of this section.

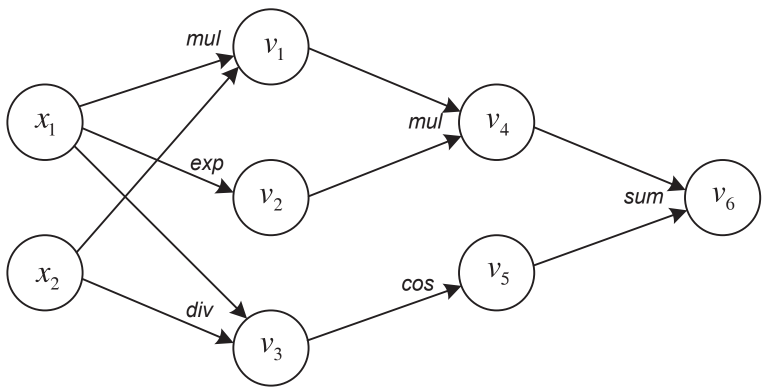

Figure 5.3: Computational graph for the function $f(x) = x_1 x_2 e^{x_1} + \cos(x_1/x_2)$.

Forward Mode

In the forward mode, new variables $\dot{v}_j = \frac{\partial v_j}{\partial x_i}$ are associated with each intermediate variable v_j in order to get the derivative of the function f with respect to one of the independent variables x_i. There are N independent variables $(x_1, ..., x_N)$ and M intermediate variables $(v_1, ..., v_M)$. The new variables can be calculated using the chain rule as:

$$\dot{v}_i = \frac{dv_i}{dx} = \sum_{j=1}^{N} \frac{\partial v_i}{\partial x_j} \frac{dx_j}{dx}. \tag{5.9}$$

Given $[x_1, x_2]^{\mathrm{T}}$ and $[\dot{x}_1, \dot{x}_2]^{\mathrm{T}}$, it is possible to calculate:

$$v_1 = x_1 x_2, \qquad \dot{v}_1 = \dot{x}_1 x_2 + x_1 \dot{x}_2, \tag{5.10a}$$

$$v_2 = e^{x_1}, \qquad \dot{v}_2 = e^{x_1} \dot{x}_1, \tag{5.10b}$$

$$v_3 = x_1/x_2, \qquad \dot{v}_3 = \frac{\dot{x}_1 x_2 - x_1 \dot{x}_2}{x_2^2}, \tag{5.10c}$$

$$v_4 = v_1 v_2, \qquad \dot{v}_4 = \dot{v}_1 v_2 + v_1 \dot{v}_2, \tag{5.10d}$$

$$v_5 = \cos(v_3), \qquad \dot{v}_5 = -\sin(v_3)\dot{v}_3, \tag{5.10e}$$

$$v_6 = v_3 + v_5, \qquad \dot{v}_6 = \dot{v}_3 + \dot{v}_5. \tag{5.10f}$$

The new associated intermediate variable $\dot{v}_6$ gives a directional derivative $\dot{v}_6 = \nabla f^T [\dot{x}_1, \dot{x}_2]^{\mathrm{T}}$. The vector $p = [\dot{x}_1, \dot{x}_2]^{\mathrm{T}}$ is referred to as *seed vector*. In order to calculate the derivative of f with respect to the variable x_1, the operations shown above have to be performed by choosing $[\dot{x}_1, \dot{x}_2]^{\mathrm{T}} = [1, 0]^{\mathrm{T}}$. In order to calculate the full gradient of f, the algorithm has to be called N times (in this case $N = 2$), with different seed vectors $([1, 0]^T, [0, 1]^T)$.

This technique can be extended to the computation of Jacobians in a straight forward manner. As explained in the automatic differentiation literature ((Griewank and Walther, 2008), (Nocedal and Wright, 2006) or in (Andersson, 2013)), if the Jacobians are sparse, a more efficient choice of the seed vectors can be made using graph coloring techniques to get the full Jacobians in a computationally cheaper way.

Reverse Mode

In the reverse mode – also called adjoint mode – instead of calculating the derivatives of each intermediate variable, the derivatives of the function (output variable) with respect to the intermediate variables are calculated. For this purpose, a new set of variables, called the adjoint variables $\bar{v}_i = \frac{\partial f}{\partial v_i}$ are associated with each variable v_i. All of them are initialized with the value 0, except for the one associated with the last intermediate variable, which is set to one ($v_L = 1$). Then, using the chain rule, the partial derivatives of the function f with respect to each one of the intermediate variables can be written as

$$\bar{v}_i = \frac{df}{dv_i} = \sum_{j \in I(i)} \frac{df}{dv_j} \frac{\partial v_j}{\partial v_i}, \tag{5.11}$$

where $I(i)$ contains the indices of the children nodes of v_i. The same definition holds for the adjoint variables $\bar{x}_i$ associated to the input variables x_i. In a computer program, as soon as all the quantities necessary to compute one of these operations are available, the corresponding operation is performed and updated in the variable. The information about all the children might not be available at the same time at which the variables are updated though. For this reason, the additional variables are initialized to $v_i = 0, \forall i = 1, ..., 5$ and $\bar{v}_6 = 1$ because it is the last node and it has no children, so that $\bar{v}_6 = \frac{df}{dv_i} = 1$. Thus for a given $[x_1, x_2]^{\mathrm{T}}$, the operations necessary to compute the gradient of f in the reverse mode are:

$$v_1 = x_1 x_2, \tag{5.12a}$$

$$v_2 = e^{x_1}, \tag{5.12b}$$

$$v_3 = x_1 / x_2, \tag{5.12c}$$

$$v_4 = v_1 v_2, \tag{5.12d}$$

$$v_5 = \cos(v_3), \tag{5.12e}$$

$$v_6 = v_4 + v_5, \tag{5.12f}$$

Now the graph in Fig. 5.3 is evaluated backwards

$$\bar{v}_6 = \frac{\partial f}{v_6} = 1 \text{ (given initialization)} \tag{5.12g}$$

$$\bar{v}_5 + = \frac{df}{dv_6}\frac{\partial v_6}{\partial v_5} = \bar{v}_6 1 \tag{5.12h}$$

$$\bar{v}_4 + = \frac{df}{dv_6}\frac{\partial v_6}{\partial v_4} = \bar{v}_6 1 \tag{5.12i}$$

$$\bar{v}_3 + = \frac{df}{dv_5}\frac{\partial v_5}{\partial v_3} = \bar{v}_5(-\sin(v_3)) \tag{5.12j}$$

$$\bar{v}_2 + = \frac{df}{dv_4}\frac{\partial v_4}{\partial v_2} = \bar{v}_3 v_1 \tag{5.12k}$$

$$\bar{v}_1 + = \frac{df}{dv_4}\frac{\partial v_4}{\partial v_1} = \bar{v}_3 v_2 \tag{5.12l}$$

$$\bar{x}_1 + = \frac{df}{dv_1}\frac{\partial v_1}{\partial x_1} = \bar{v}_1 x_2 \tag{5.12m}$$

$$\bar{x}_1 + = \frac{df}{dv_2}\frac{\partial v_2}{\partial x_1} = \bar{v}_2 e^{x_1} \tag{5.12n}$$

$$\bar{x}_1 + = \frac{df}{dv_3}\frac{\partial v_3}{\partial x_1} = \bar{v}_3 \frac{1}{x_2} \tag{5.12o}$$

$$\bar{x}_2 + = \frac{df}{dv_1}\frac{\partial v_1}{\partial x_2} = \bar{v}_1 x_1 \tag{5.12p}$$

$$\bar{x}_2 + = \frac{df}{dv_3}\frac{\partial v_3}{\partial x_2} = \bar{v}_3 \frac{-x_1}{x_2^2}. \tag{5.12q}$$

The final values accumulated in the variables $\bar{x}_1, \bar{x}_2$ constitute the full gradient of the function f:

$$\nabla f(x) = [\bar{x}_1, \bar{x}_2]^{\mathrm{T}}. \tag{5.13}$$

The main advantage of the reverse mode with respect to the forward mode is that the algorithm has to be executed only once to calculate the full gradient of a scalar function. As discussed in (Nocedal and Wright, 2006), the computation effort needed to evaluate the gradient ∇f is at most five times the time needed to evaluate the function f, no matter how many input variables x_i exist. The disadvantage is that, unlike in the forward mode, the full computational graph has to be stored for its backward evaluation.

5.3 Implementation of Multi-stage NMPC

Unless explicitly mentioned, the dynamics of the system are discretized using a simultaneous approach for all the results presented in this thesis. Then the first order and second order derivatives of the cost function and the constraints are calculated using automatic differentiation. For the rest of the thesis, it is assumed that the models and constraints under consideration are differentiable. All this information is given to an NLP solver which solves the optimization problem at each sampling instance.

The algorithms presented in this thesis have been implemented using the optimization tool CasADi (Andersson et al., 2012a), which makes it possible to greatly improve the computational performance compared to a simple implementation using Matlab. An illustrative comparison will be shown in Part III of this thesis.

CasADi is an open-source framework for C++ and Python for numerical optimization and optimal control. The main feature of CasADi is that it provides the users with a flexible framework to implement a wide range of optimal control algorithms in an easy and efficient way, rather than providing the user with a black-box Optimal Control Problem (OCP) solver. A nonlinear programming problem in CasADi can be represented in a general high-level symbolic representation which internally is represented as an expression graph which may contain also operations such as ODE/DAE integrators. After formulating the NLP, it can be solved with any available solver or by a self-written NLP solver. The solver used relies on CasADi to calculate first and second order derivative information automatically. The generation of the Jacobian of the nonlinear constraints and of the Hessian of the Lagrangian function is done via a state-of-the-art implementation of automatic differentiation as presented in (Griewank and Walther, 2008). If ODE or DAE integrators are part of the symbolic expressions, forward and backward sensitivity analysis is invoked automatically, see (Cao et al., 2003) for more details. CasADi relieves the user from the implementation effort and from the possible errors that are common when calculating and passing derivative information. Furthermore, it offers a convenient working environment in the high-level language Python (see (Andersson et al., 2012b) for a simple example). This makes it possible to significantly reduce the necessary effort to implement an optimal control solver. The results presented in (Andersson et al., 2012a) suggest that CasADi is faster than other state-of-the-art software such as the AMPL Solver Library (ASL) when solving standard benchmark problems. In this work the resulting nonlinear programming optimization problems are solved using IPOPT (Wächter and Biegler, 2006) which uses first and second order exact derivative information provided automatically by CasADi. As described before, CasADi makes it possible to solve the optimal control problem using direct multiple shooting or direct collocation for the discretization of the ODEs. In this thesis, Radau collocation points are used for the collocation approach, using interpolating polynomials of degree 2. For the multiple shooting approach the integrators from the SUNDIALS toolbox (Hindmarsh et al., 2005) are used.

The real plant is simulated with the calculated control input using also the integrators from the SUNDIALS toolbox with a high accuracy. All the optimization problems are solved on a standard laptop with an Intel i-5 processor at 2.30GHz running Ubuntu on a virtual machine with one core and 1 GB of RAM.

The implementation of the multi-stage NMPC approach using CasADi has been done in collaboration with Joel Andersson and Prof. Moritz Diehl from KU Leuven within the framework of the European Project EMBOCON.

Chapter 6

DO-MPC: An Environment for the Easy Development of NMPC Solutions

Linear model predictive control is a control strategy that has been successfully applied at many industrial plants as reported in (Qin and Badgwell, 2003). For nonlinear model predictive control, many simulation results have been published in the last years, including large-scale and highly nonlinear systems based on rigorous models (e.g. (Huang et al., 2009b) or (Idris and Engell, 2012)). Although some companies develop industrial NMPC implementations for some processes (see (IPCOS, 2014), (Pluymers et al., 2008), or (Cybernetica, 2014)) its practical use is still in its early stages and not a widespread reality.

A decade ago, one of the main reasons for this gap between academic research and industrial practice was the long computation time that was necessary to solve the resulting complex nonlinear programming problems. In the last years the progress on algorithms and computation power has made it possible to significantly reduce the computation times needed to solve NMPC problems even to the microsecond range (Houska et al., 2011a), if appropriate tools are used. Most of these tools have recently been developed in the academia such as MUSCOD II (Diehl et al., 2001), ACADO (Houska et al., 2011b), NMPC tools (Amrit and Rawlings, 2008), OptCon (Nagy, 2008) or the MPT Toolbox (Herceg et al., 2013) among others. These tools can solve different kind of problems including nonlinear model predictive control formulations.

However, one of the main obstacles for the knowledge transfer from academia to industry via these tools is their lack of modularity and the lack of sustainability of the implementations. Most of the tools named above require an implementation of the model in a particular syntax, and an interface of the tool with other necessary components such as a simulator or an observer. If the model needs to be changed, all the codes have to be adjusted accordingly. The lack of modularity also makes it difficult to compare the computational performance and solution quality of different approaches or tools (or to combine parts of different tools) because they require a completely new implementation, which will result in an obscure and time-consuming comparison.

In order to deal with this problem, this chapter presents a new concept for the modularization of an NMPC implementation, dividing it into four main components: Model, optimizer, observer and simulator. This is realized within the environment DO-MPC, which includes a modular implementation of an NMPC approach to which any other software available can be coupled by means of rewriting a standardized interface that represents the needs of the specific problem. Automatic plotting and logging of the data as well as the implementation of multi-stage NMPC are also included. The presented environment attempts to overcome two of the main problems of real NMPC implementations. Firstly, thanks to the modularity the implementations are simple and reusable. Secondly, the use of multi-stage NMPC makes it possible to take into account possible plant model mismatches. The tool is implemented in C using CasADi as a building block in order to achieve very good performance with a low implementation effort.

The work presented in this chapter is based on previous work done by Christian Schoppmeyer in the European Project EMBOCON, and an important part was done in the framework of the master thesis of Alexandru Tatulea (Tatulea-Codrean, 2014). This joint work led to the publication (Lucia et al., 2014e).

6.1 A Modular NMPC Development Environment

One of the main outcomes of the European research project EMBOCON (Embedded Optimization for Resource Constrained Platforms) was the development of the open source software platform GEMS (Generic EMBOCON Minimal Supervisor) (Schoppmeyer, 2013).

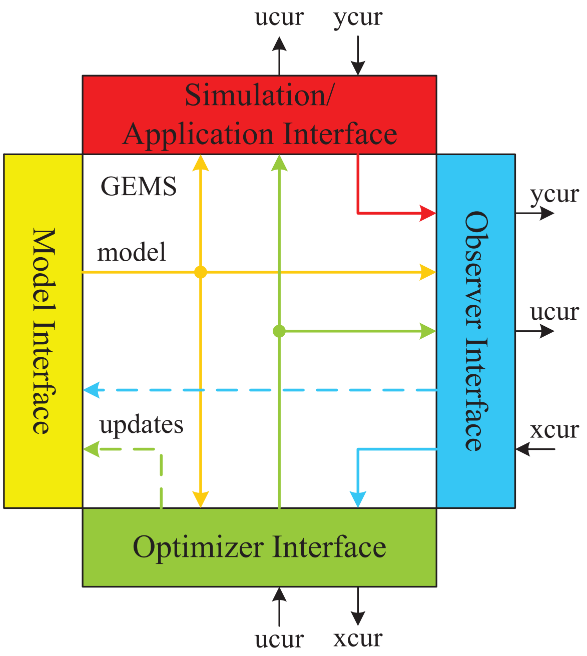

Figure 6.1: Modular scheme of an NMPC implementation with four main blocks: Model, optimizer, observer and simulator.

The central idea of GEMS is to offer a set of general and standarized interfaces to simplify the process of developing and deploying a model-based control algorithm to a real system, and to offer a so-called supervisor that manages the flow of information between the different parts of the implementation. For this purpose, the implementation of a model-based control approach is divided into four main components: the model, the optimizer, the observer and the simulation or real application (see Fig. 6.1). The exchange of information between the different modules is managed and logged by a supervisor. Using this conceptual idea, existing or newly developed algorithms for control, for simulation or for state estimation of a system can be implemented based on the GEMS interfaces, which are programmed in plain C-code to facilitate the use of existing tools and the extension of them. The only necessary step is to provide the information required by GEMS in the template form of its interfaces. The information exchanged between the different modules includes the current control inputs, states and outputs (denoted as ucur, xcur, and ycur in Fig. 6.1). In addition, the information about the right hand sides of the ODEs or the derivatives can be also exchanged an updated if e.g. parameter estimation is available. A complete definition of the interfaces can be found in (Schoppmeyer, 2013).

The model module should contain the ODEs or DAEs defining the system, and also the description of the control task (objective and constraints). The optimizer module should contain the discretization method to transform the optimal control problem to a nonlinear programming problen and a suitable solver. The observer module should include a mapping from the measurements to the states (and also possible parameter estimation) and finally the simulator or application module should include an integrator to simulate the system or an interface to an input-ouput device to control a real plant. With the modular implementation, when the models are updated or changed, there is no need to update the code in each one of the different implementation parts, because they will automatically get the information about the new model via the interfaces. This enhances the sustainability of an NMPC implementation. Note that if different models need to be used for the different modules, e.g. in the simulator a complex model is used but a simplified one in the controller, this can also be taken into account by defining several model modules for the same problem.

All the different modules are implemented as shared libraries within GEMS. They have to be compiled before hand so that the supervisor can load the corresponding library for each module at run-time. This ensures portability, transparency and configurability of the final NMPC implementation. The implementation is done in the Linux OS which makes the real-time managing of the modules possible. GEMS is freely available and it can be downloaded from (Schoppmeyer, 2013).

6.2 DO-MPC: An Environment for an Easy, Modular, Robust and Efficient Development of NMPC

The desired features of an NMPC development platform described before have been implemented in DO-MPC (TU DOrtmund MPC). DO-MPC is an environment that uses the main ideas of GEMS to provide users with an easy, modular, robust and efficient way to realize sustainable implementations of NMPC.

The main added value of DO-MPC with respect to other available MPC tools is that instead of offering a black-box solution to the MPC problem, it provides a platform to develop own NMPC realizations with a very low effort. We use CasADi (Andersson et al., 2012a) as a building block to develop the necessary modules and interfaces. Furthermore, we provide templates using the scripting language Python for each one of the modules including an implementation of the multi-stage NMPC approach. These templates provide a user-friendly environment for the formulation of a new NMPC problem that only requires the definition of the model equations, the control problem and the estimation problem. However, it is not necessary to use the provided templates and the user can couple any existing software just by writing an interface to DO-MPC.

Once the information for the different modules is provided using the templates, CasADi automatically generates C-code. The autogenerated C-code is compiled to shared libraries, which are loaded at run-time by DO-MPC. This ensures that the exchange of the different modules can be done by just loading a different shared library at run-time.

Fig. 6.2 shows an example of a configuration of DO-MPC. This includes the choice of the 4 modules (model, optimizer, observer and simulator) which uniquely determine an NMPC implementation. In the example the multi-stage NMPC with CasADi/IPOPT is used as optimizer, an Extended Kalman Filter as observer and the SUNDIALS (Hindmarsh et al., 2005) integrators as simulator.

Along with the simplicity of the implementation, one of the main features of DO-MPC is its modular implementation that makes it possible to use different configurations in parallel. Different optimizers, cost functions, or observers can be tested online, while maintaining the rest of the modules. This can be very useful for monitoring or comparison purposes. Additionally, a degree of redundancy can be introduced to cope with possible failures of some of the algorithms. For example, different optimization algorithms can be run in parallel to cope with possible convergence problems of the optimizer. An example of the structure of DO-MPC can be seen in Fig. 6.3. DO-MPC also features an automatic data visualization tool which shows any variable previously defined, along with the predictions that the NMPC controller is computing to enhances the understanding of the controller performance. Using several configurations in parallel makes it possible to switch to another configuration when a problem is detected in one of them, avoiding the injection of wrong control to the plant.

The monitoring of the performance of an NMPC controller is an important subject of research and the DO-MPC environment facilitates its use, while already including techniques based on multivariate statistics (AlGhazzawi and Lennox, 2009). Other techniques presented recently in the literature will also be implemented in the future (Patwardhan et al., 2002), (Zagrobelny et al., 2013). The structure of the tool and a minimal graphical

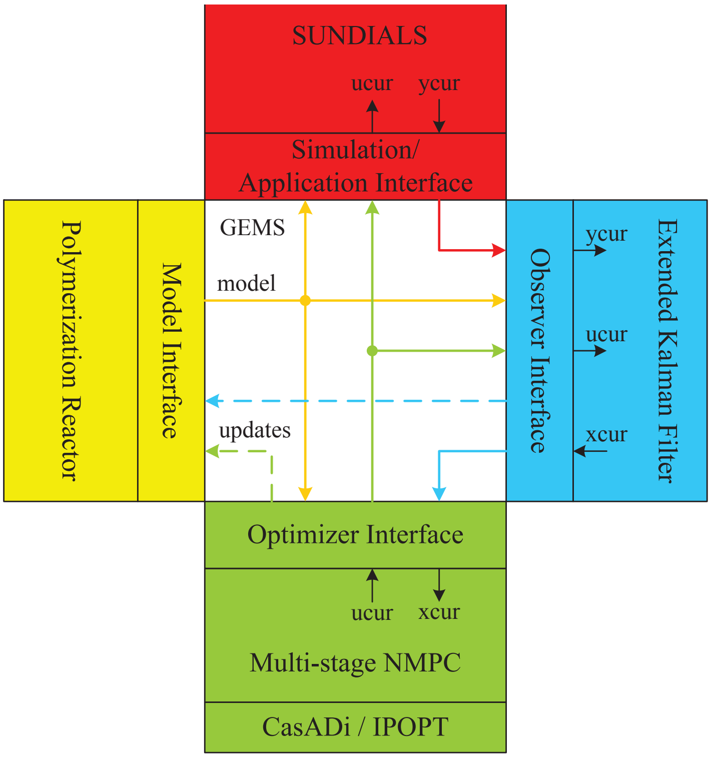

Figure 6.2: Example of a DO-MPC configuration.

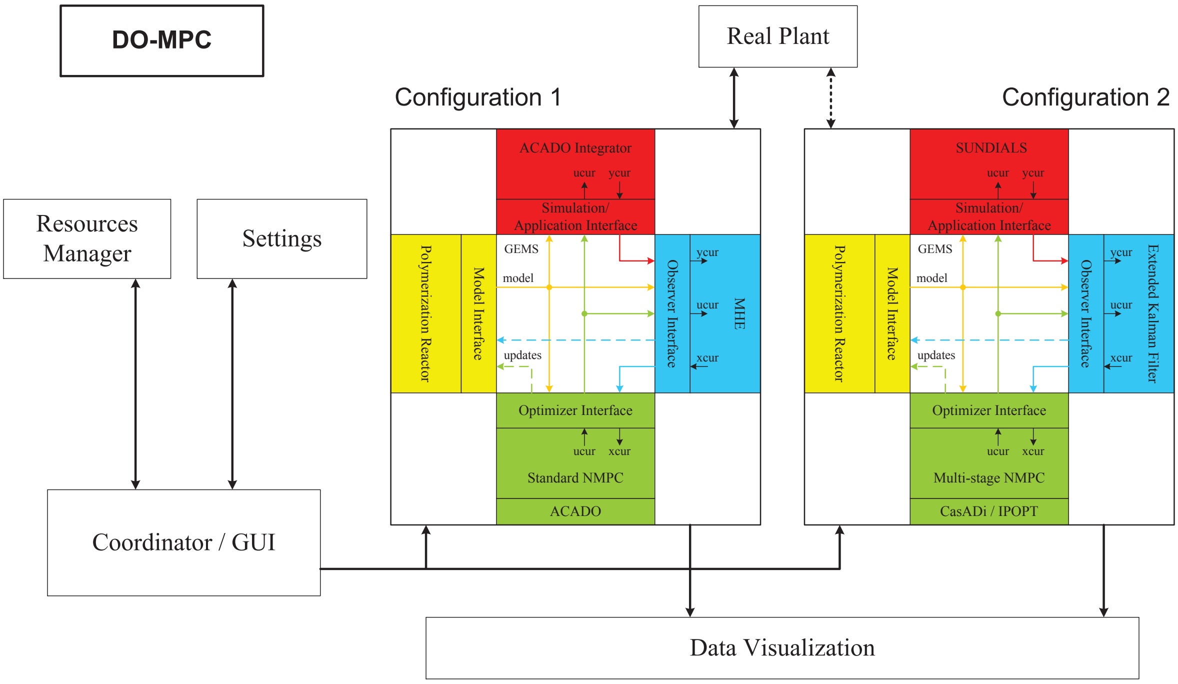

Figure 6.3: Example of an NMPC implementation with DO-MPC with two configurations running in parallel.

user interface has been developed using the Qt5.1 framework for C++ (Qt5.1, 2014). In order to ensure the real-time capabilities of the tool, the implementation is done in a multithreaded way. This ensures that the different computationally demanding tasks (optimization, plotting, monitoring) are deployed on concurrent threads so that their execution is performed without interruptions. Together with the use of the auto-generated efficient C-code in the form of shared libraries for each one of the modules, this leads to a very efficient implementation which at the same time is transparent, modular and sustainable. A simplified version of the DO-MPC tool can be obtained from (Lucia et al., 2015).

This chapter presented a generic environment for nonlinear model-predictive control that supports the development and testing of NMPC solutions. The tool can be used to achieve an easy, but still very efficient implementation of an NMPC controller. The modular implementation makes the tool suitable as a framework to compare in a transparent manner different MPC algorithms, solvers or estimators. It is also easy to use the tool for the implementation of the algorithms in a real plant by just implementing a simple interface between the available data acquisition system and the simulator interface of DO-MPC. Then the algorithms can be tested in the real plant by exchanging the simulator module by the module with the interface to the real plant as it will be shown in Chapter 10.

DO-MPC has been developed specifically for receding horizon implementations and it is therefore not suitable for the development of other optimal control strategies in which the information flow between the different modules (optimizer, simulator, observer) differs significantly from the flow of an MPC implementation. DO-MPC requires that the user installs the software that he or she wants to use (CasADi, IPOPT, SUNDIALS,...). The self-contained generation of code which does not make use of external libraries is not supported.

Chapter 7

Solution via Scenario Decomposition Techniques

As it was shown in the previous chapters, the formulation of the multi-stage NMPC approach results in large optimization problems, which have to be solved in a defined amount of time if the algorithm is going to be applied to a real system. There are different possibilities to achieve a fast solution of the resulting optimization problems. One is to use a collocation approach with exact first and second order derivatives calculated using automatic differentiation, and solving the optimization problem with solvers that can exploit sparsity, as explained in Chapter 5.

It is also possible to exploit the tree structure of the multi-stage optimization problems at the linear algebra level using methods based on the Schur complement as shown in (Steinbach, 2000), which also makes the parallelization of most of the calculations possible. However, this techniques require a complete implementation of a new tailored NLP solver. This chapter analyzes the possibilities of decomposition algorithms that can be implemented in a straightforward manner using general-purpose NLP solvers such as IPOPT, and it is therefore not a complete review of decomposition algorithms.

In particular, scenario decomposition approaches (see (Birge, 1997) for a review) such as the Progressive Hedging Algorithm (PHA) presented in (Rockafellar and Wets, 1991) are studied. The main idea of this approach is to take advantage of the fact that each scenario in the scenario tree is an independent problem except for the non-anticipativity constraints that link the scenarios by forcing all control inputs branching at the same node to be the equal. This algorithm relaxes the non-anticipativity constraints by adding to the cost function of the scenario a term which penalizes their violation and then solves each scenario independently. The algorithm iterates until the non-anticipativity constraints are fulfilled and a feasible solution for the original problem is found.

The approach is shortly described in the rest of this chapter. Most of the work reported in this chapter has been carried out in the framework of the master thesis by (Subramanian, 2012).

7.1 Decomposition Approaches for Multi-stage Optimization

Decomposition of large-scale optimization problems has already been used for a long time (see (Birge, 1997) or (Ruszczynski, 1997) for a review). The monolithic problem is divided into smaller subproblems which are solved independently. For the case of multi-stage stochastic optimization problems, the different approaches are usually classified into scenario decomposition or stage decomposition. In the scenario decomposition approaches each smaller subproblem corresponds to a scenario of the tree and in the stage decomposition each subproblem corresponds to a time stage of the tree. In this work, the scenario decomposition approach is chosen (see Fig. 7.1) because the the robust horizon concept helps to reduce the amount of coupling between the scenarios. Therefore, a scenario decomposition approach is more suitable, compared to the stage-decomposition (Benders decomposition) approach applied for linear MPC in (Muñoz de la Peña et al., 2005a). A dual scenario decomposition approach has been recently proposed in (Maestre et al., 2012) to a linear example, but this chapter is focused on the nonlinear case using a different algorithm.

7.1.1 Scenario Decomposition

The decomposition algorithm used in this thesis is the progressive hedging algorithm of (Rockafellar and Wets, 1991). It can be applied to nonlinear non-convex problems and it has been also proven in (Rockafellar and Wets, 1991) that the solution of the algorithm always converges to a local optimum of the original cost function, if conver-

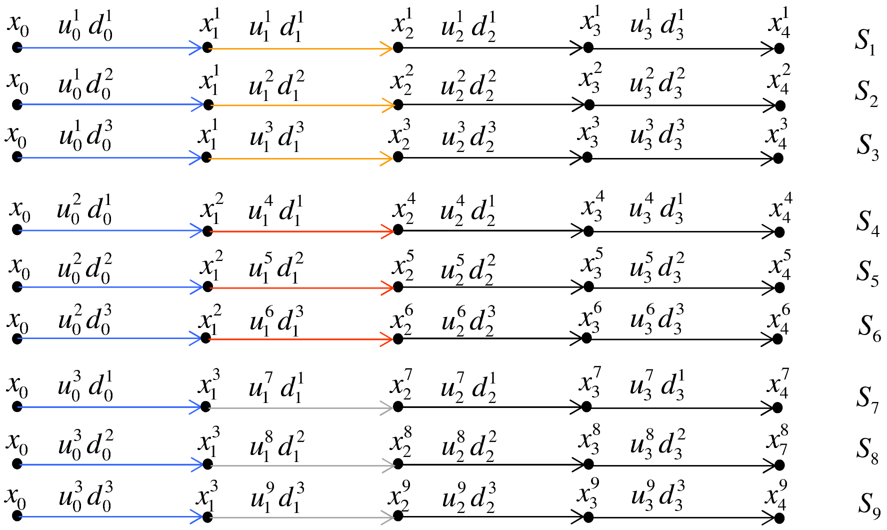
x_0 $u_0^1\, d_0^1$ x_1^1 $u_1^1\, d_1^1$ x_2^1 $u_2^1\, d_2^1$ x_3^1 $u_3^1\, d_3^1$ x_4^1 S_1
x_0 $u_0^1\, d_0^2$ x_1^1 $u_1^2\, d_1^2$ x_2^2 $u_2^2\, d_2^2$ x_3^2 $u_3^2\, d_3^2$ x_4^2 S_2
x_0 $u_0^1\, d_0^3$ x_1^1 $u_1^3\, d_1^3$ x_2^3 $u_2^3\, d_2^3$ x_3^3 $u_3^3\, d_3^3$ x_4^3 S_3
x_0 $u_0^2\, d_0^1$ x_1^2 $u_1^4\, d_1^1$ x_2^4 $u_2^4\, d_2^1$ x_3^4 $u_3^4\, d_3^1$ x_4^4 S_4
x_0 $u_0^2\, d_0^2$ x_1^2 $u_1^5\, d_1^2$ x_2^5 $u_2^5\, d_2^2$ x_3^5 $u_3^5\, d_3^2$ x_4^5 S_5
x_0 $u_0^2\, d_0^3$ x_1^2 $u_1^6\, d_1^3$ x_2^6 $u_2^6\, d_2^3$ x_3^6 $u_3^6\, d_3^3$ x_4^6 S_6
x_0 $u_0^3\, d_0^1$ x_1^3 $u_1^7\, d_1^1$ x_2^7 $u_2^7\, d_2^1$ x_3^7 $u_3^7\, d_3^1$ x_4^7 S_7
x_0 $u_0^3\, d_0^2$ x_1^3 $u_1^8\, d_1^2$ x_2^8 $u_2^8\, d_2^2$ x_3^8 $u_3^8\, d_3^2$ x_7^8 S_8
x_0 $u_0^3\, d_0^3$ x_1^3 $u_1^9\, d_1^3$ x_2^9 $u_2^9\, d_2^3$ x_3^9 $u_3^9\, d_3^3$ x_4^9 S_9

Figure 7.1: Scenario tree representation of the uncertainty evolution using scenario decomposition for multi-stage NMPC with robust horizon $N_R = 2$ and prediction horizon $N_P = 4$. The arrows with the same colors represent the non-anticipativity constraints.

gence is achieved. The main idea of the algorithm is to relax the non-anticipativity constraints by penalizing the difference between the control inputs that should satisfy the non-anticipativity constraints and a fictitious common control input. In this manner, the scenarios can be solved as independent subproblems. Then, the procedure is repeated until the non-anticipativity constraints are fulfilled. Thus, instead of solving the original monolithic problem defined in (3.5) the progressive hedging algorithm solves an independent problem for each scenario i:

$$\min_{x_k^j, u_k^j \forall (j,k) \in I_i} \quad \omega_i J_i(X_i, U_i) + \sum_{k=0}^{N_P - 1} \lambda_k^j (u_k^j - \hat{u}_k^j) + \rho_k^j ||u_k^j - \hat{u}_k^j||^2 \tag{7.1a}$$

subject to:

$$x_{k+1}^j = f(x_k^{p(j)}, u_k^j, d_k^{r(j)}), \qquad \forall\, (j, k+1) \in I_i, \tag{7.1b}$$

$$g(x_{k+1}^j, u_k^j) \leq 0\ , \qquad \forall\ (j, k+1) \in I_i, \tag{7.1c}$$

$$\tag{7.1d}$$

where I_i is the set of indices in the original tree of the nodes corresponding to scenario S_i, that is $I_i = \{(j,k) : x_j^k \in S_i\}$. $\hat{u}_k^j \in \mathbb{R}^{n_u}$ is the fictitious value towards which the control input at stage k should converge for each of the scenarios in which the non-anticipativity constraints have to be satisfied. If a robust horizon $N_R \leq N_P$ is used, the non-anticipativity constraints are applied only until $k = N_R$ because it is assumed that afterwards the uncertainty remains constant and therefore the control inputs can be adapted independently and then the scenarios are independent. For this reason $\hat{u}_k^j$ is chosen such that $\hat{u}_k^j = u_k^j$ for $k \geq N_R$ so that the additional terms in the cost function (7.1a) vanish if there are no non-anticipativity constraints. The choice of $\hat{u}_k^j$ at each iteration greatly influences the performance of the algorithm. A simple approach to obtain $\hat{u}_k^j$ is to use the average value of the control inputs that have to satisfy each non-anticipativity constraint when the scenarios are solved independently, that is,

$$\hat{u}_k^j = \sum_{u_k^l | x_k^{p(j)} = x_k^{p(l)}} \pi^{r(l)} u_k^l,$$

where $\pi^{r(l)}$ denotes the probability associated with the branch in the scenario tree at which the control input u_k^l is located. The idea is that after several iterations all the non-anticipativity constrains are satisfied with a desired tolerance ϵ and therefore the augmented cost function converges towards the original one. The parameters $\lambda_k^j \in \mathbb{R}^{n_u}$ and $\rho_k^j \in \mathbb{R}$ are updated at each iteration of the algorithm to improve the convergence. A usual update rule for λ_k^j is: $\lambda_k^j \leftarrow \lambda_k^j + \rho_k^j(u_k^j - \hat{u}_k^j)$ and the parameter ρ_k^j is usually increased after each iteration, with a maximum value that avoids excessively large penalty terms $\rho_k^j \leftarrow \min(\beta\rho_k^j, \rho_{\max})$, where β is a parameter that determines the increase of ρ_k^j. The progressive hedging algorithm used in this thesis for multi-stage NMPC is summarized in Algorithm 1. Further discussion about the updates of the parameters and about the convergence properties of the algorithm can be found in (Rockafellar and Wets, 1991).

7.1.2 Reducing the Number of Iterations

For a complex nonlinear problem, usually the number of iterations that the PHA algorithm needs to converge is large, and therefore the possible advantages in computation time that could be obtained are lost. One possibility to reduce the number of iterations is the use of a bundle decomposition approach. This method is a hybrid between a monolithic scheme and a scenario decomposition approach. In this case, each subproblem of the decomposition algorithm explained in the previous subsection is not a single scenario but a subset (bundle) of all scenarios (see Fig. 7.2). If this subset is properly chosen, for

Algorithm 1 Progressive Hedging Algorithm

```
Require: $\rho_k^j = \rho_{\text{ini}}; \hat{u}_k^j = [0, ..., 0]; \lambda_k^j = [0, ..., 0], \forall(j,k) \in I$
  Solve problem (7.1) $\forall i = 1...N$
  $\hat{u}_k^j = \sum_{u_k^l | x_k^{p(j)} = x_k^{p(l)}} \pi^{r(l)} u_k^l$
  while $\|u_k^j - \hat{u}_k^j\|^2 \geq \epsilon, \forall(j,k) \in I$ do
    for $i = 1 : N$ do
      $\lambda_k^j \leftarrow \lambda_k^j + \rho_k^j(u_k^j - \hat{u}_k^j), \forall(j,k) \in I_i$
      Solve problem (7.1)
    end for
    $\hat{u}_k^j = \sum_{u_k^l | x_k^{p(j)} = x_k^{p(l)}} \pi^{r(l)} u_k^l$
    $\rho_k^j \leftarrow \min(\beta \rho_k^j, \rho_{\max})$
  end while
```

example using very different scenarios in each bundle, the number of iterations needed by the decomposition algorithm to converge decreases significantly, as it is shown in Chapter 8.

7.1.3 Discussion

Instead of using the progressive hedging algorithm, other approches can be used for the scenario decomposition of the multi-stage NMPC problems. For example, it is possible to formulate the problem as a price-based coordination problem as done in (Marti et al., 2013), in which the prices are obtained using the sensitivities of the solution with respect to the control inputs of each scenario. This avoids the difficult tuning of the parameters λ_k^j and ρ_k^j that have to be initialized and updated for the progressive hedging algorithm.

Furthermore, the choice of the values towards which the control inputs have to converge to fulfill the non-anticipativity constraints ($\hat{u}_k^j$) can be done in a smarter way than by taking the weighted average of all the independent control inputs that should satisfy the constraints. For example, if one control input is limited by some active constraints, and it cannot be moved because then the system would violate the constraints, it is beneficial to include this value as $\hat{u}_k^j$, because this will increase the speed of convergence. However, finding this value is not straightforward.

Decomposition approaches are computationally advantageous only if the computation

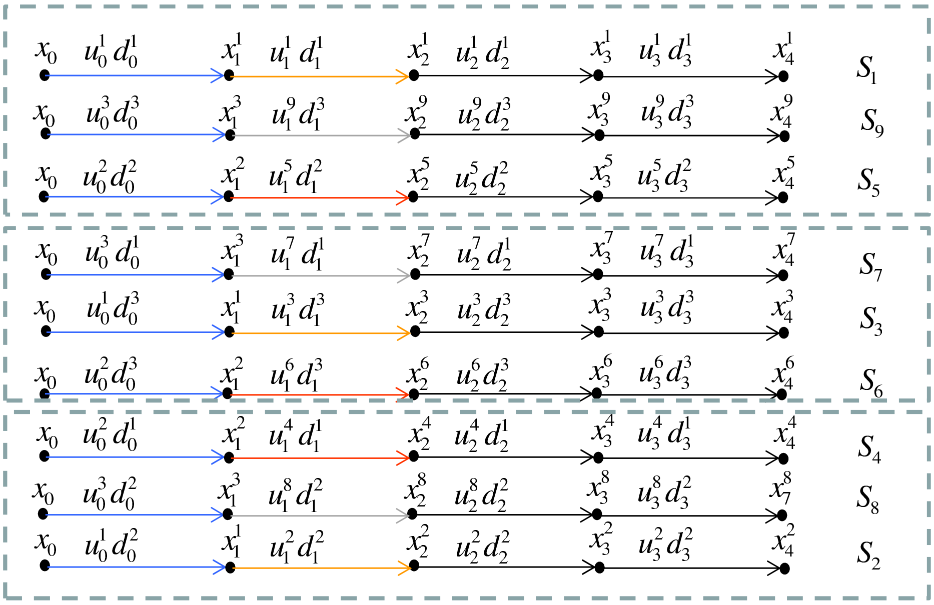

Figure 7.2: Scenario tree representation of the uncertainty evolution using bundle scenario decomposition for multi-stage NMPC with robust horizon $N_R = 2$ and prediction horizon $N_P = 4$. The scenarios in the same box represent the bundles and the arrows with the same color represent the non-anticipativity constraints.

time grows clearly faster than linearly with the number of scenarios, or if a high degree of parallelization is possible so that the independent scenarios can be solved in parallel. However, if efficient tools as the ones described in this thesis are used, with exact computation of first and second order derivative information, the computation time increases only slightly more than linearly with the number of scenarios and therefore it is hardly possible to achieve a shorter computation time with decomposition approaches, even if some paralellization is possible. Nevertheless, decomposition approaches require much less computer memory for the solution of the problems. This may be an important advantage for huge problems. Illustrative results for the decomposition approach and its comparison with the monolithic calculation are provided in the next chapter.

Part III

Simulation and Experimental Results

Chapter 8

Multi-stage NMPC of Polymerization Processes

This chapter presents results that show the advantages of multi-stage NMPC with respect to several other approaches by simulation studies of two different polymerization processes.

Polymerization processes are characterized by strongly exothermic reactions with nonlinear kinetics, and often a very precise control of the reaction temperature is necessary to ensure that the final product will have the desired quality. Moreover, several parameters such as the quality of the raw materials, the heat transfer coefficients or other disturbances are not exactly known and introduce uncertainty in the nonlinear models, which makes the satisfaction of the tight constraints a difficult task. This chapter shows that multi-stage NMPC is a very promising strategy to achieve this task with a low degree of conservativeness.

The first section of this chapter includes a comparison of multi-stage NMPC with a standard NMPC approach in which the uncertainties are ignored and with a min-max approach that optimizes for the worst-case value of the uncertainty for the Chylla-Haase benchmark problem presented in (Chylla and Haase, 1993). Also, the approach has been applied to the situation when only some states can be measured and the remaining states are estimated with an Extended Kalman Filter (EKF). It is shown that multi-stage NMPC also has satisfactory performance in this situation and it performs better than a standard NMPC approach even if the uncertainty is estimated. This section is largely identical to (Lucia et al., 2013a).

The second section of this chapter shows the results of multi-stage NMPC applied to an industrial batch polymerization reactor model provided by BASF SE in the framework of the European Project EMBOCON (EMBOCON, 2014). Here it is shown that multi-stage NMPC performs better than the other robust strategies that have been described in Chapter 3. Most of the work presented in this sections is identical to (Lucia et al., 2014b).

8.1 The Chylla-Haase Polymerization Reactor

The Chylla-Haase Reactor control problem which has been introduced in (Chylla and Haase, 1993) is a well-known benchmark problem for temperature control of semi-batch polymerizations, which constitute strongly nonlinear time-varying systems. A schematic depiction of this process is given in Fig. 8.1. It consists of a pilot scale stirred tank reactor where an emulsion polymerization reaction takes place and a heat exchange system with a jacket and a recirculation loop. The heat exchanger can be used both for heating the reactor up or for cooling it down. Depending on whether the system is operated in the heating or in the cooling mode, medium-pressure steam or cold water is injected into the recirculation loop. As it is usual in the polymer industry, the reactor is used to produce different products, i.e. polymer grades. The different products are obtained from different recipes, which consist of a sequence of charging, heating, feeding and holding steps that may or may not be repeated, as it is described in the references (Chylla and Haase, 1993), (Graichen et al., 2006) and (Beyer et al., 2008). The end-use properties of the product mainly depend on the temperature at which the polymerization takes place, therefore a very precise temperature control is required in order to guarantee that the final product will have an acceptable quality. In (Chylla and Haase, 1993), it is stated that the reactor temperature must stay within a very tight range of ±0.6 K around the specified reaction temperature.

In the original version of the Chylla-Haase benchmark reactor (CHBR) problem the monomer is dosed into the reactor at a constant flow rate, which means that there is no degree of freedom for increasing the process productivity by reducing the duration of the feeding period. In (Finkler et al., 2013), a modified version of the CHBR has been proposed in which the monomer inlet flow varies over time. This turns the CHBR into a

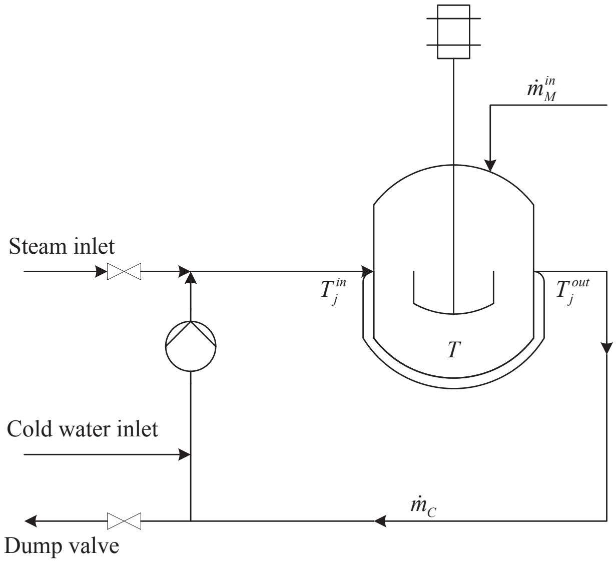

Figure 8.1: Scheme of the Chylla-Haase Benchmark Reactor.

much more challenging problem that involves simultaneous optimization of the monomer feed and of the cooling/heating along the batch and includes online process performance optimization under tight constraints. This version of the problem is the one considered in the remainder of this section.

Modeling of the Process

An experimentally validated model that describes the system behavior was given in (Chylla and Haase, 1993). This model has been widely investigated by the process control community and several mistakes in its formulation have been identified and corrected. The reader is directed to (Graichen et al., 2006) and (Beyer et al., 2008) for a more detailed discussion on these corrections. The model that is used in this investigation consists of a set of ODEs given by equations (8.1a) to (8.1e):

$$\dot{m}_\mathrm{M} = \dot{m}_\mathrm{M}^\mathrm{in} - r_\mathrm{P}, \tag{8.1a}$$

$$\dot{m}_\mathrm{P} = r_\mathrm{P}, \tag{8.1b}$$

$$\dot{T} = \frac{\dot{m}_\mathrm{M}^\mathrm{in} C_\mathrm{p,M}(T_\mathrm{amb} - T) + UA(\bar{T}_\mathrm{J} - T)}{m_\mathrm{M} C_\mathrm{p,M} + m_\mathrm{P} C_\mathrm{p,P} + m_\mathrm{W} C_\mathrm{p,W}} + \frac{UA_\mathrm{loss}(T_\mathrm{amb} - T) + r_\mathrm{P}\Delta H_\mathrm{P}}{m_\mathrm{M} C_\mathrm{p,M} + m_\mathrm{P} C_\mathrm{p,P} + m_\mathrm{W} C_\mathrm{p,W}}, \tag{8.1c}$$

$$\dot{T}_\mathrm{j}^\mathrm{out} = \frac{\dot{m}_\mathrm{C} C_\mathrm{p,W}(T_\mathrm{j}^\mathrm{in}(t-\theta_1) - T_\mathrm{j}^\mathrm{out}) + UA(T - \bar{T}_\mathrm{j})}{m_\mathrm{C} C_\mathrm{p,W}}, \tag{8.1d}$$

$$\dot{T}_{\mathrm{j}}^{\mathrm{in}} = \dot{T}_{\mathrm{j}}^{\mathrm{out}}(t-\theta_2) + \frac{T_{\mathrm{j}}^{\mathrm{out}}(t-\theta_2) - T_{\mathrm{j}}^{\mathrm{in}}}{\tau_{\mathrm{p}}} + \frac{K_{\mathrm{p}}(c)}{\tau_{\mathrm{p}}}, \tag{8.1e}$$

where m_{M}, m_{P}, and m_{W} are the holdups of monomer, polymer, and water inside the reactor, T, $T_{\mathrm{j}}^{\mathrm{out}}$, $T_{\mathrm{j}}^{\mathrm{in}}$, T_{amb} are the inner reactor temperature, the water temperature at the jacket outlet, the water temperature at the jacket inlet, and the ambient temperature, r_{P} is the polymerization rate, ΔH_{P} is the reaction enthalpy, U is the overall heat transfer coefficient, A is the heat transfer area, $\dot{m}_{\mathrm{M}}^{\mathrm{in}}$ is the monomer inlet flow, $\dot{m}_{\mathrm{C}}$ is the flow rate of water across the jacket, m_{C} is the mass of water inside the jacket, $C_{\mathrm{p,M}}$, $C_{\mathrm{p,P}}$, and $C_{\mathrm{p,W}}$ are the thermal capacities of monomer, polymer, and water, respectively, θ_1 and θ_2 are time delays, τ_{P} is the jacket time constant; the heating/cooling usage c is a control command that can vary from 0 to 100%, and the heating/cooling function K_{p} is a function of c with the split-range characteristic given by (8.2), with T_{inlet} and T_{steam} being the cold water and the medium-pressure steam temperatures:

$$K_{\mathrm{p}}(c) = \begin{cases} 0.8 \times 30^{-c/50}(T_{\mathrm{inlet}} - T_{\mathrm{j}}^{\mathrm{in}}), & c < 50\%, \\ 0, & c = 50\%, \\ 0.15 \times 30^{c/50-2}(T_{\mathrm{steam}} - T_{\mathrm{j}}^{\mathrm{in}}), & c > 50\%. \end{cases} \tag{8.2}$$

The expressions for the computation of r_{P}, U, and A are given as:

$$r_{\mathrm{P}} = i_{\mathrm{p}} k m_{\mathrm{M}}, \tag{8.3}$$

$$U = \frac{1}{h^{-1} + h_{\mathrm{f}}^{-1}}, \tag{8.4}$$

$$A = \left(\frac{m_{\mathrm{M}}}{\rho_{\mathrm{M}}} + \frac{m_{\mathrm{P}}}{\rho_{\mathrm{P}}} + \frac{m_{\mathrm{W}}}{\rho_{\mathrm{W}}}\right)\frac{P}{B_1} + B_2, \tag{8.5}$$

where i_{p} is the monomer impurity factor, k is the first order kinetic constant, $1/h$ is the film heat transfer coefficient and $1/h_{\mathrm{f}}$ is the fouling factor which depends on the number of batches that have been produced since the last cleaning. The additional semi-empirical relations for the computation of k and h as a function of the system states are given by equations (8.6) to (8.12):

$$k = k_0 e^{\frac{E}{RT}}(k_1\mu)^{k_2}, \tag{8.6}$$

$$\mu = c_0 e^{c_1 f} 10^{c_2(\frac{a_0}{T} - c_3)}, \tag{8.7}$$

$$f = \frac{m_{\mathrm{P}}}{m_{\mathrm{M}} + m_{\mathrm{P}} + m_{\mathrm{W}}}, \tag{8.8}$$

$$h = d_0 e^{d_1 \mu_{\mathrm{W}}}, \tag{8.9}$$

$$\mu_{\mathrm{W}} = c_0 e^{c_1 f} 10^{c_2(\frac{u_0}{T_{\mathrm{W}}} - c_3)}, \tag{8.10}$$

$$T_{\mathrm{W}} = \frac{T + \bar{T}_{\mathrm{j}}}{2}, \tag{8.11}$$

$$\bar{T}_{\mathrm{J}} = \frac{T_{\mathrm{j}}^{\mathrm{out}} + T_{\mathrm{j}}^{\mathrm{in}}}{2}, \tag{8.12}$$

where μ is the viscosity of the reactant mixture and f is the mass fraction of polymer inside the reactor. The values for all parameters that appear in the model equations can be found in the literature on the CHBR ((Chylla and Haase, 1993), (Graichen et al., 2006) and (Beyer et al., 2008)).

Modeling of the Uncertainties

As it was discussed in (Chylla and Haase, 1993), the main uncertainties and disturbances that affect the CHBR can be condensed in three variables, the impurity factor i_{p}, the fouling factor $1/h_{\mathrm{f}}$ and the ambient temperature T_{amb}. The impurity factor i_{p} describes the fluctuations in the reaction rate caused by impurities in the raw materials. It varies randomly from batch to batch within the range 0.8 to 1.2 following a uniform distribution and it is usually considered as constant during one batch. The fouling factor $1/h_{\mathrm{f}}$ describes the decrease in U due to the formation of a polymer film on the reactor wall during the successive batches. It varies from 0 to 0.704 $\mathrm{m}^2\mathrm{KkW}^{-1}$. The ambient temperature describes the variations in the temperature of the monomer inlet feed as well as the reactor and jacket temperatures at the beginning of the batch. It can vary from 280 to 305 K. In, (Chylla and Haase, 1993) it was also suggested that the time delays θ_1 and θ_2 may vary by $\pm 25\%$ when compared to the nominal values. As it turned out during the simulations that these variations are not significant, they are neglected here. In (Chylla and Haase, 1993) data for two different products (A and B) was given, product A is considered here.

Tables 8.1 and 8.2 provide a summary of the benchmark problem that has 5 differential states, 7 algebraic states to model the delayed states and 2 control inputs.

Table 8.1: State constraints and typical values during the feeding phase.

States	Min.	Max.	Typ. values	Unit
m_{M}	0	inf.	0 - 6	kg
m_{P}	0	inf.	0 - 50	kg
T	81.632	82.832	81.632 - 82.832	°C
$T_{\mathrm{j}}^{\mathrm{out}}$	5	95	25 - 95	°C
$T_{\mathrm{j}}^{\mathrm{in}}$	5	95	20 - 95	°C

Table 8.2: Control input constraints and typical values during the feeding phase.

Control inputs	Min.	Max.	Typ. values	Unit
c	0	100	0 - 50	–
$\dot{m}_{\mathrm{M}}^{\mathrm{in}}$	0	0.0227	0 - 0.0227	kg s^{-1}

8.1.1 Standard, Min-max and Multi-stage Economic NMPC of the Polymerization Reactor Example with Full Feedback Information

This section presents a comparison of the performance of standard, multi-stage and min-max NMPC of the Chylla-Haase benchmark reactor control problem under uncertainty. It is considered that all the states can be measured exactly. In this case, only the uncertainty in the impurity factor is taken into account, since the ambient temperature can be measured and the fouling factor can be estimated depending on the number of the batch[1]. As described in the benchmark problem, three different values are considered for the impurity factor, i.e. $i_{\mathrm{p}} = \{0.8, 1.0, 1.2\}$.

The benchmark problem consists of three different phases: heating, feeding and holding. The discussion is focused on the feeding part, because it represents the most challenging control problem. For each phase, a different optimization problem is formulated since the control objectives and constraints are different. The optimization problem solved at

[1] As it is described in (Beyer et al., 2008), the reactor is cleaned every five batches and the fouling factor may vary from 0 (clean reactor) to 0.704 m^{2}KkW^{-1} (fouled reactor).

each sampling time for the heating part is:

$$\min_{x_k^j, u_k^j \in I} \sum_{i=1}^{N} \omega_i \sum_{k=0}^{N_\mathrm{P}-1} q(T_k^j - T_\mathrm{set})^2 + r_\Delta \Delta {c_k^j}^2 \ , \qquad \forall\, T_k^j,\, c_k^j \in S_i, \tag{8.13a}$$

subject to: (8.13b)

$$x_{k+1}^j = f(x_k^{p(j)}, u_k^j, d_k^{r(j)}), \qquad \forall\, (j, k+1) \in I, \tag{8.13c}$$

$$5 \le T_{\mathrm{j},k}^{\mathrm{in},j} \le 95, \qquad \forall\, (j,k) \in I, \tag{8.13d}$$

$$5 \le T_{\mathrm{j},k}^{\mathrm{out},j} \le 95, \qquad \forall\, (j,k) \in I, \tag{8.13e}$$

$$0 \le c_k^j \le 100, \qquad \forall\, (j,k) \in I, \tag{8.13f}$$

$$0 \le \dot{m}_{\mathrm{M},k}^{\mathrm{in},j} \le 0, \qquad \forall\, (j,k) \in I, \tag{8.13g}$$

$$u_k^j = u_k^l \text{ if } x_k^{p(j)} = x_k^{p(l)}, \qquad \forall\ (j,k), (l,k) \in I. \tag{8.13h}$$

The cost function (8.13a) includes a tracking term of the reactor temperature, and a regularization term for the control movements in the jacket to avoid oscillations. Each node in the scenario tree contains all the states and control inputs in the model, i.e., $x_k^j = [m_{\mathrm{M},k}^j, m_{\mathrm{P},k}^j, T_k^j, T_k^{\mathrm{in},j}, T_k^{\mathrm{out},j}]^T$ and $u_k^j = [c_k^j, \dot{m}_{\mathrm{M},k}^{\mathrm{in},j}]^T$. The tuning parameters q and r_Δ are chosen to be $q = 1$ and $r_\Delta = 0.5$ and T_set is the desired reaction temperature. The different constraints of the optimization problem represent the system dynamics (8.13c), the constraints in the jacket temperature (8.13d, 8.13e), since water is assumed to be used in the jacket, and the constraints on the inputs (8.13f, 8.13g). For all the optimization problems in this section, the dynamics of the system are discretized with an implicit Euler method, so that the state at the next sampling time x_{k+1}^j can be written as:

$$x_{k+1}^j = x_k^{p(j)} + t_\mathrm{step} \cdot f(x_k^{p(j)}, u_k^j, d_k^{r(j)}),$$

where $f(\cdot)$ is the right-hand side of the differential equations that define the model. Once the optimization problem is solved, the real system is simulated using a precise solution of the model and not the Euler approximation. The sampling time used in all the optimization problems is the same, $t_\mathrm{step} = 30$ s. The prediction horizon is equal to the control horizon and the same for all the problems presented in this section, $N_P = 5$ steps (150 s in total).

For the holding part, the same optimization problem is solved at each sampling time, including a constraint for the reactor temperature (8.14d) and using as tuning parameters $q = 1$ and $r_\Delta = 0.1$.

$$\min_{x_k^j, u_k^j \in I} \sum_{i=1}^{N} \omega_i \sum_{k=0}^{N_\mathrm{P}-1} q(T_k^j - T_\mathrm{set})^2 + r_\Delta \Delta {c_k^j}^2 \ , \qquad \forall\, T_k^j,\, c_k^j \in S_i, \tag{8.14a}$$

$$\begin{aligned}
&\text{subject to:} && && (8.14b)\\
&x_{k+1}^{j} = f(x_k^{p(j)}, u_k^j, d_k^{r(j)}), && \forall\, (j, k+1) \in I, && (8.14c)\\
&T_{set} - 0.6 \leq T_k^j \leq T_{\mathrm{set}} + 0.6 && \forall\, (j,k) \in I, && (8.14d)\\
&5 \leq T_{\mathrm{j},k}^{\mathrm{in},j} \leq 95 && \forall\, (j,k) \in I, && (8.14e)\\
&5 \leq T_{\mathrm{j},k}^{\mathrm{out},j} \leq 95 && \forall\, (j,k) \in I, && (8.14f)\\
&0 \leq c_k^j \leq 100 && \forall\, (j,k) \in I, && (8.14g)\\
&0 \leq \dot{m}_{\mathrm{M},k}^{\mathrm{in},j} \leq 0 && \forall (j,k) \in I, && (8.14h)\\
&u_k^j = u_k^l \text{ if } x_k^{p(j)} = x_k^{p(l)} && \forall\, (j,k), (l,k) \in I. && (8.14i)
\end{aligned}$$

Since this work is focused on the feeding period, a fixed time is used for the heating phase (30 minutes) and a fixed time of 15 minutes for the holding phase, which is entered once all the necessary monomer has been fed into the reactor.

As discussed in (Engell, 2007), the optimal operation of the plant under the presence of uncertainties is the main goal of process control, in contrast to classical set-point tracking approaches that can be sometimes even counterproductive. Therefore, an economic cost function is chosen for the feeding phase, and no tracking term for the temperature is used. The cost function to be minimized is:

$$\min_{x_k^j, u_k^j \in I} \sum_{i=1}^{N} \omega_i \sum_{k=0}^{N_\mathrm{p}-1} r(\dot{m}_{\mathrm{M},k}^{\mathrm{in},j} - \dot{m}_{\mathrm{M}}^{\max})^2 + r_\Delta \Delta {c_k^j}^2 \quad , \qquad \forall\, \dot{m}_{\mathrm{M},k}^{\mathrm{in},j},\, c_k^j \in S_i, \quad (8.15a)$$

$$\begin{aligned}
&\text{subject to:} && && (8.15b)\\
&x_{k+1}^{j} = f(x_k^{p(j)}, u_k^j, d_k^{r(j)}), && \forall\, (j, k+1) \in I, && (8.15c)\\
&T_{\mathrm{set}} - 0.6 \leq T_k^j \leq T_{\mathrm{set}} + 0.6, && \forall\, (j,k) \in I, && (8.15d)\\
&5 \leq T_{\mathrm{j},k}^{\mathrm{in},j} \leq 95, && \forall\, (j,k) \in I, && (8.15e)\\
&5 \leq T_{\mathrm{j},k}^{\mathrm{out},j} \leq 95, && \forall\, (j,k) \in I, && (8.15f)\\
&0 \leq c_k^j \leq 100, && \forall\, (j,k) \in I, && (8.15g)\\
&0 \leq \dot{m}_{\mathrm{M},k}^{\mathrm{in},j} \leq \dot{m}_{\mathrm{M}}^{\max} && \forall (j,k) \in I, && (8.15h)\\
&u_k^j = u_k^l \text{ if } x_k^{p(j)} = x_k^{p(l)} && \forall\, (j,k), (l,k) \in I. && (8.15i)
\end{aligned}$$

where $\dot{m}_{\mathrm{M}}^{\max}$ is the maximum allowed feeding rate, and the tuning parameters are $r = 10$, $r_\Delta = 0.5$.

Fig. 8.2 shows the result for standard NMPC when there is no model mismatch. Fig. 8.2 shows the control inputs and the reactor temperature for a whole batch. During the heating phase until $t = 30$ min no monomer is fed and the reactor is heated up to the reaction temperature. During the feeding phase monomer is feed as fast as possible as long as it is possible to satisfy the temperature constraints using the cooling jacket. Once the cooling is saturated (around $t = 45$ min) the monomer feed is adjusted so that the constraint is not violated. In the holding phase no monomer is fed and the reactor temperature is controlled using the cooling jacket. Since in this case there is no error between the prediction and the evolution of the real plant (in both models the impurity factor is $i_{\mathrm{p}} = 1.0$), the performance is good and no constraint violations occur. However, as it can be seen in Fig. 8.3 and in Fig. 8.4, if the model used in the optimizer differs from the model of the plant, infeasibilities may occur. In this case, the monomer impurity factor of the real plant is $i_{\mathrm{p}} = 0.8$ and $i_{\mathrm{p}} = 1.2$ and the impurity factor used in the model of the optimizer is $i_{\mathrm{p}} = 1.0$. Since the end-use properties of the polymer depend strongly on the temperature at which the polymerization takes place, this is not an acceptable performance of the controller.

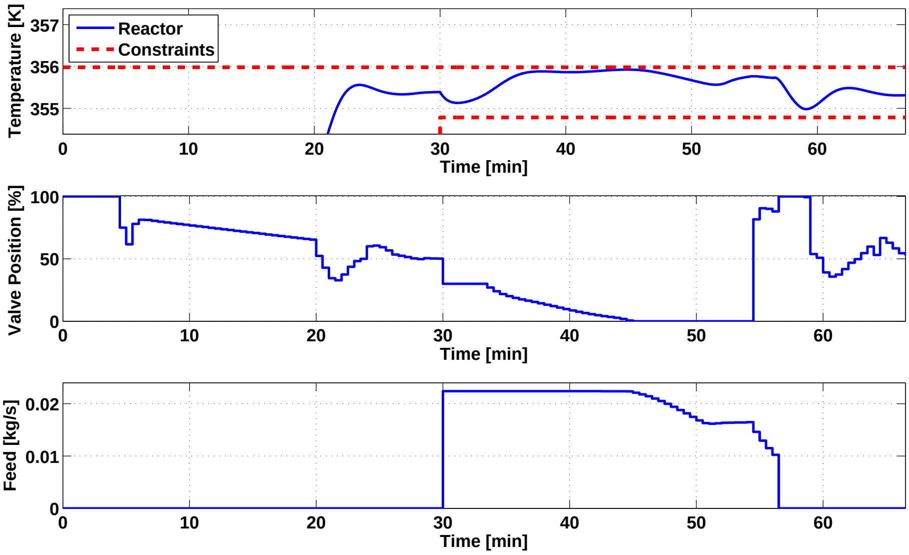

Figure 8.2: Reactor temperature, valve position and monomer feed rate for standard NMPC, controller impurity factor = plant impurity factor = 1.0.

In contrast to standard NMPC, if multi-stage NMPC is used, the constraints are not violated for any of the possible values of the impurity factor. Fig. 8.5 shows the reactor temperature and the control inputs when the real purity factor is $i_{\mathrm{p}} = 0.8$ and the robust horizon is considered to be $N_R = 1$, that is, the tree branches only in the first stage

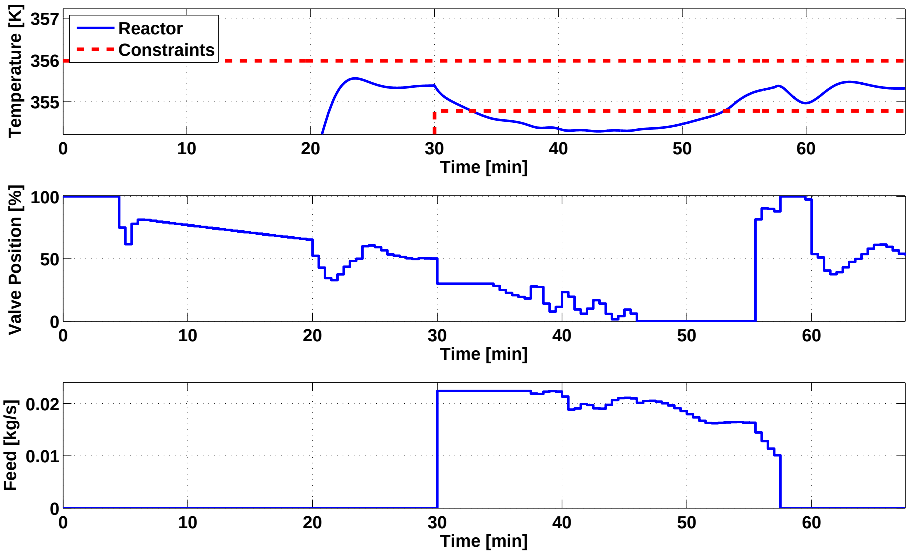

Figure 8.3: Reactor temperature, valve position and monomer feed rate for standard NMPC, controller impurity factor = 1.0, plant impurity factor = 0.8.

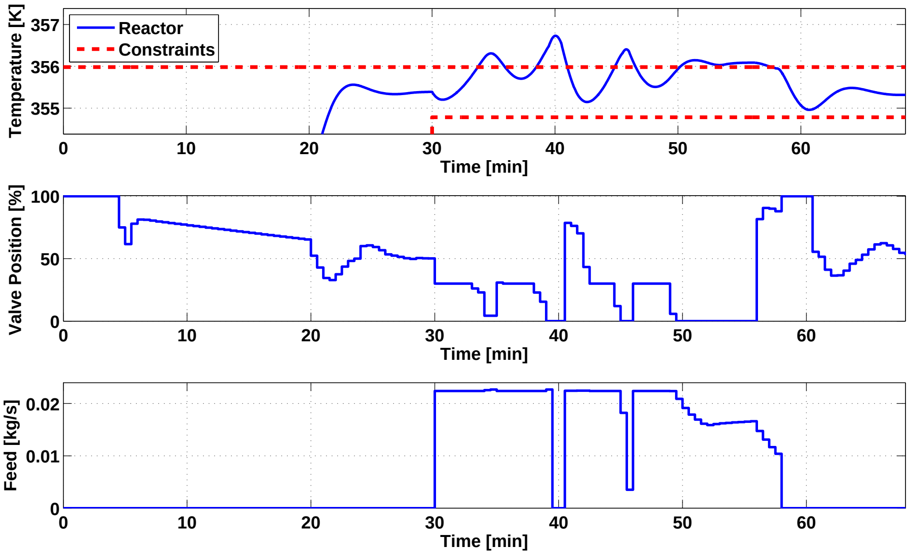

Figure 8.4: Reactor temperature, valve position and monomer feed rate for standard NMPC, controller impurity factor = 1.0, plant impurity factor = 1.2.

and thereafter the uncertainty is considered to be constant. Fig. 8.6 and Fig. 8.7 show that the multi-stage NMPC controller can satisfy the constraints for all cases of the uncertainty included in the scenario tree. It can be observed in Fig. 8.7 that there is a minor violation of the upper temperature constraint at time $t = 34$ min. This violation is due to the Euler approximation of the dynamics in the prediction and can be avoided by choosing a smaller t_{step}. Here it is assumed that such a violation is negligible. Fig. 8.8 and Fig. 8.9 show the reactor temperature and the monomer feed rate obtained when running multi-stage NMPC for many values of the uncertainty between the pre-specified bounds, including also values that are not explicitly included in the tree. It can be seen that also in this case the constraints are satisfied for all the values of the uncertainty.

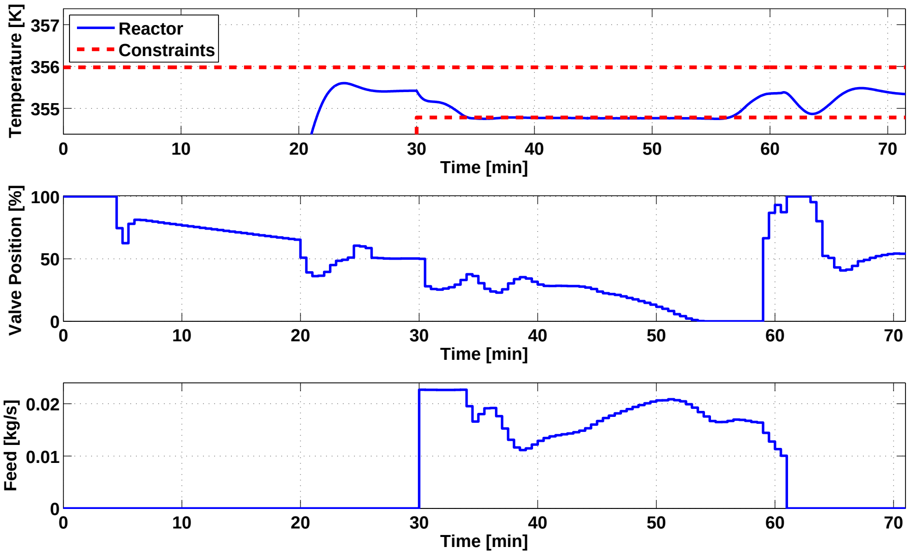

Figure 8.5: Reactor temperature, valve position and monomer feed rate for multi-stage NMPC, plant impurity factor = 0.8.

Fig. 8.10 shows a comparison of the results for different robust horizons. As it can be seen, the performance is very similar and therefore the case with robust horizon equal to one will be used in the remainder of the paper since it provides a good solution with a lower computational cost.

As mentioned in Chapter 3, the framework of multi-stage NMPC includes the closed-loop min-max NMPC formulation by simply exchanging the summation by the maximization operator in the general cost function defined in (3.5) if all the probabilities ω_i are equal. In order to obtain a differentiable cost function, a slack variable is used as cost function and additional constraints (8.16j) are imposed such that the new slack variable is larger than the cost of any of the scenarios. In this way a differentiable min-max formulation is obtained. The mathematical formulation of the multi-stage min-max NMPC can be

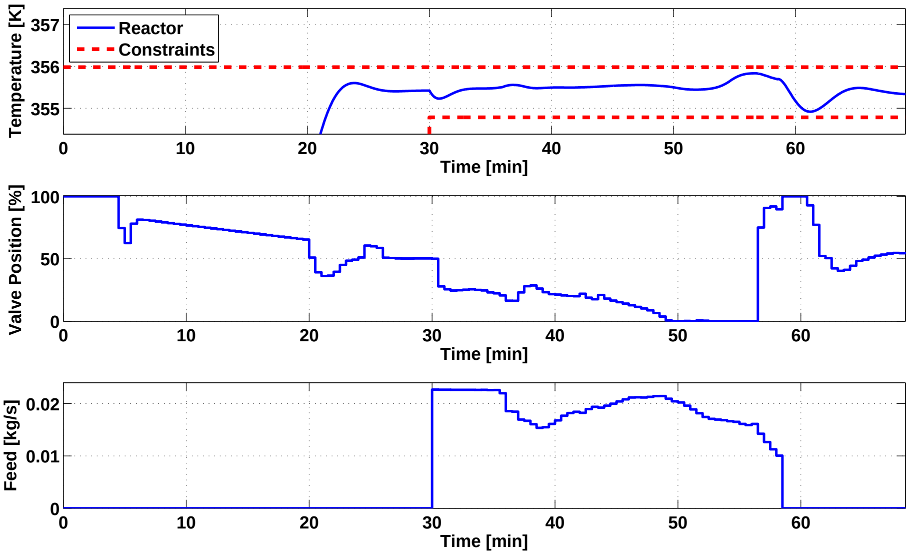

Figure 8.6: Reactor temperature, valve position and monomer feed rate for multi-stage NMPC, plant impurity factor = 1.0.

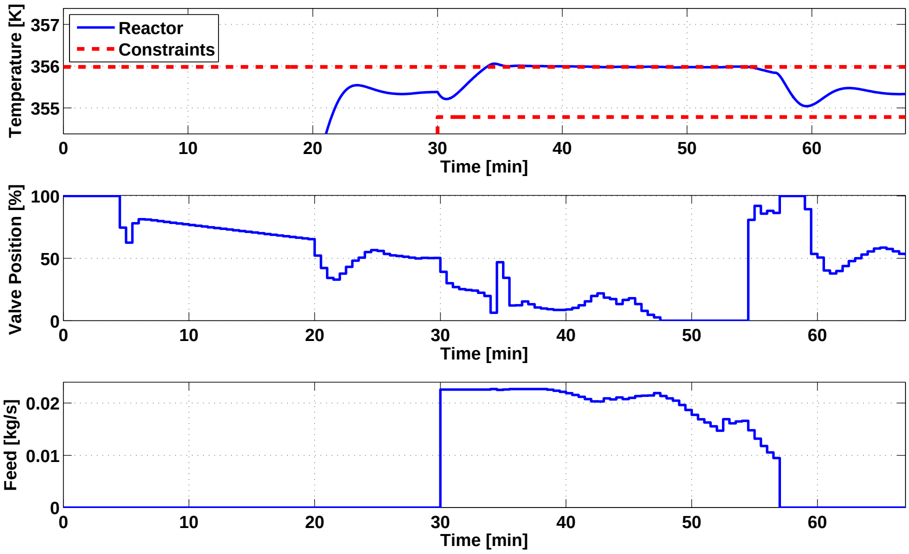

Figure 8.7: Reactor temperature, valve position and monomer feed rate for multi-stage NMPC, plant impurity factor = 1.2.

written as:

$$\min_{x_k^j, u_k^j \in \epsilon} \gamma, \tag{8.16a}$$

subject to: (8.16b)

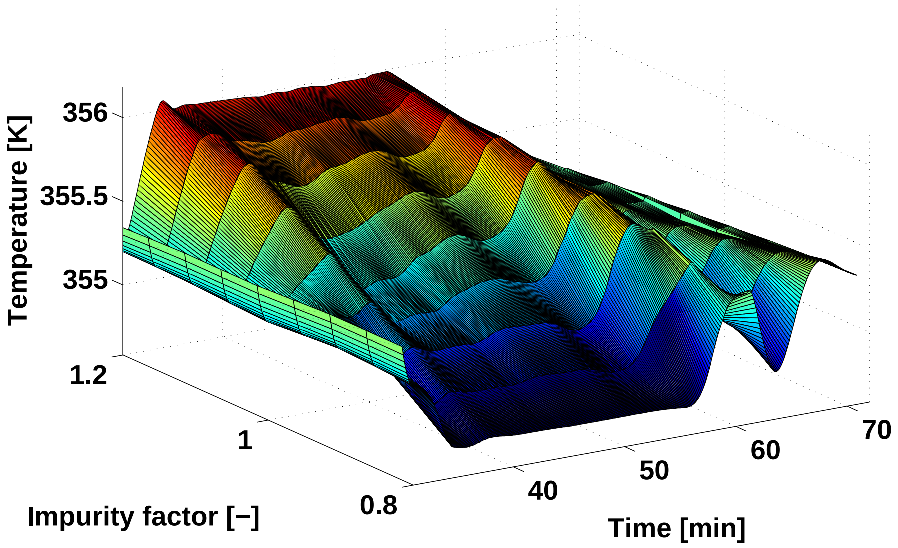

Figure 8.8: Reactor temperature for multi-stage NMPC for different values of the plant impurity factor.

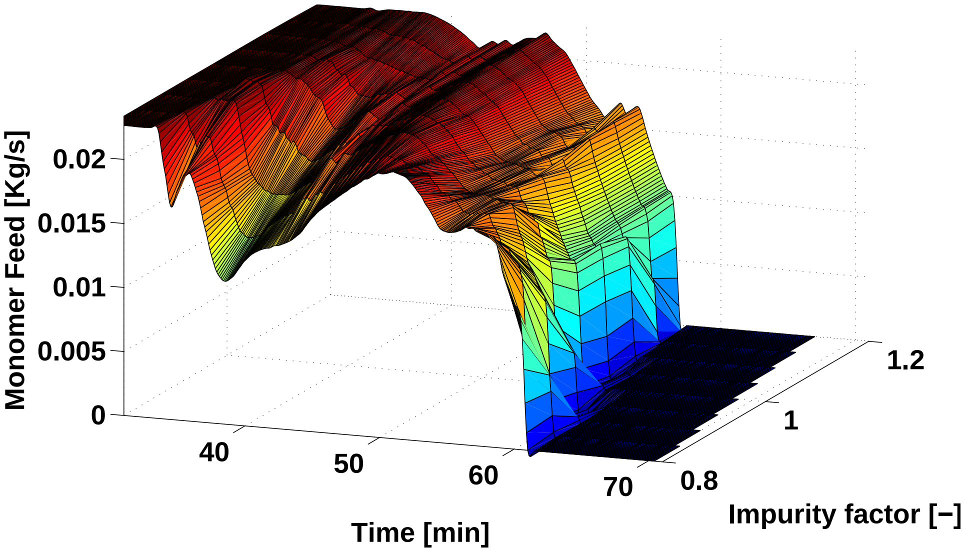

Figure 8.9: Monomer feed rate for multi-stage NMPC for different values of the plant impurity factor.

$$x_{k+1}^{j} = f(x_k^{p(j)}, u_k^j, d_k^{r(j)}), \qquad \forall\,(j, k+1) \in I, \tag{8.16c}$$

$$T_{\text{set}} - 0.6 \le T_k^j \le T_{\text{set}} + 0.6, \qquad \forall\,(j, k) \in I, \tag{8.16d}$$

$$5 \le T_{\text{j},k}^{\text{in},j} \le 95, \qquad \forall\,(j, k) \in I, \tag{8.16e}$$

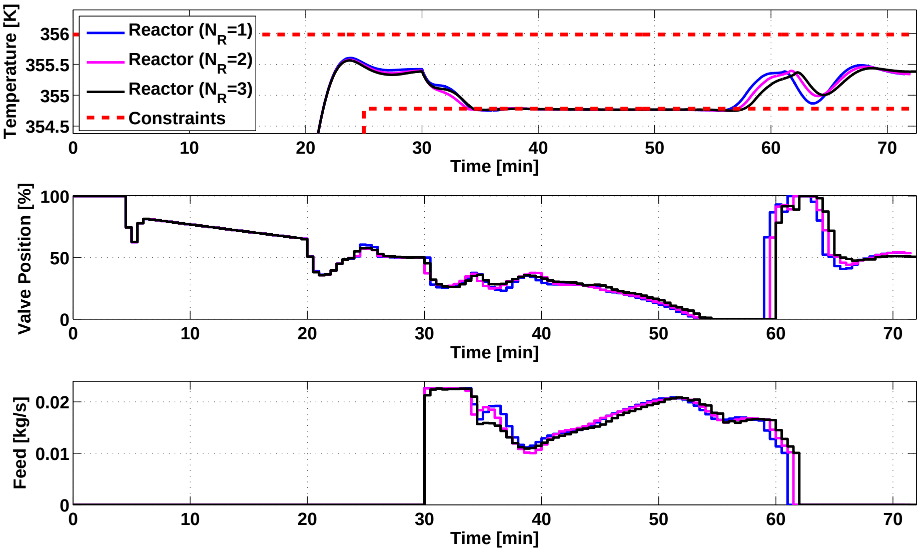

Figure 8.10: Reactor temperature, valve position and monomer feed rate for multi-stage NMPC, for different robust horizons N_P and for plant impurity factor $= 0.8$.

$$5 \leq T^{\text{out},j}_{\text{j},k} \leq 95, \qquad \forall\,(j,k) \in I, \tag{8.16f}$$

$$0 \leq c^{j}_{k} \leq 100, \qquad \forall\,(j,k) \in I, \tag{8.16g}$$

$$0 \leq \dot{m}^{\text{in}}_{\text{M}} \leq \dot{m}^{\text{max}}_{\text{M}}, \qquad \forall (j,k) \in I, \tag{8.16h}$$

$$u^{j}_{k} = u^{l}_{k} \text{ if } x^{p(j)}_{k} = x^{p(l)}_{k}, \qquad \forall\,(j,k),(l,k) \in I, \tag{8.16i}$$

$$\gamma \geq \omega_i \cdot J_i \qquad \forall i = 1..N, \tag{8.16j}$$

where J_i is the cost of scenario i defined for the feeding phase as:

$$J_i = \sum_{k=0}^{N_\text{p}-1} r(\dot{m}^{\text{in},j}_{\text{M},k} - \dot{m}^{\text{max}}_{\text{M}})^2 + r_\Delta \Delta {c^{j}_{k}}^{2} \;\; , \forall\, \dot{m}^{\text{in},j}_{\text{M},k},\, c^{j}_{k} \in S_i, \tag{8.17}$$

As expected, min-max NMPC also satisfies the temperature constraints for all possible values of the uncertainty. However, the performance is decreased due to the worst-case nature of the approach. A comparison of multi-stage and min-max NMPC is shown in Fig. 8.11 for the case of impurity factor $i_\text{p} = 0.8$. Note that multi-stage min-max NMPC is not the same as classical min-max NMPC where no recourse is included.

Table 8.3 shows a comparison of the feeding times for each controller, including whether the corresponding controller is capable of maintaining the temperature between the specified bounds or not. It can be seen that multi-stage NMPC obtains the best performance. The average computation time per optimization problem iteration needed for the different NMPC algorithms is shown in Table 8.4. This computation times illustrate that the

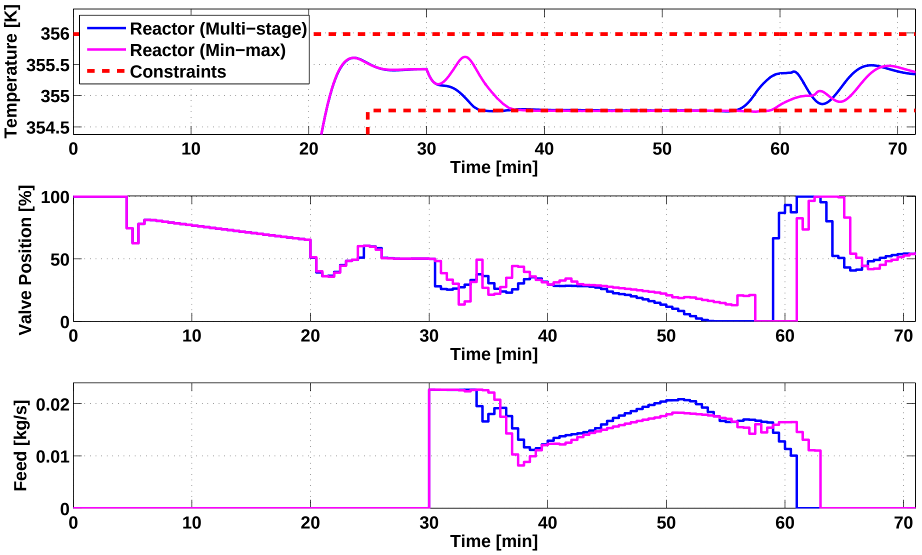

Figure 8.11: Reactor temperature, valve position and monomer feed rate for multi-stage and min-max NMPC, plant impurity factor = 0.8.

solution is possible in real time for a robust horizon $N_R = 1$. As it will be shown in the next section, an efficient implementation using automatic differentiation to provide exact first and second order derivative information that greatly improves the computation times, enlarging the range of applications under real time constraints. The NLP contains 82 variables for standard NMPC, 222 for multi-stage NMPC with robust horizon $N_R = 1$, 558 with $N_R = 2$, and 1314 with $N_R = 3$. The hardware used is a 4-core Intel i5 processor (2.67 GHz) with 4 Gb of RAM and the resulting NLPs are solved using SNOPT via TOMLAB/MATLAB.

In order to illustrate the influence of the design of the scenario tree in the performance of the controller, Table 8.5 presents different designs of the scenario tree with robust horizon 1 and the resulting performance of the controller using multi-stage NMPC with these trees is summarized in Table 8.6. It can be seen that as a guideline for the design of the tree, extreme scenarios should be chosen in order to achieve robust constraint satisfaction and that the inclusion of intermediate scenarios increases the performance of the resulting controller.

Table 8.3: Performance comparison between standard, multi-stage ($N_R = 1$) , and min-max ($N_R = 1$) NMPC.

		Feeding time in minutes		
Impurity plant	Impurity controller	Standard NMPC	Multi-stage NMPC	Min-max NMPC
0.8	0.8 1.0 1.2	27.00 infeasible infeasible	31.00	33.00
1.0	0.8 1.0 1.2	infeasible 26.50 infeasible	28.00	28.50
1.2	0.8 1.0 1.2	infeasible infeasible 26.50	27.00	27.50

8.1.2 Multi-stage NMPC with State and Parameter Estimation

In a real plant, measurements of all state variables usually are not available. Since this information is necessary for the initialization of the optimizer, state estimation is required if the states cannot be measured. As shown in (Lucia et al., 2012), multi-stage NMPC can also be applied using state estimation based upon noisy temperature measurements. However, in many cases, it is possible to estimate some uncertain parameters online as well. Then standard NMPC can be applied using the estimated value of the uncertainty in the model that is employed in the optimization. This raises the question whether there is an advantage in using a robust formulation in this case. As it is demonstrated here, the nominal controller with state estimation is not robust against a change in the parameter value, even if it is perfectly estimated at the next sampling time, while the use of the multi-stage NMPC scheme robustifies the system also against changes of the operation conditions of the plant, such as a sudden change on the impurity factor of the monomer input. In order to illustrate this advantage of multi-stage NMPC, an Extended Kalman Filter is simulated based on the assumption that only noisy temperature measurements are available with a standard deviation of $\sigma = 0.05$K as described in the

Table 8.4: Average computation times for one iteration of standard, multi-stage, and min-max NMPC algorithms for different robust horizons.

NMPC algorithm	**Computation time (s)**	**# of scenarios**
Standard	0.74	1
Multi-stage ($N_R = 1$)	2.98	3
Min-max ($N_R = 1$)	5.73	3
Multi-stage ($N_R = 2$)	18.65	9
Min-max ($N_R = 2$)	31.39	9
Multi-stage ($N_R = 3$)	97.48	27
Min-max ($N_R = 3$)	145.28	27

Table 8.5: Description of several scenario trees with robust horizon $N_R = 1$ with different numbers of branches and values of the uncertainty taken into account.

Scenario tree	**Values of i_p in the tree**
Tree 1	{1.0}
Tree 2	{0.9, 1.1}
Tree 3	{0.8, 1.2}
Tree 4	{0.8, 1.0, 1.2}
Tree 5	{0.8, 0.9, 1.0, 1.1, 1.2}

original benchmark problem (Graichen et al., 2006). The Extended Kalman Filter (EKF) was implemented in the same way as in (Beyer et al., 2008). The estimated state vector is $\hat{x} = [\hat{m}_\mathrm{M}, \hat{m}_\mathrm{P}, \hat{T}, \hat{T}_\mathrm{j}^\mathrm{out}, \hat{q}_0, \hat{U}]^T$ using equations (8.1a) to (8.1d), augmented by $\dot{q}_0 = 0$ and $\dot{U} = 0$. The estimated heat of reaction $\hat{Q}_\mathrm{rea}$ is calculated as:

$$\hat{Q}_\mathrm{rea} = \hat{q}_0 \cdot \hat{m}_\mathrm{M}.$$

The nonzero diagonal elements of the covariance parameters were tuned by simulation and are shown in Table 8.7. The performance of the Extended Kalman Filter is shown in Fig. 8.12, and it can be seen that, the heat of reaction Q_rea, the heat transfer coefficient U and the monomer mass m_M are estimated well. Using the estimated heat of reaction $\hat{Q}_\mathrm{rea}$, it is possible to compute the estimated impurity factor using $\hat{Q}_\mathrm{rea} = \Delta H_\mathrm{P} \hat{i}_p \, \hat{k} \, \hat{m}_\mathrm{M} / MW_\mathrm{M}$,

Table 8.6: Performance comparison of multi-stage NMPC for the different scenario trees

	Feeding time in minutes				
Imp. plant	**Tree 1**	**Tree 2**	**Tree 3**	**Tree 4**	**Tree 5**
0.8	infeasible	infeasible	31.50	31.00	31.00
0.9	infeasible	27.00	29.00	28.50	28.50
1.0	26.50	26.50	28.50	28.00	28.00
1.1	infeasible	26.50	28.50	28.00	28.00
1.2	infeasible	infeasible	27.00	27.00	26.50

Table 8.7: Tunning parameters used in the covariance matrix Q_{EKF} of the EKF.

	Period		
Element in $Q_{\mathbf{EKF}}$	**Heating**	**Feeding**	**Holding**
$\sigma^2(\hat{T}_{\text{j}}^{\text{out}})$	5×10^{-3}	1×10^{-7}	5×10^{-3}
$\sigma^2(\hat{q}_0)$	5×10^{-3}	6×10^{-2}	5×10^{-3}
$\sigma^2(\hat{U})$	5×10^{-5}	5×10^{-3}	5×10^{-6}

where MW_{M} is the molar mass of the monomer, and $\hat{k}$ is the estimated reaction constant which can be computed using (8.6).

It is assumed that a step change in the quality of the raw materials occurs in the middle of the batch ($t = 45$ min). The impurity factor of the input monomer feed changes from $i_{\text{p}}^{\text{in}} = 0.8$ to $i_{\text{p}}^{\text{in}} = 1.0$. Then, the new monomer fed into the reactor is mixed with the monomer that was inside the reactor such that after a certain time, the impurity factor of the monomer that reacts and turns into polymer changes from the initial $i_{\text{p}} = 0.8$ to the final $i_{\text{p}} = 1.0$. In order to take this effect into account, the model of the plant (8.1a–8.1e) is augmented by the following differential equation:

$$\frac{\mathrm{d}i_p}{\mathrm{d}t} = \frac{\dot{m}_{\text{M}}^{\text{in}}(i_{\text{p}}^{\text{in}}\frac{K_i^{\text{in}}+\dot{m}_{\text{M}}^{\text{in}}}{\dot{m}_{\text{M}}^{\text{in}}} - i_{\text{p}}) - K_i^{\text{in}} i_{\text{p}}}{m_{\text{M}} + m_{\text{P}} + m_{\text{W}}}, \tag{8.18}$$

where i_{p} is the impurity factor of the monomer that actually reacts (see Eq. (8.3)) and determines the heat produced by the reaction, and i_{p}^{in} is the impurity factor of the

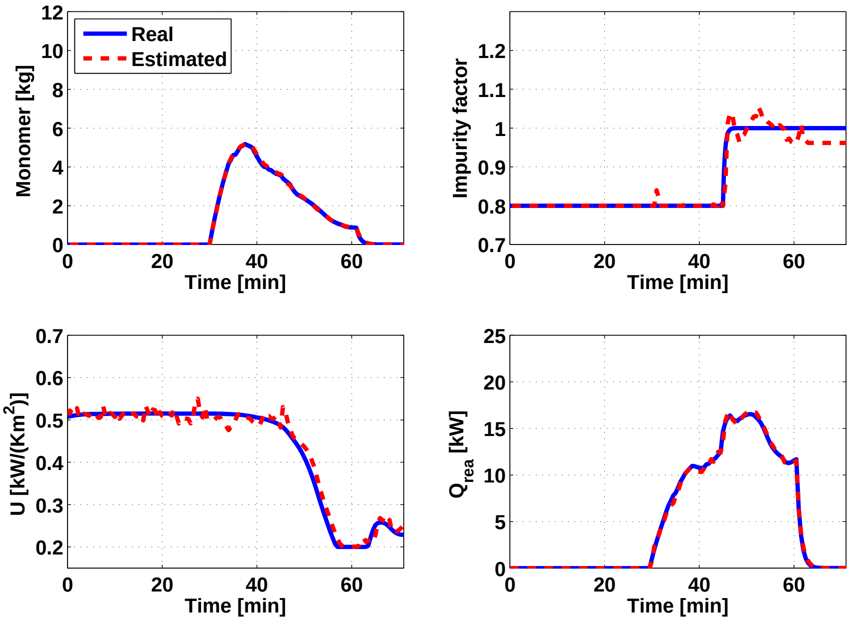

Figure 8.12: Performance of the Extended Kalman Filter estimation for one batch using the multi-stage NMPC as a controller

monomer input. The constant K_i^{in} controls the speed at which the monomer that is reacting goes from the initial impurity factor ($i_{\mathrm{p}} = 0.8$) to the final impurity factor ($i_{\mathrm{p}} = 1.0$). We choose $K_{\mathrm{p}}^{\mathrm{in}} = 3$ so that this change takes around 3 minutes in the case shown in Fig. 8.12. In Fig. 8.13, different simulation results are presented in order to compare the performance of the proposed robust multi-stage NMPC with standard NMPC with estimation of the reaction rate.

Fig. 8.13.a shows the performance of standard NMPC assuming noise-free full-state feedback and perfect knowledge of the uncertainty. As it can be seen, due to the almost step-shaped disturbance in the impurity factor, constraint violations occur even though it is assumed that the value of the impurity factor is exactly known at the next sampling time. A similar behavior can be seen in Fig. 8.13.c where the standard NMPC controller with noisy temperature measurements and the EKF is simulated. In contrast, multi-stage NMPC handles the step disturbance nicely, avoiding constraint violations both for full-state feedback (Fig. 8.13.b) and when only noisy temperature measurements are available (Fig. 8.13.d). Note that in this case no information about the current value of the uncertainty is included in multi-stage NMPC. This means that it would not be even necessary to estimate it. These results show that the lack of robustness against a

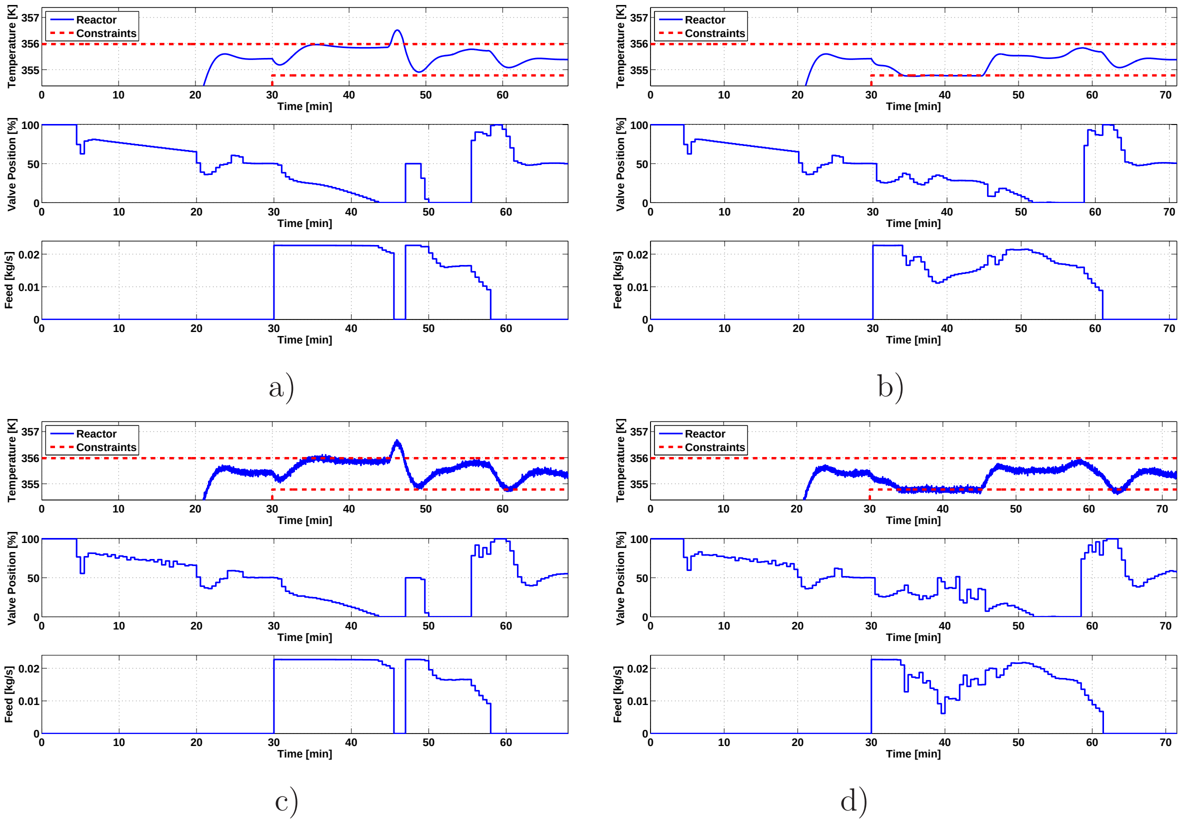

Figure 8.13: Simulation of standard NMPC with full-state feedback and perfect knowledge about the uncertainty (a), and with the EKF (c), and simulation of robust NMPC with full-state feedback (b), and with the EKF (d). In all cases a step disturbance in the impurity factor of the monomer input enters at time $t = 45$ min from $i_{\mathrm{p}}^{\mathrm{in}} = 0.8$ to $i_{\mathrm{p}}^{\mathrm{in}} = 1.0$.

sudden change of the uncertainty of standard NMPC is not due to the estimation error or the measurement noise, since it is observed also in the case of full-state feedback and perfect knowledge of the uncertainty, but it is inherent to the approach, because it does not take into account that the uncertainty may vary. On the other hand, multi-stage NMPC can deal with constant uncertainties as well as with sudden changes of the uncertain parameter because of the anticipation of the uncertainty in the scenario tree. The price for this robustness is a slightly longer feeding time (around 10% longer in this case).

If the uncertainty can be estimated, it is reasonable to use this information to update the weights of the branches in the scenario tree according to the estimated value of the uncertainty. In this manner, the cost function will be minimized taking into account that the estimated value is the most probable one, but at the same time it is guaranteed that the closed-loop will be still robust for all the possible values of the uncertainty. Fig. 8.14 shows a comparison of multi-stage NMPC and multi-stage NMPC enhanced by the estimation of i_p for the case where the impurity is $i_\mathrm{p} = 1.0$. For multi-stage NMPC the uncertainty is not estimated and therefore the probability of each scenario is equal ($\omega_1 = 1/3$, $\omega_2 = 1/3$, $\omega_3 = 1/3$), while for the enhanced multi-stage NMPC the value of the impurity factor in the middle branch ($i_\mathrm{p} = 1.0$) of the scenario tree is replaced by the estimated impurity with a high probability ($\omega_\mathrm{est} = 0.99$). The enhanced multi-stage NMPC improves the performance of the controller, the feeding time is reduced by around 5% while maintaining the robustness of the approach.

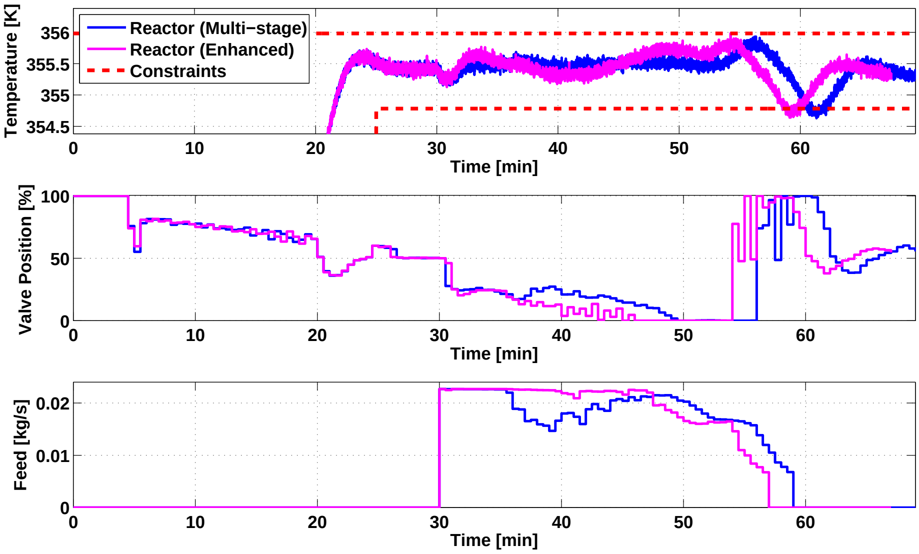

Figure 8.14: Simulation of multi-stage NMPC and enhanced multi-stage NMPC for the case where the uncertain parameter and the states are estimated using an EKF.

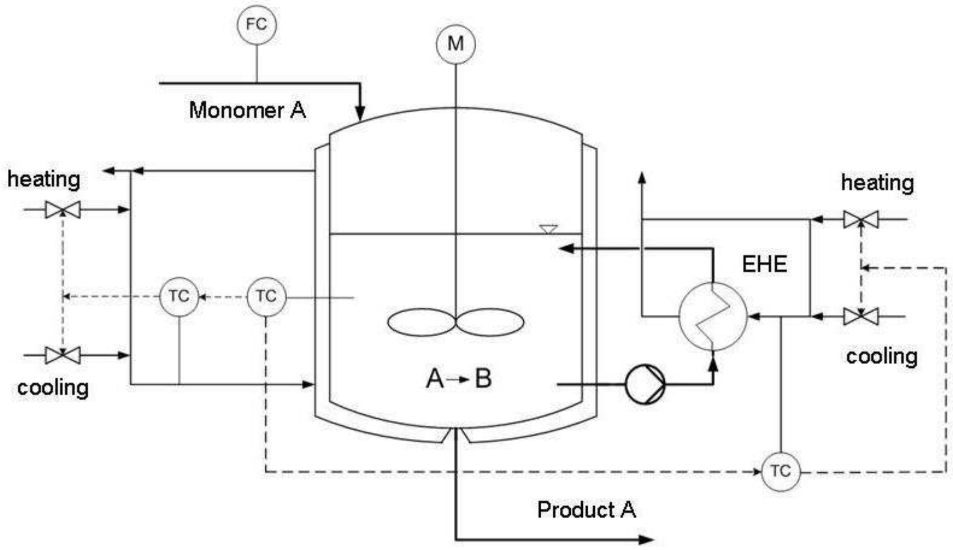

Figure 8.15: Industrial batch polymerization reactor with an external heat exchanger.

8.2 An Industrial Polymerization Reactor

This section shows a comparison of multi-stage NMPC with other robust NMPC strategies applied to an industrial batch polymerization reactor and it is based on the results published in (Lucia et al., 2014b). In addition, it is demonstrated that the efficient implementation with CasADi using automatic differentiation as described in Chapter 5 makes it possible to solve the problem in real-time for much longer horizons and more uncertain parameters when compared to the TOMLAB/MATLAB implementation used for the example in the previous section.

Description of the Model

The results of this section were obtained for a realistic industrial batch polymerization reactor model provided by BASF SE as a case study in the context of the EU-funded project EMBOCON (EMBOCON, 2014). A scheme of the system under consideration can be seen in Fig. 8.15. The system consists of a reactor into which monomer is fed. The monomer turns into a polymer in a strongly exothermic chemical reaction. The reactor is equipped with a jacket and with an External Heat Exchanger (EHE) that can both be used to control the temperature inside the reactor.

The process is modeled by a set of 8 ordinary differential equations:

$$\dot{m}_{\mathrm{W}} = \dot{m}_{\mathrm{F}}\, \omega_{\mathrm{W,F}}, \tag{8.19a}$$

$$\dot{m}_{\mathrm{A}} = \dot{m}_{\mathrm{F}}\omega_{\mathrm{A,F}} - k_{\mathrm{R1}}\, m_{\mathrm{A,R}} - k_{\mathrm{R2}}\, m_{\mathrm{AWT}}\, m_{\mathrm{A}}/m_{\mathrm{ges}}, \tag{8.19b}$$

$$\dot{m}_{\mathrm{P}} = k_{\mathrm{R1}}\, m_{\mathrm{A,R}} + k_{\mathrm{R2}}\, m_{\mathrm{AWT}}\, m_{\mathrm{A}}/m_{\mathrm{ges}}, \tag{8.19c}$$

$$\dot{T}_{\mathrm{R}} = 1/(c_{\mathrm{p,R}} m_{\mathrm{ges}})\, [\dot{m}_{\mathrm{F}}\, c_{\mathrm{p,F}}\, (T_{\mathrm{F}} - T_{\mathrm{R}}) + \Delta H_{\mathrm{R}} k_{\mathrm{R1}} m_{\mathrm{A,R}} \tag{8.19d}$$

$$- k_{\mathrm{K}} A\, (T_{\mathrm{R}} - T_{\mathrm{S}}) - \dot{m}_{\mathrm{AWT}}\, c_{\mathrm{p,R}}\, (T_{\mathrm{R}} - T_{\mathrm{EK}})], \tag{8.19e}$$

$$\dot{T}_{S} = 1/(c_{\mathrm{p,S}} m_{\mathrm{S}})\, [k_{\mathrm{K}} A\, (T_{\mathrm{R}} - T_{\mathrm{S}}) - k_{\mathrm{K}} A\, (T_{\mathrm{S}} - T_{\mathrm{M}})], \tag{8.19f}$$

$$\dot{T}_{\mathrm{M}} = 1/(c_{\mathrm{p,W}} m_{\mathrm{M,KW}})\, [\dot{m}_{\mathrm{M,KW}}\, c_{\mathrm{p,W}} \left(T_{\mathrm{M}}^{\mathrm{IN}} - T_{\mathrm{M}}\right) \tag{8.19g}$$

$$+ k_{\mathrm{K}} A\, (T_{\mathrm{S}} - T_{\mathrm{M}})], \tag{8.19h}$$

$$\dot{T}_{\mathrm{EK}} = 1/(c_{\mathrm{p,R}} m_{\mathrm{AWT}})\, [\dot{m}_{\mathrm{AWT}} c_{\mathrm{p,W}}\, (T_{\mathrm{R}} - T_{\mathrm{EK}}) \tag{8.19i}$$

$$- \alpha\, (T_{\mathrm{EK}} - T_{\mathrm{AWT}}) + k_{\mathrm{R2}}\, m_{\mathrm{A}}\, m_{\mathrm{AWT}} \Delta H_{\mathrm{R}}/m_{\mathrm{ges}}], \tag{8.19j}$$

$$\dot{T}_{\mathrm{AWT}} = [\dot{m}_{\mathrm{AWT,KW}}\, c_{\mathrm{p,W}}\, (T_{\mathrm{AWT}}^{\mathrm{IN}} - T_{\mathrm{AWT}}) \tag{8.19k}$$

$$- \alpha\, (T_{\mathrm{AWT}} - T_{\mathrm{EK}})]/(c_{\mathrm{p,W}} m_{\mathrm{AWT,KW}}), \tag{8.19l}$$

where: (8.19m)

$$U = m_{\mathrm{P}}/(m_{\mathrm{A}} + m_{\mathrm{P}}), \tag{8.19n}$$

$$m_{\mathrm{ges}} = m_{\mathrm{W}} + m_{\mathrm{A}} + m_{\mathrm{P}}, \tag{8.19o}$$

$$k_{\mathrm{R1}} = k_0 e^{\frac{-E_a}{R(T_{\mathrm{R}}+273.15)}} \left(k_{\mathrm{U1}}\,(1-U) + k_{\mathrm{U2}} U\right), \tag{8.19p}$$

$$k_{\mathrm{R2}} = k_0 e^{\frac{-E_a}{R(T_{\mathrm{EK}}+273.15)}} \left(k_{\mathrm{U1}}\,(1-U) + k_{\mathrm{U2}} U\right), \tag{8.19q}$$

$$k_{\mathrm{K}} = (m_{\mathrm{W}}\, k_{\mathrm{WS}} + m_{\mathrm{A}}\, k_{\mathrm{AS}} + m_{\mathrm{P}}\, k_{\mathrm{PS}})/m_{\mathrm{ges}}, \tag{8.19r}$$

$$m_{\mathrm{A,R}} = m_{\mathrm{A}} - m_{\mathrm{A}} m_{\mathrm{AWT}}/m_{\mathrm{ges}}. \tag{8.19s}$$

The model includes mass balances for the water, monomer and product hold-ups (m_{W}, m_{A}, m_{P}) and energy balances for the temperatures of the reactor (T_{R}), of the vessel (T_{S}), of the jacket (T_{M}), of the mixture in the external heat exchanger (T_{EK}) and of the coolant leaving the external heat exchanger (T_{AWT}). The variable U denotes the polymer-monomer ratio in the reactor, m_{ges} represents the total mass, k_{R1} is the reaction rate inside the reactor and k_{R2} is the reaction rate in the external heat exchanger. The overall heat transfer coefficient of the mixture inside the reactor is denoted as k_{K} and $m_{\mathrm{A,R}}$ represents the current amount of monomer inside the reactor.

The available control inputs are the feed flow $\dot{m}_{\mathrm{F}}$, the coolant temperature at the inlet of the jacket $T_{\mathrm{M}}^{\mathrm{IN}}$ and the coolant temperature at the inlet of the external heat exchanger

$T_{\mathrm{AWT}}^{\mathrm{IN}}$.

The complete set of parameters of the model used in this work as provided by BASF SE, together with a short description of each parameter, is presented in Table 8.8.

Description of the Control Problem

The control task under consideration is the production of one batch of polymer in the minimum possible time while satisfying safety constraints and constraints on the quality of the resulting product by manipulating the control inputs $\dot{m}_{\mathrm{F}}$, $T_{\mathrm{M}}^{\mathrm{IN}}$ and $T_{\mathrm{AWT}}^{\mathrm{IN}}$.

The temperature at which the polymerization reaction takes place strongly influences the properties of the resulting polymer. For this reason, the temperature of the reactor should be maintained in a range of ± 2.0°C around the desired reaction temperature $T_{\mathrm{set}} = 90$°C in order to ensure that the produced polymer has the required properties.

Real processes are also subject to important safety constraints that are incorporated to account for possible failures of the equipment. In this case, the maximum temperature that the reactor would reach in the case of a cooling failure is constrained to be below 109°C. The temperature that the reactor would achieve in the case of a complete cooling failure can be calculated as:

$$T_{\mathrm{adiab}} = \frac{\Delta H_{\mathrm{R}}}{c_{\mathrm{p,R}}} \frac{m_{\mathrm{A}}}{m_{\mathrm{ges}}} + T_{\mathrm{R}}. \tag{8.20}$$

To model the safety constraint, the model in (8.19a) is extended by an additional differential state (T_{adiab}), the differential equation of which is obtained by differentiating (8.20) with respect to time:

$$\dot{T}_{\mathrm{adiab}} = \frac{\Delta H_{\mathrm{R}}}{m_{\mathrm{ges}} c_{\mathrm{p,R}}} \dot{m}_{\mathrm{A}} - (\dot{m}_{\mathrm{W}} + \dot{m}_{\mathrm{A}} + \dot{m}_{\mathrm{P}}) \left(\frac{m_{\mathrm{A}} \Delta H_{\mathrm{R}}}{m_{\mathrm{ges}}^2 c_{\mathrm{p,R}}} \right) + \dot{T}_{\mathrm{R}}. \tag{8.21}$$

The new state is constrained to be below the safe temperature ($T_{\mathrm{adiab}} \leq 109$°C). Instead of adding a new state it is also possible to consider the nonlinear constraint directly.

The maximum amount of material that can be fed into the reactor is $\int \dot{m}_{\mathrm{F}} \, \mathrm{d}t = 30000$ kg. After all the material has been fed into the reactor (feeding phase) the reaction continues with the remaining monomer (holding phase) and the batch is considered to be

Table 8.8: Parameters of the industrial polymerization reactor model

Parameter	Description	Value	Unit
R	Gas constant	8.314	$\frac{\text{kJ}}{\text{kmol}\cdot\text{K}}$
$c_{\text{p,W}}$	Specific heat capacity of the coolant	4.2	$\frac{\text{kJ}}{\text{kg}\cdot\text{K}}$
$c_{\text{p,S}}$	Specific heat capacity of the steel	0.47	$\frac{\text{kJ}}{\text{kg}\cdot\text{K}}$
$c_{\text{p,F}}$	Specific heat capacity of the feed	3	$\frac{\text{kJ}}{\text{kg}\cdot\text{K}}$
$c_{\text{p,R}}$	Specific heat capacity of the reactor contents	5.0	$\frac{\text{kJ}}{\text{kg}\cdot\text{K}}$
k_{WS}	Heat transfer coeff. water-steel	4800	$\frac{\text{W}}{\text{m}^2\cdot\text{K}}$
T_{F}	Feed temperature	25	°C
A	Heat exchange surface of the jacket	65	m^2
$m_{\text{M,KW}}$	Mass of coolant in the jacket	5000	kg
m_{S}	Mass of reactor steel	39000	kg
m_{AWT}	Mass of the product in the EHE	200	kg
$m_{\text{AWT,KW}}$	Mass of the coolant in the EHE	1000	kg
$\dot{m}_{\text{M,KW}}$	Coolant flow of the jacket	300000	$\frac{\text{kg}}{\text{h}}$
$\dot{m}_{\text{AWT,KW}}$	Coolant flow of the EHE	100000	$\frac{\text{kg}}{\text{h}}$
$\dot{m}_{\text{AWT}}$	Product flow to the EHE	20000	$\frac{\text{kg}}{\text{h}}$
E_{a}	Activation energy	8500	$\frac{\text{kJ}}{\text{kmol}}$
ΔH_{R}	Specific reaction enthalpy	950	$\frac{\text{kJ}}{\text{kg}}$
k_0	Specific reaction rate	7	$\frac{1}{s}$
k_{U2}	Reaction parameter 1	32	–
k_{U1}	Reaction parameter 2	4	–
$w_{\text{W,F}}$	Mass fraction of water in the feed	0.333	–
$w_{\text{A,F}}$	Mass fraction of A in the feed	0.667	–
k_{AS}	Heat transfer coeff. monomer-steel	1000	$\frac{\text{W}}{\text{m}^2\cdot\text{K}}$
k_{PS}	Heat transfer coeff. product-steel	100	$\frac{\text{W}}{\text{m}^2\cdot\text{K}}$
α	Experimental coefficient	3600000	$\frac{1}{\text{s}}$

finished when the desired amount of polymer is produced ($m_{\mathrm{P}}^{\mathrm{end}} = 20680$ kg). In order to avoid the switching between different optimization problems as in the previous section for the feeding and the holding phase, an additional state is added that accounts for the accumulated material that has been fed, that is, $\dot{m}_{\mathrm{A}}^{\mathrm{acc}} = \dot{m}_{\mathrm{F}}$ and another constraint is included such that $0 < m_{\mathrm{F}}^{\mathrm{acc}} < m_{\mathrm{F}}^{\max} = 30000$ kg. The constraints on the states and control inputs, as well as the initial conditions can be seen in Table 8.9 and in Table 8.10.

Table 8.9: Initial conditions and state constraints.

State	Init. cond.	Min.	Max.	Unit
m_{W}	10000	0	inf.	kg
m_{A}	853	0	inf.	kg
m_{P}	26.5	0	inf.	kg
T_{R}	90.0	$T_{\mathrm{set}} - 2.0$	$T_{\mathrm{set}} + 2.0$	°C
T_{S}	90.0	0	100	°C
T_{M}	90.0	0	100	°C
T_{EK}	35.0	0	100	°C
T_{AWT}	35.0	0	100	°C
T_{adiab}	104.897	0	109	°C
$m_{\mathrm{F}}^{\mathrm{acc}}$	0	0	30000	kg

Table 8.10: Bounds on the manipulated variables.

Control	Min.	Max.	Unit
$\dot{m}_{\mathrm{F}}$	0	30000	$\frac{\mathrm{kg}}{\mathrm{h}}$
$T_{\mathrm{M}}^{\mathrm{IN}}$	60	100	°C
$T_{\mathrm{AWT}}^{\mathrm{IN}}$	60	100	°C

In a real system, usually the model parameters cannot be determined exactly, what represents an important source of uncertainty. For this example two of the most critical parameters of the model are considered not to be precisely known and vary with respect to their nominal value. In particular, it is assumed that the specific reaction enthalpy

ΔH_{R} and the specific reaction rate k_0 are constant but uncertain, having values that can vary ±30% with respect to their nominal values reported in Table 8.8.

Mathematical Formulation

The control task is formulated as the following optimization problem:

$$\min_{x_k^j, u_k^j \forall (j,k) \in I} \quad J_{\mathrm{batch}}(x_{k+1}^j, u_k^j), \tag{8.22a}$$

subject to:

$$\begin{aligned}
x_{k+1}^j &= f(x_k^{p(j)}, u_k^j, d_k^{r(j)}), & \forall\, (j, k+1) \in I, && \text{(8.22b)}\\
T_{\mathrm{set}} - 2.0 \leq T_{\mathrm{R},k}^j &\leq T_{\mathrm{set}} + 2.0, & \forall\, (j,k) \in I, && \text{(8.22c)}\\
0 \leq T_{\mathrm{S},k}^j &\leq 100, & \forall\, (j,k) \in I, && \text{(8.22d)}\\
0 \leq T_{\mathrm{M},k}^j &\leq 100, & \forall\, (j,k) \in I, && \text{(8.22e)}\\
0 \leq T_{\mathrm{EK},k}^j &\leq 100, & \forall\, (j,k) \in I, && \text{(8.22f)}\\
0 \leq T_{\mathrm{AWT},k}^j &\leq 100, & \forall\, (j,k) \in I, && \text{(8.22g)}\\
0 \leq T_{\mathrm{adiab},k}^j &\leq 109, & \forall\, (j,k) \in I, && \text{(8.22h)}\\
0 \leq m_{\mathrm{A},k}^{\mathrm{acc},j} &\leq m_A^{\max}, & \forall\, (j,k) \in I, && \text{(8.22i)}\\
0 \leq \dot{m}_{\mathrm{F},k}^j &\leq 30000, & \forall\, (j,k) \in I, && \text{(8.22j)}\\
60 \leq T_{\mathrm{M},k}^{\mathrm{IN},j} &\leq 100, & \forall\, (j,k) \in I, && \text{(8.22k)}\\
60 \leq T_{\mathrm{AWT},k}^{\mathrm{IN},j} &\leq 100, & \forall\, (j,k) \in I, && \text{(8.22l)}\\
u_k^j &= u_k^l \text{ if } x_k^{p(j)} = x_k^{p(l)} & \forall\, (j,k), (l,k) \in I, && \text{(8.22m)}
\end{aligned}$$

where the constraints are applied to all the states and all control inputs along each scenario with

$$x_k^j = [m_{\mathrm{W},k}^j, m_{\mathrm{A},k}^j, m_{\mathrm{P},k}^j, T_{\mathrm{R},k}^j, T_{\mathrm{S},k}^j, T_{\mathrm{M},k}^j, T_{\mathrm{EK},k}^j, T_{\mathrm{AWT},k}^j, T_{\mathrm{adiab},k}^j, m_{\mathrm{A},k}^{\mathrm{acc},j}]^{\mathrm{T}},$$

$$u_k^j = [\dot{m}_{\mathrm{F},k}^j, T_{\mathrm{M},k}^{\mathrm{IN},j}, T_{\mathrm{AWT},k}^{\mathrm{IN},j}]^{\mathrm{T}}.$$

The discretized dynamics of the system are included as constraints in (8.22b), (8.22c-8.22i) denote the constraints on the states, (8.22j-8.22l) represent the constraints on the input variables, and the non-ancitipativity constraints are included in (8.22m).

Two possible cost functions $J_{\mathrm{batch}}(x_{k+1}^j, u_k^j)$ are proposed. The first one, (8.23), represents the maximization of the mass of polymer together with a set-point tracking term for the reactor temperature. In addition, small regularization terms are added in order to penalize the control movements so that a smooth control input is achieved as a result of the optimization problem. The mixed tracking cost function results as:

$$\begin{aligned} J_{\text{track}}(x^j_{k+1}, u^j_k) = \sum_{i=1}^{N} \omega_i \sum_{k=0}^{K-1} & -m^j_{\text{P},k+1} + q(T^j_{\text{R},k+1} - T_{\text{set}})^2 \\ & + r_1(\Delta \dot{m}^j_{\text{F},k})^2 + r_2(\Delta T^{\text{IN},j}_{\text{M},k})^2 + r_3(\Delta T^{\text{IN},j}_{\text{AWT},k})^2 \quad , \\ & \forall\ m^j_{\text{P},k+1}, T^j_{\text{R},k+1}, \dot{m}^j_{\text{F},k}, T^{\text{IN},j}_{\text{M},k}, T^{\text{IN},j}_{\text{AWT},k} \in S_i, \end{aligned} \tag{8.23}$$

where q, r_1, r_2 and r_3 are tuning parameters. A different possibility is to avoid the use of a tracking term and to use only an economically motivated cost function:

$$\begin{aligned} J_{\text{eco}}(x^j_{k+1}, u^j_k) = \sum_{i=1}^{N} \omega_i \sum_{k=0}^{N_p-1} & -m^j_{\text{P},k+1} \\ & + r_1(\Delta \dot{m}^j_{\text{F},k})^2 + r_2(\Delta T^{\text{IN},j}_{\text{M},k})^2 + r_3(\Delta T^{\text{IN},j}_{\text{AWT},k})^2 \quad , \\ & \forall\ m^j_{\text{P},k+1}, \dot{m}^j_{\text{F},k}, T^{\text{IN},j}_{\text{M},k}, T^{\text{IN},j}_{\text{AWT},k} \in S_i. \end{aligned} \tag{8.24}$$

For both cases the cost is calculated as the sum over all the N scenarios S_i with $i = 1, ..., N$ along the prediction horizon N_P. Here it is implicitly assumed that the batch time is minimized by maximizing the amount of polymer produced within the prediction horizon. For the multi-stage NMPC implementation a scenario tree with 9 scenarios is considered that results from the combination of the maximum, the minimum and the nominal values of the uncertain parameters. The tree is branched only in the first stage ($N_R = 1$) because as seen in the previous section, a larger robust horizon results in a similar performance with a higher computational cost. For all the results shown in the remainder of the section the sampling time of the NMPC controller is $t_{\text{step}} = 50$ s with a prediction horizon of $N_P = 20$ steps. The tuning parameters chosen for the cost functions presented in (8.23) and (8.24) are $r_1 = 0.1$, $r_2 = 0.02$, $r_3 = 0.01$ and $q = 10000$.

As explained in Chapter 5, all the algorithms presented in this section have been implemented using the optimization tool CasADi to pass first and second order derivative information, which has been automatically generated, to IPOPT. This makes it possible to greatly improve the computational speed that was reported in the previous section and in the publication (Lucia et al., 2013a) with little implementation effort. The real plant is simulated with the calculated control input using the integrators from the SUNDIALS toolbox with a high accuracy. For comparison purposes, results with a collocation approach and with a multiple shooting approach are presented. For the collocation approach Radau collocation points are used, with interpolating polynomials of degree 2. For the multiple shooting approach the integrators from the SUNDIALS toolbox are used.

8.2.1 Tracking NMPC vs. Multi-stage Economic NMPC of a Polymerization Reactor under Uncertainty

This section presents first a comparison of tracking and economic NMPC without uncertainty and afterwards a comparison of standard tracking NMPC (with typical modifications to account for uncertainty) and multi-stage NMPC under uncertainty.

First it is assumed that the model is perfect, i.e. there is no plant-model mismatch. Then the described control problem can be solved by standard NMPC, using either the economic cost function $J_{\text{eco}}(x_{k+1}^j, u_k^j)$ or the mixed cost function $J_{\text{track}}(x_{k+1}^j, u_k^j)$ in which the center of the allowed temperature range is tracked. As expected, the use of economic NMPC improves the performance of the controller as can be seen in Fig. 8.16. The batch time is reduced by around 7.5%. Fig. 8.16 also shows a comparison between the solution obtained using collocation and multiple shooting. Since both results are very similar, only the results with the collocation approach are shown in the remainder of the section . The current multiple-shooting implementation has not been optimized for its performance and therefore the computation times are much higher than the ones obtained by the collocation approach (73.23 s vs. 0.072 s per optimization problem for standard NMPC with a mixed tracking objective).

As can be seen in Fig. 8.16 and in general when using an economic cost function, the process is typically operated at one of its constraints, and this constraint may vary along the operation. Since models for industrial applications are imperfect, constraints violations will be unavoidable, unless additional measures are taken. A typical approach used in practice is to track a conservative set-point and to expect that the tracking controller is able to maintain the system near the set-point despite the several uncertainties affecting it. This is however not enough to deal with significant uncertainties. Fig. 8.17 (left) shows the results of standard NMPC for different values of the uncertain parameters ΔH_{R} and k_0. Each line in the plot represents the state and control trajectories with different values of the uncertain parameter for each batch (varying from batch to batch between ±30% with respect to their nominal values) using the same controller. The parameters are kept constant along each batch. It is clear that the standard NMPC controller with tracking

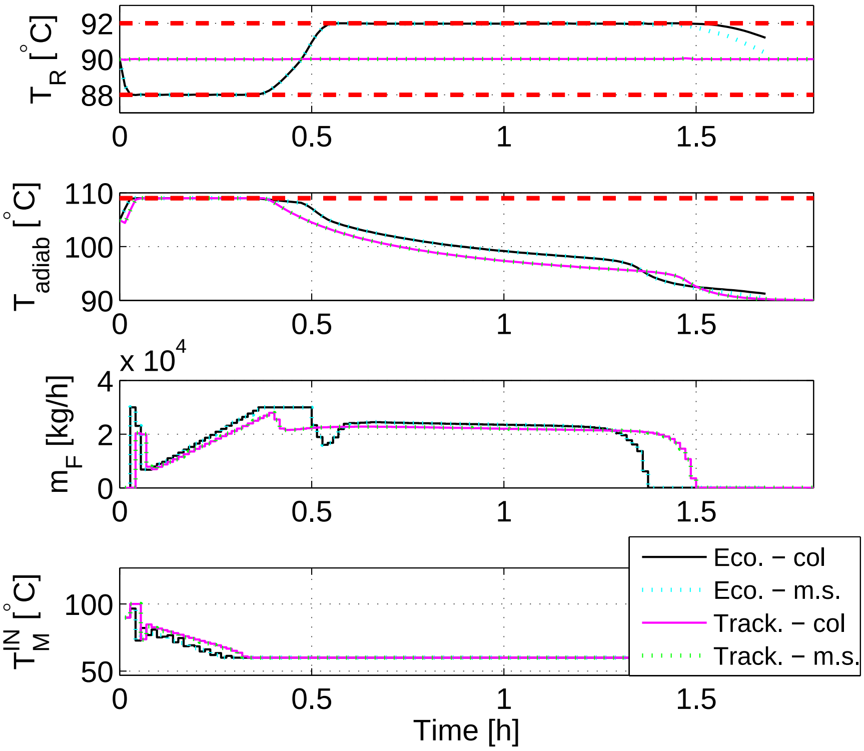

Figure 8.16: Reactor temperature, safety temperature (with constraints indicated), monomer feed and jacket temperature for mixed and for economic NMPC using collocation and multiple shooting for a perfect model (no uncertainty).

of the mean value of the allowed temperature range fails to satisfy the constraints for several scenarios.

In order to increase the amount of feedback used in the standard NMPC scheme, a simple modification which introduces a bias-term for the set-point can be used. By doing this, the set-point that is used in the optimizer at each sampling time ($T_{\text{set}}^{\text{opt}}$) is updated using a proportional rule, i.e., $T_{\text{set}}^{\text{opt}} \leftarrow T_{\text{set}}^{\text{opt}} + K(T_{\text{set}} - T_{\text{R}})$, with $K = 0.015$ where T_{set} is the real setpoint and T_{R} is the state of the real plant. The performance of the controller is improved (see Fig. 8.17 (right)) but there still are important violations of the safety constraints and for the reactor temperature that could lead to a deficient quality of the product and are not tolerable for safety reasons.

If the results are carefully analyzed, the violations occur only for certain values of the un-

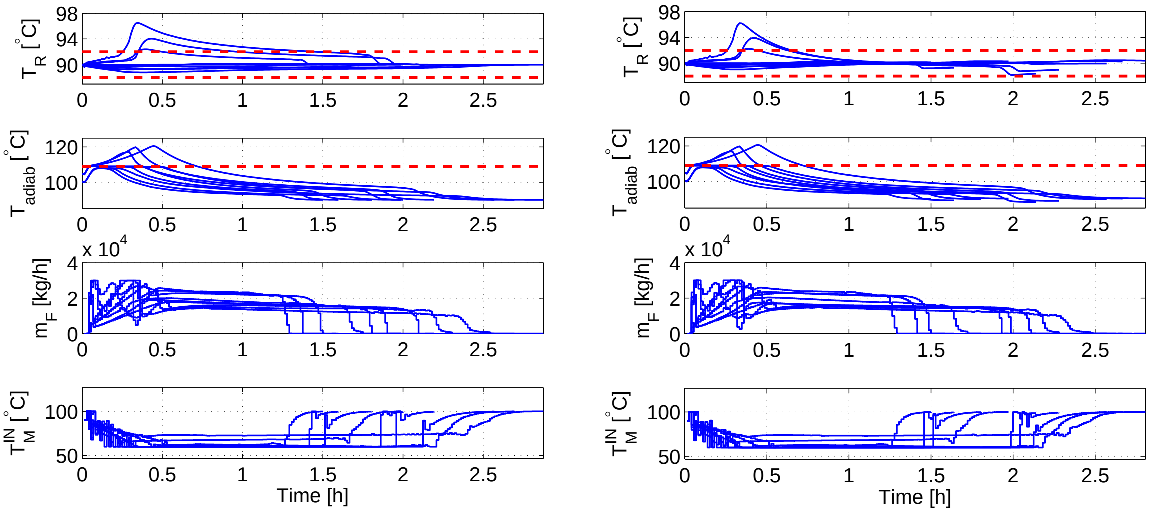

Figure 8.17: Reactor temperature, safety temperature (with constraints indicated), monomer feed and jacket temperature for standard NMPC (left) and for standard NMPC with bias term (right) with a mixed tracking cost function.

certain parameters, normally when the real ΔH_{R} and k_0 values are higher than expected, because then more heat than expected is generated, leading to constraint violations. A possible way to deal with this problem with the standard NMPC scheme with a tracking cost function is to consider the worst-case value of the uncertain parameters, i.e $\Delta H_{\mathrm{R}} = 1.30 \cdot \Delta H_{\mathrm{R}}$ and $k_0 = 1.30 \cdot k_0$ in the prediction model. With this conservative choice of the uncertain parameters, standard NMPC with a mixed tracking cost function and a bias-term is able to satisfy the constraints for all cases of the uncertainty as can be seen in Fig. 8.18 (left). However, the obtained batch times are significantly higher than the ones obtained with the proposed multi-stage NMPC with economic cost function (Fig. 8.18 (right))

A summary of the results obtained with the different controllers for all the scenarios is shown in Table 8.11. It is clear that the use of multi-stage NMPC with a simple scenario tree can improve the performance of the controller significantly compared to a conservative choice of the uncertain parameters. Even in the case of the nominal value of the parameters, multi-stage NMPC achieves a very similar performance compared to the standard NMPC approach, while being robust for all the possible values of the uncertainty. This is due to the fact that an economic cost function is used in the multi-stage case and that the uncertainty is taken into account by using a scenario tree, whereas a tracking

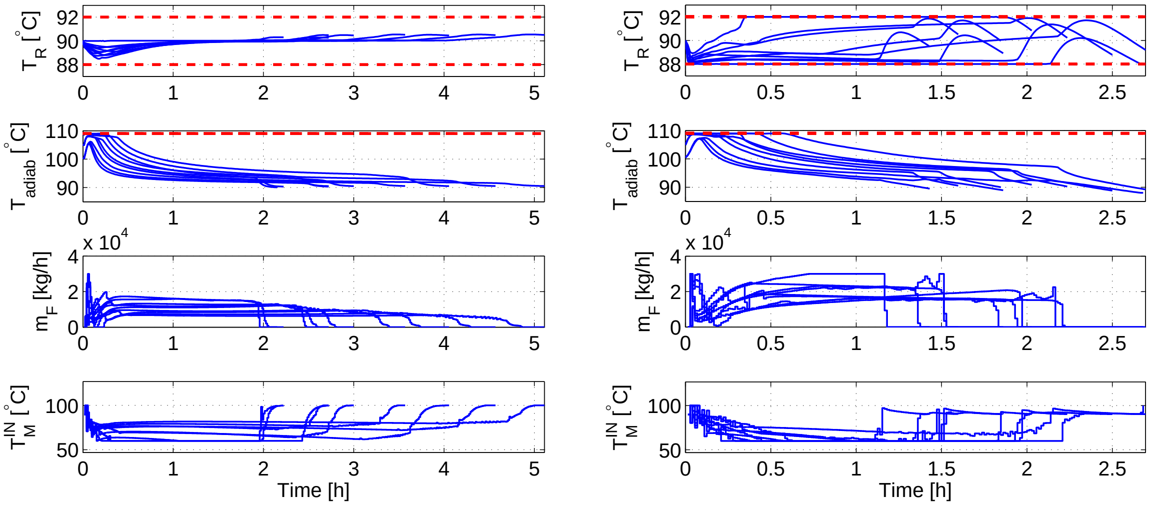

Figure 8.18: Reactor temperature, safety temperature (with constraints indicated), monomer feed and jacket temperature for standard NMPC with bias term and the worst case value of the parameters in the model with mixed tracking cost function (left) and for multi-stage NMPC with an economic cost function (right).

term of a conservative set-point is used in standard NMPC. Note that in this case, the improvement in the batch time averaged over all the scenarios is approximately 60%, which clearly shows the large economic potential of the multi-stage approach compared to a conservative choice of parameters.

This improvement is achieved by considering the uncertainty explicitly using a robust NMPC closed-loop approach by means of a scenario tree, which increases the computational complexity of the approach. However, if efficient tools are used – as the ones described in this thesis – it is possible to solve the resulting optimization problems consistently faster than the sampling time of the system ($t_{\text{step}} = 50$ s) which enables an implementation of the proposed scheme for this industrial case study. The average and maximum computation times per optimization problem obtained for each controller are reported in Table 8.12.

Table 8.11: Performance comparison between standard NMPC, standard NMPC with bias term, standard NMPC with bias term using the worst-case value of the parameters in the model used in the optimizer and multi-stage NMPC.

		Batch time in hours			
Unc. in ΔH_R	**Unc. in** k_0	**Standard NMPC**	**Standard NMPC+bias**	**Std. worst-case NMPC+bias**	**Multi-stage NMPC**
+30%	+30%	infeasible	infeasible	2.15	2.03
+30%	+0%	infeasible	infeasible	2.72	2.24
+30%	-30%	infeasible	infeasible	4.05	2.69
+0%	+30%	1.60	1.64	2.22	1.60
+0%	+0%	1.81	1.81	3.00	1.84
+0%	-30%	2.69	2.67	4.57	2.50
-30%	+30%	1.50	1.50	2.72	1.43
-30%	+0%	1.99	1.97	3.57	1.86
-30%	-30%	2.88	2.80	5.11	2.68
Av. batch time [h]		infeasible	infeasible	3.35	2.10

Table 8.12: Average and maximum computation times per optimization problem (in seconds) of standard NMPC, standard NMPC with bias term, standard NMPC with bias term using the worst-case value of the parameters in the model used in the optimizer and multi-stage NMPC.

	Standard NMPC	**Standard NMPC+bias**	**Std. worst-case NMPC+bias**	**Multi-stage NMPC**
Average [s]	0.072	0.071	0.059	1.134
Maximum [s]	0.230	0.190	0.179	1.550

8.2.2 Comparison of Robust Economic NMPC formulations for a Polymerization Reactor under Uncertainty

This section presents a comparison of different robust NMPC approaches that have been presented in the literature over the last years and that were described in Chapter 3, i.e., multi-stage NMPC, open-loop robust NMPC, robust NMPC with affine constant policies and robust NMPC with affine time-varying policies. Although in this case the stability of the closed-loop system is not guaranteed, no stability or recursive feasibility problems were encountered for any of the simulations presented.

The robustness of the controllers is achieved in all cases by enforcing the constraints on each node of a scenario tree which is obtained by combining the maximum, minimum and nominal values of the uncertain parameters as in the previous section. The same tree and the same economic cost function (8.24) are used for all the controllers.

Fig. 8.19 shows the results obtained when solving the optimization problem (8.24) in the form of a robust open-loop NMPC approach as formulated in (3.7). Since the constraints are checked for all the nodes in the scenario tree, no constraint violations occur for any of the scenarios. However, in the open-loop robust approach, the fact that at the next sampling time a newly computed control input will be able to counteract the effect of the realized uncertainty is ignored. This leads to a higher degree of conservativeness and therefore to longer batch times.

This can be partially compensated if a certain amount of feedback is introduced in the prediction by including a feedback gain as an additional optimization variable. If a robust NMPC controller with affine constant feedback policies is used by formulating the optimization problem as in (3.13) with the nominal case in the cost function, shorter batch times are obtained, as can be seen in Fig. 8.20 (left). If the number of degrees of freedom is increased by optimizing over time-varying state feedback policies (as in (3.14)) the performance can be increased even more, while preserving the robustness of the solution for all the different scenarios at the cost of a higher computation cost. The results for robust NMPC with time-varying feedback policies can be seen in Fig. 8.20 (right).

A summary of the performance of each controller for all the scenarios is given in Table 8.13. It can be seen that the use of any of the approaches presented in this section leads to a significant reduction of the batch times with respect to the ones obtained with standard

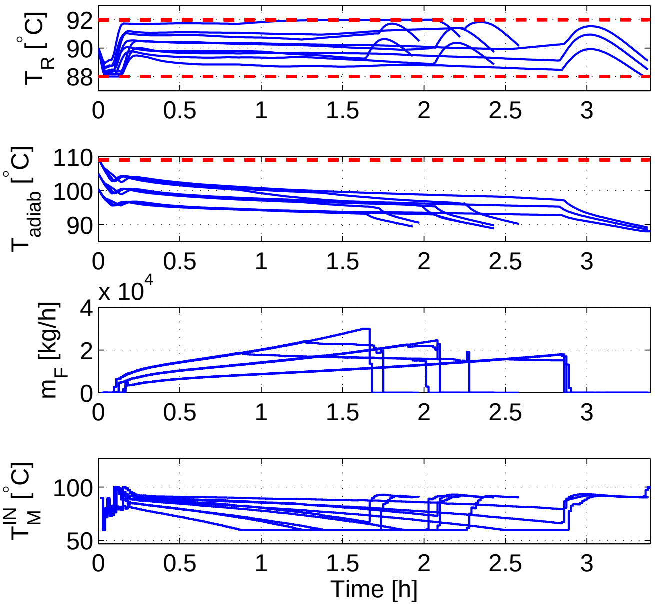

Figure 8.19: Reactor temperature, safety temperature (with constraints indicated), monomer feed and jacket temperature for open-loop robust NMPC with an economic cost function.

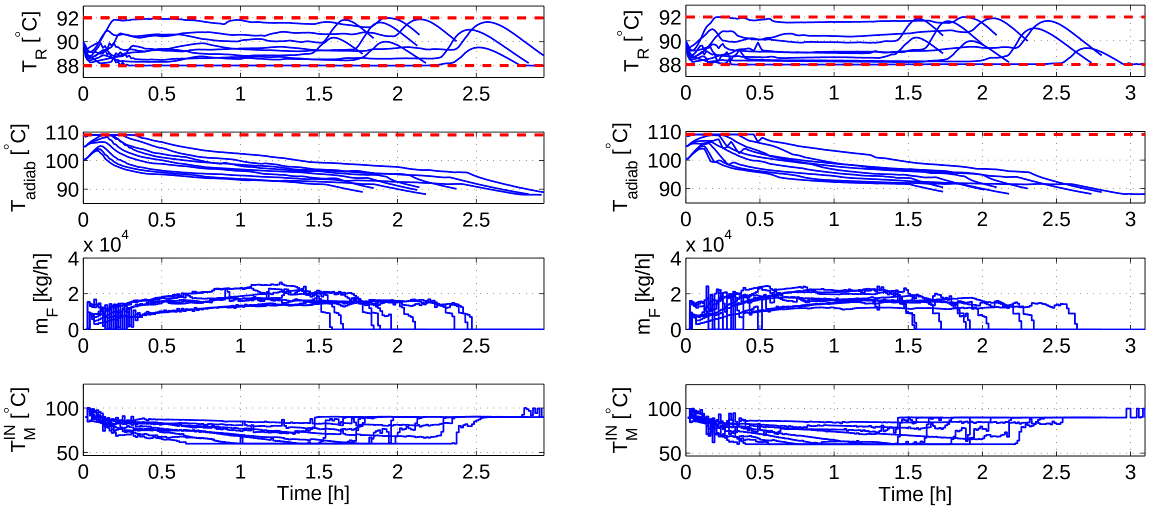

Figure 8.20: Reactor temperature, safety temperature (with constraints indicated), monomer feed and jacket temperature for robust NMPC with constant (left) and time varying (right) affine control policies with an economic cost function.

NMPC and a conservative choice of the uncertain parameters. It is important to note that the open-loop approach leads to significantly higher batch times (25% with respect to multi-stage NMPC) because recourse is not introduced in the predictions of the NMPC controller. The performance can be enhanced by using constant affine or time-varying affine feedback policies but the best performance is still achieved by multi-stage NMPC. Note that even for the nominal case of the uncertainty, which is the only one entering the cost function of the affine controllers, multi-stage NMPC achieves a better performance. This is due to the conservativeness introduced in the system by the fixed structure of the feedback. The computation times needed for the solution of each controller are reported in Table 8.14. The computation time needed by the open-loop approach is very similar to the one needed by the multi-stage even though it has less degrees of freedoms (free control inputs) because the same constraints are included in order to enforce robustness. The introduction of the state feedback matrix K as optimization variable includes new degrees of freedom (see Table 8.15 for a comparison) and many non-zero entries in the Jacobian of the constraints and in the Hessian of the Lagrangian due to the feedback structure, which leads to computation times per iteration that are much higher than the ones obtained for multi-stage NMPC, especially in the case of time-varying feedback policies. Following (Goulart et al., 2006), we use only the nominal case in the cost function for the cases of affine policies. If all the scenarios are included in the cost, the average performance for the robust NMPC with affine policies (both constant and time-varying) is improved, but it is still worse than in the multi-stage NMPC case for all the scenarios and it leads to larger computation times.

8.2.3 Results with Scenario Decomposition

This subsection shows results with the progressive hedging algorithm presented in Chapter 7 applied to the industrial batch polymerization reactor. As tuning parameters of the Algorithm 1 a tolerance $\epsilon = 10^{-5}$ and $\rho_{\text{init}} = 0$ are chosen. The parameter ρ_k^j is increased at each iteration by a factor $\beta = 2$ with a maximum of $\rho_{\max} = 10^4$. A robust horizon

Table 8.13: Performance comparison between open-loop robust NMPC, robust NMPC with affine constant policies, robust NMPC with affine time-varying policies and multi-stage NMPC.

		Batch time in hours			
Unc. in ΔH_R	**Unc. in** k_0	**Open-loop robust NMPC**	**Constant affine policies**	**Time-varying affine policies**	**Multi-stage NMPC**
+30%	+30%	2.22	2.14	2.09	2.03
+30%	+0%	2.58	2.38	2.31	2.24
+30%	-30%	3.38	2.93	2.81	2.69
+0%	+30%	1.97	1.85	1.74	1.60
+0%	+0%	2.42	2.13	2.01	1.84
+0%	-30%	3.38	2.83	2.74	2.50
-30%	+30%	1.93	1.78	1.74	1.43
-30%	+0%	2.43	2.18	2.18	1.86
-30%	-30%	3.39	2.92	3.00	2.68
Av. batch time [h]		2.63	2.34	2.29	2.10

Table 8.14: Average and maximum computation times per optimization problem (in seconds) of open-loop robust NMPC, robust NMPC with affine constant policies, robust NMPC with affine time-varying policies and multi-stage NMPC.

	Open-loop robust NMPC	**Constant affine policies**	**Time-varying affine policies**	**Multi-stage NMPC**
Average [s]	1.113	13.87	45.43	1.134
Maximum [s]	2.540	128.2	182.6	1.550

Table 8.15: Degrees of freedom (free control input variables) available for open-loop robust NMPC, robust NMPC with affine constant policies, robust NMPC with affine time-varying policies and multi-stage NMPC.

	Open-loop robust NMPC	**Constant affine policies**	**Time-varying affine policies**	**Multi-stage NMPC**
D.o.F.	60	90	660	540

$N_R = 1$ is used so that the non-anticipativity constraints are applied only at the first time stage. As shown in Fig. 8.21 the decomposition approach obtains a very similar performance compared to the monolithic multi-stage approach. Minor differences can be observed due to the presence of local optima.

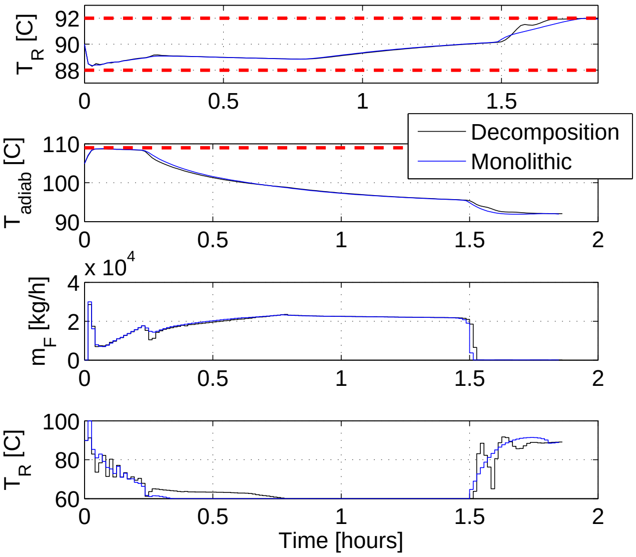

Figure 8.21: Reactor temperature, safety temperature (with constraints indicated), monomer feed and jacket temperature for open-loop robust NMPC with an economic cost function.

Table 8.16 shows the average computation times, the necessary memory to solve the optimization problems and the number of variables of the NLPs resulting from a monolithic formulation of the multistage approach for different number of scenarios. The increasing number of scenarios is obtained by increasing the number of possible realizations of the uncertainty. For example, 9 scenarios are obtained when 3 possible values are considered for each uncertainty and 25 scenarios are obtained when 5 possible values are considered for each uncertainty. If the derivative information can be calculated efficiently and a solver which exploits sparsity such as IPOPT is used, the computation time grows only slightly higher than linear with the number of scenarios. For this reason a decomposition approach could be advantageous only if a large degree of parallelization is available and convergence is achieved in few iterations of the decomposition algorithm. Table 8.17 shows the same information for the same problems solved with the progressive hedging algorithm, including the average number of iterations that the PHA needs to converge to a solution that satisfies the non-anticipativity constraints. The computation times (without parallelization) are significantly higher than in the monolithic case. Even if a complete parallelization is possible (as many computing units as scenarios) the computation time is longer due to the high number of iterations needed for the algorithm to converge. On the other hand, an important feature of the decomposition approach is that the memory necessary to solve the problem does not grow with the size of the scenario tree, since only simple subproblems are solved. If a monolithic approach is used, the memory necessary to solve the problem grows linearly with the size of the problem. For large-scale problems this might be a limitation and therefore a decomposition approach might be the only possible technique to solve the problem.

Table 8.16: Average computation time per optimization problem, memory consumption and number of variables of each optimization problem for an increasing number of scenarios using a monolithic approach.

# Scenarios	Time [s]	Memory [Mb]	# Variables
1	0.1154	35.6	860
4	0.4583	66.7	3431
9	1.1340	141.1	7716
25	3.3541	285.9	21428
49	7.3292	499.3	41996
81	12.972	850.0	69420

The monolithic multi-stage NMPC approach is not limited to the problems of the size reported in Table 8.16. If hardware with more memory is available, the implementation proposed in this thesis is able to solve the large-scale problems resulting from a large scenario tree. For example, it is possible to solve in 440 s the same industrial reactor using a prediction horizon $N_P = 20$ and a robust horizon $N_R = 3$ which results in 729 scenarios and 879,402 variables. Due to the memory limitations this computation has been realized in a computer with 16 Gb of RAM and 2 cores at 2.40 GHz.

Table 8.17: Average computation time per optimization problem (without parallelization), memory consumption, average number of iterations of the progressive hedging algorithm and number of variables of each optimization problem for an increasing number of scenarios using a decomposition approach.

# Scenarios	Time [s]	Memory [Mb]	# Iterations	# Variables
1	0.1154	35.6	-	860
4	16.320	35.6	18.99	860
9	199.98	35.6	47.12	860
25	895.25	35.6	74.22	860
49	1831.1	35.6	93.54	860
81	3690.4	35.6	122.1	860

A possible strategy to reduce the number of iterations that the progressive hedging algorithm needs to converge is to use a hybrid approach between the full decomposition of the problem into single scenarios and the monolithic approach. In this case, the scenario tree is divided into smaller sub-trees, which are called bundles, and the same algorithm is applied until the non-anticipativity constraints are applied.

The systematic choice of the bundles is not trivial, but using knowledge about the problem they can be chosen in such a way that the number of iterations is significantly reduced. It can be observed that the large number of iterations needed is mainly due to the fact that some scenarios are associated to control inputs that drive the system to the constraints. During the iterations of the PHA these control inputs are required to move towards the average control input $\hat{u}_k^j$ so that eventually convergence between all the scenarios is achieved. This movement will not occur if it causes a violation of the constraints. In this case, convergence is slow because all the other control inputs should move towards the one that is at the constraint and cannot be moved.

Following this idea, a good heuristic rule to generate the bundles is to choose for one of them the worst case scenarios, which are more likely to generate the control inputs that make the system lie on the constraints. In this way, those complicating scenarios will be solved simultaneously and the convergence will be faster. Furthermore, the weight of this bundle for the calculation of the average control input $\hat{u}_k^j$ can be increased to enhance even more the performance.

For the results presented here the first bundle is generated using the scenarios with parameter values that are a +30% larger than their the nominal values (first row in Table 8.13), a −30% (ninth row in Table 8.13) and the nominal values of the parameters (fifth row in Table 8.13). The second bundle is composed of the second, third and seventh rows in Table 8.13. The third bundle contains the rest of the scenarios.

Fig. 8.22 shows how the use of bundles reduces significantly the number of iterations that the algorithm needs to converge to a solution that satisfies the non-anticipativity constraints. If the control input of the first bundle is weighted with a factor 50 times bigger than the second and third bundles for the calculation of the average control input $\hat{u}_k^j$, the number of iterations can be further reduced as shown in Fig. 8.22.

The use of bundles can be useful to achieve a compromise with lower memory usage than the monolithic approach but with significantly better computation times due to the decrease of iterations (see Table 8.18). Nevertheless, even assuming that it is possible to perform all the optimization problems in parallel, the monolithic approach achieves lower computation times.

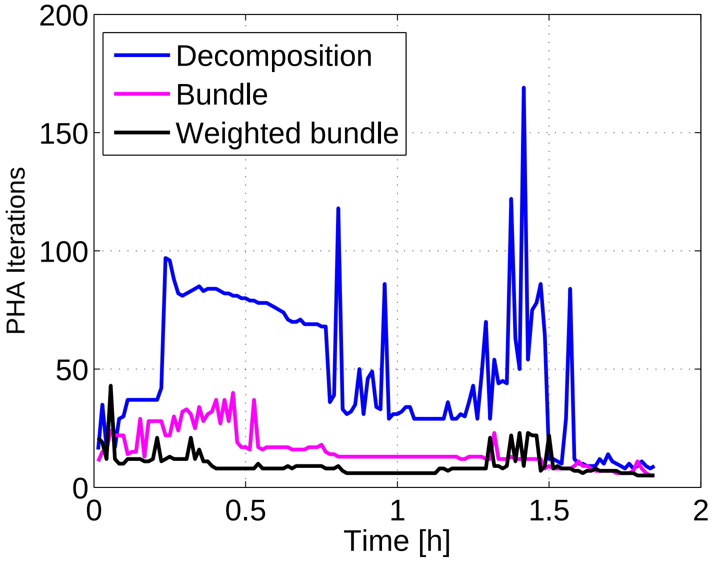

Figure 8.22: Number of iterations of the progressive hedging algorithm at each NMPC iteration for the full decomposition, the bundle decomposition and the weighted bundle decomposition.

Table 8.18: Average computation time per optimization problem (without parallelization), memory consumption, average number of iterations of the progressive hedging algorithm and number of variables of each optimization problem for the problem with 9 scenarios for the full decomposition approach, the bundle decomposition and the weighted bundle decomposition.

# Decomposition	Time [s]	Memory [Mb]	# Iterations	# Variables
Full decomposition	199.98	35.6	47.12	860
Bundle	40.12	55.40	15.65	2580
Weighted bundle	25.72	55.40	9.49	2580

8.3 Discussion

This chapter presented the central simulation results of the thesis, in which the advantages of the presented multi-stage NMPC with respect to several other approaches are shown. The chapter is structured in two parts which present two case studies for different polymerization processes.

In the first section, the potential of multi-stage NMPC was demonstrated by simulation studies of the Chylla-Haase polymerization reactor. The results show that multi-stage NMPC satisfies the temperature constraints for all possible cases of the uncertainty and leads to better performance than standard and min-max NMPC. Multi-stage NMPC outperforms standard NMPC also with perfect non-anticipative information of the uncertainty in the case of a step change in the disturbance. If the uncertain parameters can be estimated, the weights of the scenarios in the cost function of multi-stage NMPC can be changed according to the estimation results in order to enhance the performance while maintaining the robustness of the approach.

In the second section, multi-stage NMPC is applied to an industrial batch polymerization reactor example provided by BASF SE. Simulation results show that multi-stage NMPC with an economic cost is able to satisfy tight temperature constraints and safety-related constraints for all the scenarios under consideration that represent variations of ±30% in critical parameters, while standard NMPC and standard NMPC with bias term violate the constraints. Choosing a standard NMPC scheme with tracking term and with a conservative choice of the uncertain parameters results in no constraint violations but increases the obtained batch times significantly compared to multi-stage NMPC. Multi-stage NMPC is also compared to other robust approaches proposed in the literature. It is shown that the use of a robust open-loop approach can lead to an important decrease of the performance compared to approaches with feedback or recourse. The introduction of feedback using state feedback policies (constant or time-varying) improves the performance, but for nonlinear systems it can be far from the performance obtained with the use of multi-stage NMPC as in this case. Also, a comparison of the computation times and memory needed for a monolithic and for a decomposition-based solution of the optimization problem has been provided. These results suggest that if an efficient implementation is done, the decomposition approach should be preferred over the monolithic only if a large degree of parallelization is available or if not much memory is available. The stability of the multi-stage NMPC controllers has been shown by means of extensive simulations, since for real case-studies some of the assumptions necessary (especially the computation of control robust invariant sets and the positive definiteness of the cost function) are not fulfilled, or very difficult to check.

The results presented in this chapter have been obtained for challenging industrial case studies (highly nonlinear dynamics, tight constraints, economic objective). For this reason, multi-stage NMPC represents a promising technique for the practical application of robust NMPC techniques on real-world problems.

This chapter also provides an insight on the important advantages that the efficient implementation of the approach proposed in this thesis. The use of exact first and second order derivatives provided by CasADi together with the use of a collocation approach solved using IPOPT provides a crucial improvement over simple implementations using TOMLAB/MATLAB. Problems with longer horizons and more uncertainties can be solved faster and more accurately.

The multi-stage NMPC approach has been succesfully applied to other systems which are not reported in this thesis for brevity. Results obtained for a highly nonlinear bioreactor can be seen in (Lucia and Engell, 2012) and results for a penicillin production process are reported in (Lucia and Engell, 2013). Multi-stage NMPC has been also applied to a challenging example for the generation of renewable energy using kites in (Lucia and Engell, 2014).

Chapter 9

Multi-stage NMPC with Guaranteed Stability

This chapter presents a simple example for which multi-stage NMPC with an a priori guarantee of stability and recursive feasibility is achieved. The results presented here are obtained using the formulation presented in Chapter 4. This chapter is based on the publication (Lucia et al., 2014c).

9.1 Illustrative Example

A spring-damper-mass system is considered, adapted from the problem presented in (Raimondo et al., 2009). The system model can be written as:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{k_0}{m} e^{-x_1} x_1 - \frac{b^r}{M} x_2 + \frac{u}{M} \end{aligned} \tag{9.1}$$

where x_1 and x_2 are the position and the velocity of the mass m, k_0 is the constant of the nonlinear spring, u is the control input and b^r is the uncertain damping constant which is supposed to have three different possible values $b^r = \{b^1, b^2, b^3\} = \{1.0, 2.0, 4.0\}$ with probabilities $\{\pi^1, \pi^2, \pi^3\} = \{0.33, 0.34, 0.33\}$. The worst-case assumption that the damping constant varies unpredictably between the sampling instances is taken, considering a sampling time of $t_{\text{step}} = 0.4$ s. For the implementation of the multi-stage NMPC with guaranteed stability it is necessary to determine the terminal ingredients κ_f, V_f, and $\mathbb{X}_f$.

For this purpose, the next section presents an extension of the method based on linearization around the equilibrium point presented for nominal NMPC in (Rawlings and Mayne, 2009). Note that it is also possible to apply other methods based on linear differential inclusions, as the one described in (Chen et al., 2003) or on the techniques presented in (Fiacchini et al., 2010) that use different of convex functions (DC programming).

9.2 Calculation of the Terminal Ingredients

The first step is to obtain an auxiliary control law κ_f. For this purpose, each of the possible realizations of the system is linearized around the origin and discretized using the sampling time $t_{\text{step}} = 0.4$ s, getting the state space representations $x(k+1) = A^r x + B^r u,\ \forall\ r \in \{1, 2, 3\}$. In order to satisfy the non-anticipativity constraints, a unique κ_f for all possible realizations of the uncertainty has to be determined. A state feedback controller is calculated using the LQR method for the nominal system and it is checked that all realizations are closed-loop stable with this control input κ_f. In this case, $\kappa_f = Kx$ is chosen with $K = [-1.1869 - 0.9804]$. Next, the following Lyapunov equation is solved for each realization r:

$$A_K^{rT} P^r A_K^r + 2Q^* = P^{rT}, \tag{9.2}$$

with $A_K^r = A^r + B^r K$ and $Q^* = \rho(Q + K^T R K)$ where ρ is a tuning parameter, and Q and R are the matrices defining the quadratic stage cost of the NMPC controller $\ell(\cdot) = x'Qx + u'Ru$ with

$$Q = \begin{bmatrix} 30 & 0 \\ 0 & 20 \end{bmatrix}, R = 1. \tag{9.3}$$

From the Lyapunov equations using $\rho = 2$ the following positive definite matrices are obtained:

$$\begin{gathered} P^1 = \begin{bmatrix} 473.07 & 146.68 \\ 146.68 & 142.24 \end{bmatrix}, P^2 = \begin{bmatrix} 470.20 & 162.54 \\ 162.54 & 194.50 \end{bmatrix}, \\ \text{and } P^3 = \begin{bmatrix} 644.45 & 227.50 \\ 227.50 & 408.64 \end{bmatrix}. \end{gathered} \tag{9.4}$$

We define for each possible realization r different terminal cost functions as:

$$V_f^r(x) = \frac{1}{2} x^T P^r x. \tag{9.5}$$

The function $V_f^s = \sum_{r=1}^{s} \pi_0^r V_f^r$ is proposed as a local common control Lyapunov function for the real system in some neighborhood of the origin. In order to show this, it is necessary that:

$$V_f^s(f(x, Kx, d^r)) - V_f^s(x) < 0, \ \ \forall x \in (Z(a) \setminus 0), \ \forall r. \tag{9.6}$$

The set $Z(a)$ is chosen to a be a sublevel set of the corresponding Lyapunov function, i.e. $Z(a) = \text{lev}_a V_f^s = \{x | V_f^s(x) \leq a\}$. In addition to condition (9.6), it is also necessary for the proof of stability that the terminal penalty terms satisfy assumption 3.E. Then, by simulations gridding the state space, a value of a is chosen so that both conditions are satisfied. Here, we choose the maximum a that satisfies both conditions ($a = 1200$) and $\rho_d(|d|) = 0$. The terminal region $\mathbb{X}_f$ is chosen to be

$$\mathbb{X}_f = Z(a). \tag{9.7}$$

The proposed terminal set has to satisfy both the state and the control constraints, that is, $\mathbb{X}_f \subseteq \mathbb{X}$ and $K\mathbb{X}_f \subseteq \mathbb{U}$. If it is not the case, the value of a should be reduced.

In this way, $\mathbb{X}_f$ is a robust control invariant set for the system (4.1), i.e., there exists a control input (in this case $\kappa_f = Kx$) such that the system remains in $\mathbb{X}_f$ for all possible values of the uncertainty. The terminal region can be seen in Fig. 9.1. Fig. 9.1 also shows the prediction of the scenario tree with prediction horizon $N_P = 4$ equal to robust horizon $N_R = 4$ which is obtained by solving the multi-stage NMPC problem with guaranteed stability (3.5) at one time instance. It can be seen that all the constraints, including the terminal constraint are satisfied.

9.3 Results

Once all the necessary ingredients for the stability guaranteeing formulation of multi-stage NMPC have been calculated, the control problem can be stated as in (3.5) and then solved in a receding horizon fashion. For a comparison with other approaches, a prediction horizon $N_P = 4$ and an initial condition $x_0 = [5.3, 2.0]$ is considered. There are constraints on the states ($x_1 \in [-10, 10]$ and $x_2 \in [-4, 10]$) and on the control inputs ($u \in [-6, 6]$). The stage cost is $\ell(x, u) = x'Qx + u'Ru$ and the uncertainty varies randomly at each time step with the previously defined probabilities. Fig. 9.2 shows the behavior of the system for 100 different runs with random values of the damping constant

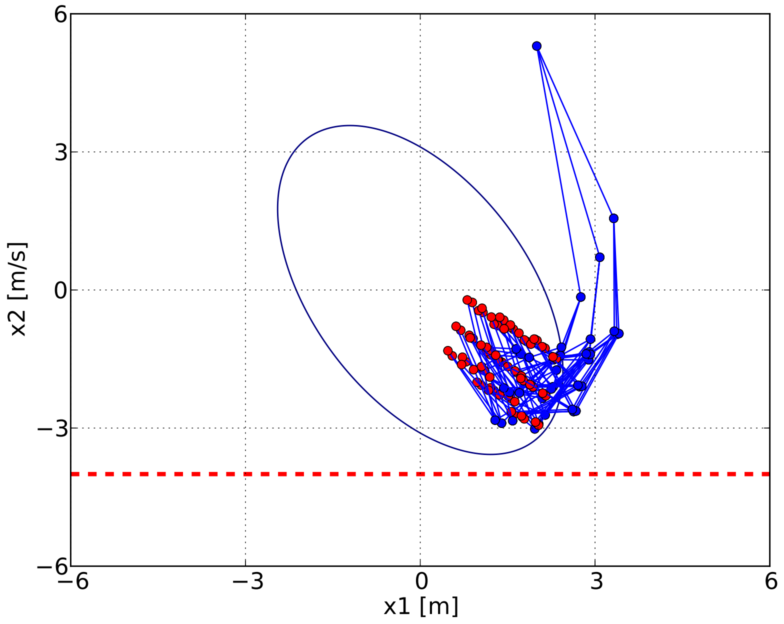

Figure 9.1: Prediction of the scenario tree for multi-stage NMPC at the initial time with a prediction horizon of $N_P = 4$. The elliptical terminal set and the state constraints are also shown. The leaf nodes are indicated in red.

with multi-stage NMPC. None of the samples violates the constraints. As expected, if standard NMPC with guaranteed stability is used (using the same terminal set) some of the realizations violate the constraints as can be seen in Fig. 9.3. In addition, the average accumulated cost for the multi-stage case is better than the one obtained for standard NMPC as can be seen in the results reported in Table 9.1. Table 9.1 also shows that an open-loop approach, in which all the control inputs at each stage are equal (no recourse), leads to a higher cost. Since the uncertain parameter enters the model in a product with the state x_2, the uncertainty has no effect at the origin and therefore all the controllers converge exactly to the origin instead of to a set around it. The optimal control problem is solved using collocation on finite elements. The algorithms are implemented using CasADi and IPOPT as explained in Chapter 5, which makes it possible to solve the problem with computation times per iteration that are far below the sampling time and it is therefore real-time implementable (see Table 9.1).

Fig. 9.4 shows a comparison of the feasibility regions ($\mathbb{X}_A(N_P)$) for different controllers, which has been obtained numerically (gridding accuracy of 0.2 m and 0.2 m/s for x_1 and x_2 respectively). As expected, the feasibility region of the multi-stage controller is larger

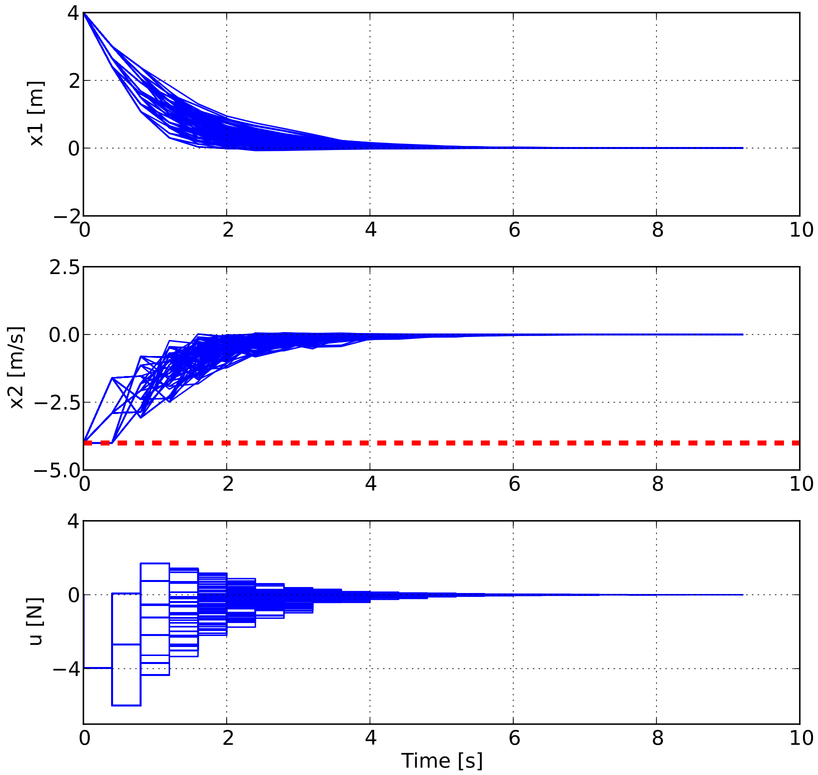

Figure 9.2: States and control trajectories for 100 different runs using multi-stage NMPC with the uncertainty varying randomly at each sampling time.

Table 9.1: Comparison of the average cost and average computation time per NMPC iteration obtained with standard, multi-stage and open-loop NMPC (100 samples).

Controller	**Cost**	**Comp. time/iter [ms]**
Standard	1567.47	5.65
Multi-stage	1558.65	33.91
Open-loop	1580.34	103.04

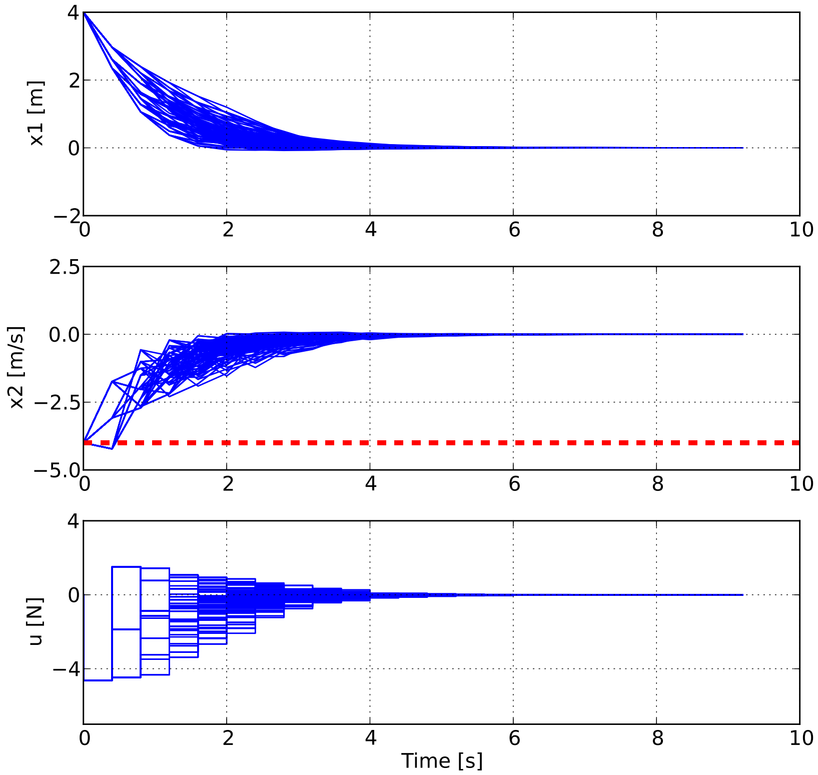

Figure 9.3: States and control trajectories for 100 different runs using standard NMPC with the uncertainty varying randomly at each sampling time.

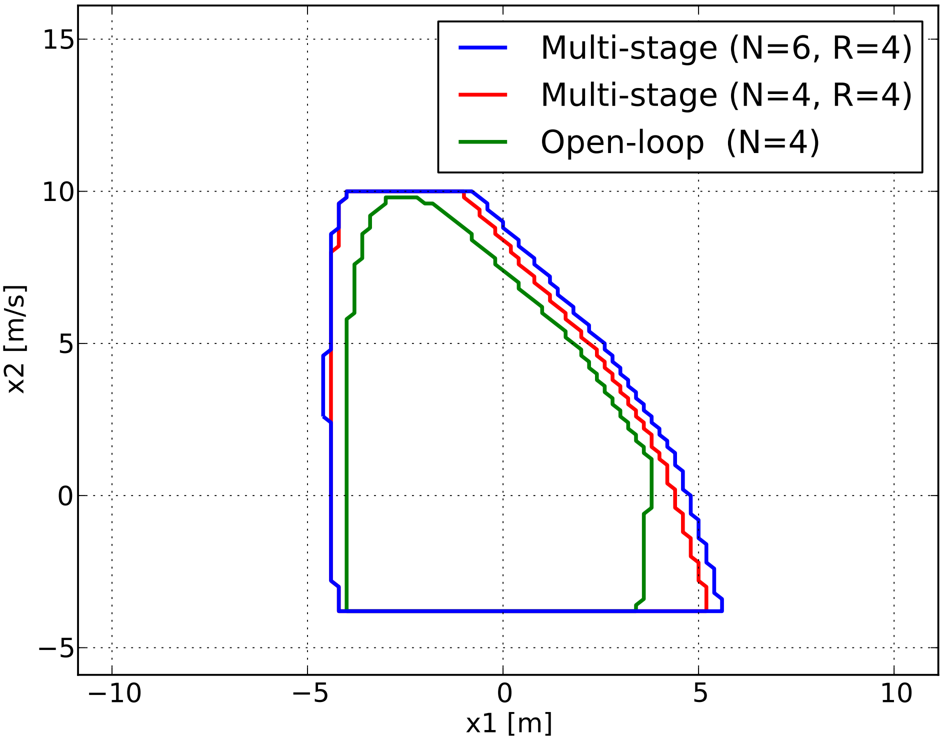

Figure 9.4: Feasibility region of open-loop NMPC, multi-stage NMPC and multi-stage NMPC with robust horizon.

than the one obtained with the open loop controller, because of the possibility of adapting the future inputs to the disturbances. If a longer prediction horizon $N_P = 6$ is used with a robust horizon $N_R = 4$ to avoid the exponential growth of the tree, Assumption 4 is satisfied for a higher value of a than Assumption 3.E, and therefore a bigger terminal set can be chosen. In this case $a = 1600$ is chosen. This results in a bigger feasibility region as shown in Fig. 9.4.

This chapter shows that an implementation of multi-stage NMPC with an a priori guarantee of stability and recursive feasibility can be done in real-time, provided that the terminal ingredients can be calculated. For the general nonlinear case it is very challenging to obtain the required terminal ingredients. In this thesis this has been done based on a linearization around the equilibrium point and a gridding of the state space (done off-line), which might not possible for systems of high dimensions due to the necessary computation power. Other methods besides the linearization approach can be used such as linear differential inclusions (LDI), as described in (Chen et al., 2003), or applying DC programming as in (Fiacchini et al., 2010). It is also possible to use machine learning techniques as the Support Vector Machine algorithms employed in (Ong et al., 2006). However, the calculation of a robust control invariant set and its corresponding terminal control law which are not overly conservative remains a challenge for general nonlinear systems, especially in the case of high order systems.

Another challenge is that the stability proof provided in this thesis is valid for a classical tracking MPC scheme, in which the stage cost is positive definite, which is used to prove that the optimal cost of the multi-stage NMPC scheme is a Lyapunov function. As it was discussed in Chapter 4, this is not always the case when an economic cost function is used and therefore new assumptions and different arguments have to be used in order to achieve an a priori guarantee for stability and recursive feasibility as described in (Angeli et al., 2012).

Chapter 10

Experimental Results of Multi-stage NMPC of a Laboratory Plant

This chapter presents the experimental results obtained from the application of standard and multi-stage NMPC to a laboratory process using DO-MPC. This simple example illustrates that the tools and methods provided in this thesis enable the implementation of the multi-stage NMPC approach at a real plant without much effort.

10.1 A Continuous Stirred Tank Reactor

The plant under consideration (see Fig. 10.1) is located at the Group of Process Dynamics and Operations at the TU Dortmund. It consists of a continuous stirred tank reactor (CSTR) equipped with a jacket. Only the thermal behavior of the CSTR is realized experimentally and the kinetics of the chemical reaction are virtual. A schematic representation of the plant is shown in Fig. 10.2.

The jacket is fed with water supplied from a thermostat, which can heat or cool the water if necessary, modifying the temperature at the inlet of the jacket T_j^{in}. The reactor is operated in a semi-batch mode, such that the input flow ($\dot{V}_{\mathrm{R}}^{\mathrm{in}}$) can be adjusted using a pump. For this experiment no real reaction takes place inside the reactor. A virtual reaction is simulated using a heating rod which produces the heat that an exothermic reaction taking place inside the reactor would produce.

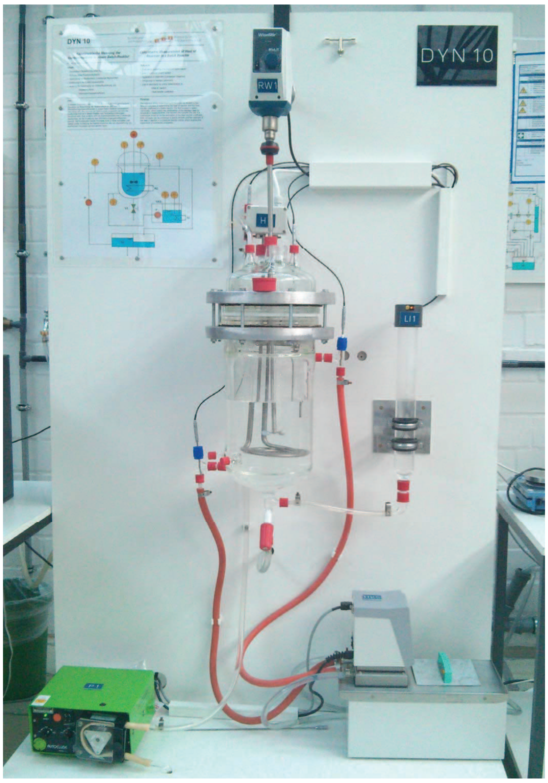

Figure 10.1: Experimental setup of the thermal part of the CSTR located at TU Dortmund.

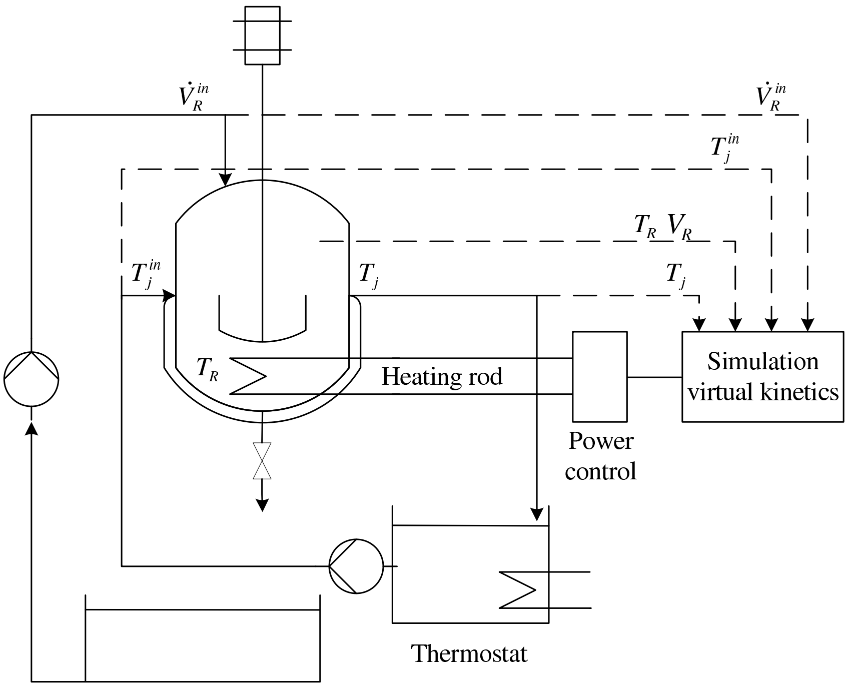

Figure 10.2: Schematic representation of the CSTR under consideration with virtual reaction kinetics.

It is considered that the following reaction takes place inside the reactor:

$$A + B \to C. \tag{10.1}$$

Using energy and mass balances, it is possible to derive the differential equations that describe the dynamics of the system and the following 7 ODEs are obtained:

$$\dot{V}_\mathrm{R} = \dot{V}_\mathrm{R}^\mathrm{in}, \tag{10.2a}$$

$$\dot{T}_\mathrm{R} = \frac{\dot{V}_\mathrm{R}^\mathrm{in}}{V_\mathrm{R}}(T_\mathrm{R}^\mathrm{in} - T_\mathrm{R}) - \frac{\alpha A(T_\mathrm{R} - T_\mathrm{J})}{\rho V_\mathrm{R} c_\mathrm{p}} - \frac{k c_\mathrm{A} c_\mathrm{B} \Delta H_\mathrm{R}}{\rho c_\mathrm{p}}, \tag{10.2b}$$

$$\dot{T}_\mathrm{J} = \frac{\dot{V}_\mathrm{J}^\mathrm{in}}{V_\mathrm{J}}(T_\mathrm{J}^\mathrm{in} - T_\mathrm{J}) + \frac{\alpha A(T_\mathrm{R} - T_\mathrm{J})}{\rho V_\mathrm{J} c_\mathrm{p}}, \tag{10.2c}$$

$$\dot{c}_\mathrm{A} = -\frac{\dot{V}_\mathrm{R}^\mathrm{in}}{V_\mathrm{R}} c_\mathrm{A} - k c_\mathrm{A} c_\mathrm{B}, \tag{10.2d}$$

$$\dot{c}_\mathrm{B} = -\frac{\dot{V}_\mathrm{R}^\mathrm{in}}{V_\mathrm{R}}(c_\mathrm{b}^\mathrm{in} - c_\mathrm{b}) - k c_\mathrm{A} c_\mathrm{B}, \tag{10.2e}$$

$$\dot{c}_\mathrm{C} = -\frac{\dot{V}_\mathrm{R}^\mathrm{in}}{V_\mathrm{R}} c_\mathrm{C} + k c_\mathrm{A} c_\mathrm{B}, \tag{10.2f}$$

$$\dot{T}_\mathrm{J}^\mathrm{in} = \frac{\bar{T}_\mathrm{J}^\mathrm{in} - T_\mathrm{J}^\mathrm{in}}{\tau_\mathrm{t}}. \tag{10.2g}$$

The model includes a mass balance for the volume of the reactor V_R where $\dot{V}_\mathrm{R}^\mathrm{in}$ is the input flow, which is a control input of the system. The energy balances in (10.2b) and in (10.2c) describe the dynamics of the temperature of the reactor T_R and of the temperature of the jacket T_J, which is assumed to be perfectly mixed. The parameter α is the heat transfer coefficient and A is the heat transfer area, which can be calculated as a function of the radius of the reactor r and of the volume of water in the reactor V_R as $A = \pi r^2 + 2\frac{V_\mathrm{R}}{r}$. The specific heat and the density of the water used in the experiment are denoted by c_p and ρ. $\dot{V}_\mathrm{J}^\mathrm{in}$ is the flow rate into the jacket, which is considered constant, and T_R^in denotes its temperature. V_J is the volume of the cooling fluid inside the jacket. The component balances for the substances A, B, C are described in (10.2d)-(10.2f), where c_i denotes the concentration of component i. The reaction constant is denoted by k and c_b^in is the concentration of component B in the inflow $\dot{V}_\mathrm{R}^\mathrm{in}$. In the considered setup the inflow only contains the substance B. The dynamics of the thermostat in (10.2g) are approximated by a first order system with a time constant τ_t. The thermostat has different dynamics depending if it is cooling or heating the water. For this reason the time constant τ_t varies depending on the temperature of the water at the inlet of the jacket:

$$\tau_t = \begin{cases} \tau_t^\mathrm{cool}, & T_\mathrm{J}^\mathrm{in} \geq \bar{T}_\mathrm{J}^\mathrm{in}, \\ \tau_t^\mathrm{heat}, & T_\mathrm{J}^\mathrm{in} < \bar{T}_\mathrm{J}^\mathrm{in}. \end{cases} \tag{10.3}$$

This introduces a discontinuity in the derivative of the model, which should be reformulated if it causes convergence problems for the NLP solver. The set-point used in the thermostat ($\bar{T}_\mathrm{J}^\mathrm{in}$) is the second control input of the system. The values of all the parameters of the model can be seen in Table 10.1.

The initial conditions of the states together with the constraints on the states are described in Table 10.2. The constraints for the control inputs are shown in Table 10.3. All the parameters have been obtained experimentally. The parameters that determine the virtual reaction are k and ΔH, which is the reaction enthalpy. These parameters are chosen so that it is possible to emulate the virtual reaction with the equipment available. The heat that would be generated by the reaction $Q_\mathrm{rea} = \Delta H \, k \, c_\mathrm{A} c_\mathrm{B} V_\mathrm{R}$ is generated by the heating rod during the experiment.

The control task of this problem consists in maximizing the amount of product C obtained ($n_\mathrm{C} = V_\mathrm{R} c_\mathrm{C}$), while satisfying the state and input constraints defined in Tables 10.2 and 10.3.

Table 10.1: Parameter values of the CSTR.

Parameter	**Value**	**Unit**
$\dot{V}_{\mathrm{J}}^{\mathrm{in}}$	$9 \cdot 10^{-6}$	$\mathrm{m}^3\mathrm{s}^{-1}$
$T_{\mathrm{R}}^{\mathrm{in}}$	299.15	K
α	0.1484	KWm^{-2}
r	0.092	m
ρ	1000	$\mathrm{kg\,m}^{-3}$
c_p	4.2	$\mathrm{kJ\,kg}^{-1}\,\mathrm{K}^{-1}$
V_{J}	0.00222	m^3
k	$1.339 \cdot 10^{-6}$	$\mathrm{m}^3\,\mathrm{mol}^{-1}\,\mathrm{s}^{-1}$
ΔH	-50	$\mathrm{kJ\,mol}^{-1}$
$c_{\mathrm{b}}^{\mathrm{in}}$	$4 \cdot 10^{6}$	$\mathrm{mol\,m}^{-3}$
$\tau_{\mathrm{t}}^{\mathrm{heat}}$	200	s^{-1}
$\tau_{\mathrm{t}}^{\mathrm{cool}}$	950	s^{-1}

Table 10.2: Initial conditions and state constraints.

State	**Init. cond.**	**Min.**	**Max.**	**Unit**
V_{R}	0.0035	0	0.01	m^3
T_{R}	50.0	48.0	52.0	°C
T_{J}	50.0	0	100.0	°C
c_{A}	2000.0	0	inf	mol m^{-3}
c_{B}	0	0	inf	mol m^{-3}
c_{C}	0	0	inf	mol m^{-3}
$T_{\mathrm{J}}^{\mathrm{in}}$	50.0	0	100	°C

Table 10.3: Bounds on the manipulated variables.

Control	**Min.**	**Max.**	**Unit**
$\dot{V}_{\mathrm{R}}^{\mathrm{in}}$	0	$9 \cdot 10^{-6}$	$\mathrm{m}^3\mathrm{s}^{-1}$
$\bar{T}_{\mathrm{J}}^{\mathrm{in}}$	30	80	°C

10.2 Simulation and Experimental Results

This section presents the simulation and experimental results of standard and multi-stage NMPC for the CSTR presented above.

The first step before applying a model-based technique to control a system is to check the quality of the available model and its prediction capabilities. With the help of this information, it has to be decided whether the available model is predicting the behavior of the system well enough or whether some improvements are needed. For this purpose, an experiment was performed in which a trajectory for the control inputs was given and the temperature measurements of the experimental setup are compared with the ones obtained by simulating the model described in (10.2). From the given initial condition the model is also simulated in parallel and the values of the concentrations are used to calculate the heat of reaction Q_{rea} which is generated using the heating rod that emulates the chemical reaction.

The obtained results are presented in Fig. 10.3. The simulation reproduces well the results obtained experimentally and it is decided that the model is accurate enough to be used for nonlinear model predictive control. Small deviations are expected due to the assumptions made (perfect mixing in the jacket), the approximation of the thermostat dynamics by a first order system, unmodeled effects (heat transfer to the environment), small inaccuracies in some of the reactor parameters and sensor noise.

In order to achieve the control task within the framework of NMPC, the following optimization problem is solved at each sampling time:

$$\min_{x_k^j, u_k^j \forall (j,k) \in I} \quad -\sum_{i=1}^{N} \omega_i \sum_{k=0}^{N_P-1} V_{\mathrm{R},k}^j c_{\mathrm{C},k}^j + r_1 \Delta \dot{V}_{\mathrm{R}}^{\mathrm{in}\,2} + r_2 \Delta \bar{T}_{\mathrm{J}}^{\mathrm{in}\,2} + \mu {\epsilon_k^j}^2, \quad \forall (j,k) \in I \tag{10.4a}$$

subject to:

$$x_{k+1}^j = f(x_k^{p(j)}, u_k^j, d_k^{r(j)}), \qquad \forall\, (j, k+1) \in I, \tag{10.4b}$$

$$0 \leq V_{\mathrm{R},k}^j \leq 0.01, \qquad \forall\, (j,k) \in I, \tag{10.4c}$$

$$48 \leq T_{\mathrm{R},k}^j + \epsilon_k^j \leq 52, \qquad \forall\, (j,k) \in I, \tag{10.4d}$$

$$0 \leq T_{\mathrm{J},k}^j \leq 100, \qquad \forall\, (j,k) \in I, \tag{10.4e}$$

$$0 \leq T_{\mathrm{J},k}^{\mathrm{in},j} \leq 100, \qquad \forall\, (j,k) \in I, \tag{10.4f}$$

$$0 \leq c_{\mathrm{A},k}^j, \qquad \forall\, (j,k) \in I, \tag{10.4g}$$

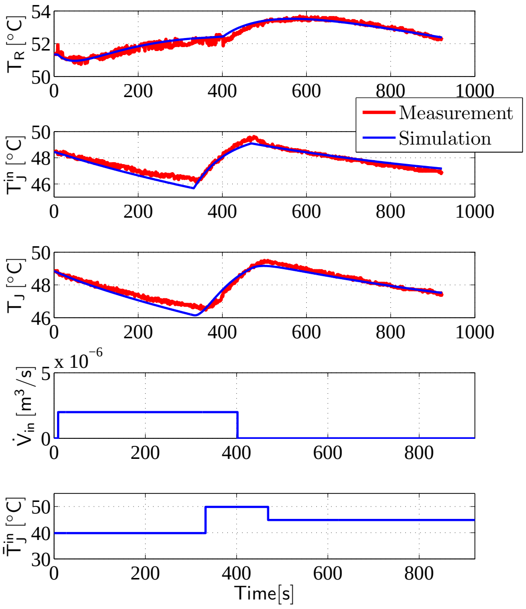

Figure 10.3: Temperature of the reactor, temperature at the inlet of the jacket and temperature of the jacket for a given trajectory of control inputs. The experimental results are shown in red and the simulation results in blue.

$$0 \leq c_{\mathrm{B},k}^{j}, \qquad \forall\,(j,k) \in I, \tag{10.4h}$$

$$0 \leq c_{\mathrm{C},k}^{j}, \qquad \forall\,(j,k) \in I, \tag{10.4i}$$

$$0 \leq c_{\mathrm{B},k}^{\mathrm{acc},j} \leq 2000, \qquad \forall\,(j,k) \in I, \tag{10.4j}$$

$$0 \leq \dot{V}_{\mathrm{R}}^{\mathrm{in}} \leq 9\,10^{-6}, \qquad \forall\,(j,k) \in I, \tag{10.4k}$$

$$30 \leq \bar{T}_{\mathrm{J}}^{\mathrm{in}} \leq 80, \qquad \forall\,(j,k) \in I, \tag{10.4l}$$

$$u_k^j = u_k^l \text{ if } x_k^{p(j)} = x_k^{p(l)} \qquad \forall\,(j,k),(l,k) \in I, \tag{10.4m}$$

Since this optimization problem is going to be solved using real measurements, the hard constraint on the temperature of the reactor is relaxed and implemented as a soft constraint by adding the parameter ϵ_k^j at each point in the prediction as shown in (10.4d) and then penalized in the cost function (10.4a) using a big penalty term $\mu = 10^6$. This is done to avoid infeasible optimization problems. Since the optimal solution is to operate at the constraint, it is likely that due to measurement errors or plant model mismatches, at some point in time the measurement lies outside of the constraint and therefore the resulting optimization problem becomes infeasible if no additional measures are taken.

This can jeopardize the performance of the control scheme giving dangerous commands to the real plant if no backup strategy is available when an infeasible optimization problem is encountered. Note that it is also possible to use an ℓ_1 penalty term to implement the soft constraints, which makes it possible to use a smaller value of μ at the cost of introducing a nondifferentiable element in the cost. Furthermore, an additional constraint (10.4j) is added to limit the amount of component B that can be fed into the reactor $c_{\mathrm{B}}^{\mathrm{acc}} = \int_0^{\infty} c_{\mathrm{b}}^{\mathrm{in}} \dot{V}_{\mathrm{R}}^{\mathrm{in}} \,\mathrm{d}t$. If this constraint is not added the batch would end with a large amount of component B in the reactor and this is not desired. To avoid oscillatory behavior of the control inputs, penalty terms on the control moves are introduced as indicated in (10.4a) using the tuning parameters $r_1 = 5 \cdot 10^7$ and $r_2 = 0.001$.

The setup of the problem and of the different parameters is chosen in such a way that the optimal solution of the problem resembles the one of the polymerization reactions presented in Chapter 8. It is thus reasonable to draw some conclusions about the possible implementation of the approach at real polymerization reactors.

The implementation of standard and multi-stage NMPC is realized using DO-MPC, both for the simulation and for the experimental results. The communication between the plant and the computer is done via USB using the data acquisition system Labjack U12 except for the thermostat, which uses serial port (RS-232) communication. To perform the experiments, the system is first driven to the initial condition described in Table 10.2. Then the NMPC controller is started and at the same time a simulation of the system is run in parallel using as control inputs those computed by the NMPC. Every second, the values of the concentrations obtained by the parallel simulation are used to calculate the heat of reaction Q_{rea} and the corresponding command is given to the heating rod to produce it. This emulates the chemical reaction that would take place in the reactor. The values of the concentrations obtained by the simulation are corrupted by white noise ($\sigma = 0.1\mathrm{mol\ m}^{-3}$) and used as measurements for the NMPC controller. That is, it is considered that all the states can be measured: The temperatures and the volume of the reactor are directly measured using suitable sensors and the concentrations are obtained from the parallel simulation of the reaction after the addition of measurement noise. The same probabilities ω_i are chosen for all the scenarios. The sampling time of the controller is $t_{\mathrm{step}} = 20$ s and the prediction horizon is $N_P = 15$ steps. For the multi-stage results, a scenario tree is generated considering as scenarios the combinations of the maximum, minimum and nominal values of the uncertain reaction parameters k and ΔH. It is considered that they can vary ±25% with respect to the nominal values reported in Table 10.1. A robust horizon of $N_R = 1$ is chosen.

Fig. 10.4 shows the performance of standard NMPC applied to the CSTR without uncertainties, i.e. the values of the reaction parameters k and ΔH are the same in the model and in the emulation of the reaction heat. On the left, the simulation results (which are obtained assuming perfect measurements without noise) are shown and the right figure contains the experimental results. In a similar way as in the polymerization processes presented in Chapter 8, the feed ($\dot{V}_{\mathrm{R}}^{\mathrm{in}}$) is adjusted such that the constraints on the temperature of the reactor are satisfied. The NMPC keeps the setpoint of the thermostat ($\bar{T}_{\mathrm{J}}^{\mathrm{in}}$) to the minimum to be able to feed more and thus to maximize the amount of component C produced. By comparing the simulation results in Fig. 10.4 (left) and the experimental results in Fig. 10.4 (right) it can be concluded that the standard NMPC of the CSTR using DO-MPC has an excellent performance and is capable of reproducing the simulation results accurately.

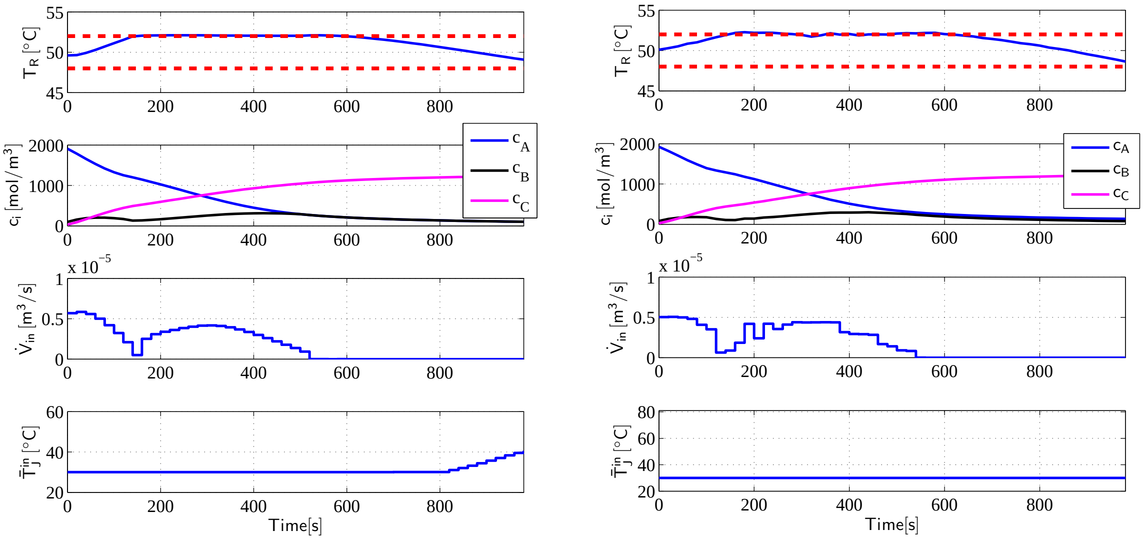

Figure 10.4: Reactor temperature, concentrations, input flow and setpoint of the thermostat for standard NMPC with no uncertainty on the reaction parameters. The left plot shows simulation results and the right plot shows experimental results.

One of the main features of the modular implementation of NMPC realized in DO-MPC is that once a satisfactory simulation result has been obtained in simulation after tuning the different parameters – as in Fig. 10.4 (left) – , the only necessary step to obtain the experimental results in Fig. 10.4 (right) is to exchange the simulator module that uses an integrator (in this case SUNDIALS) for the module that contains the interface with the real plant. This is illustrated in Fig. 10.5 which shows the DO-MPC configurations used to obtain the results reported in Fig. 10.4. This makes the process of transferring the simulation results to reality simple and transparent, and possible errors can be tracked easily.

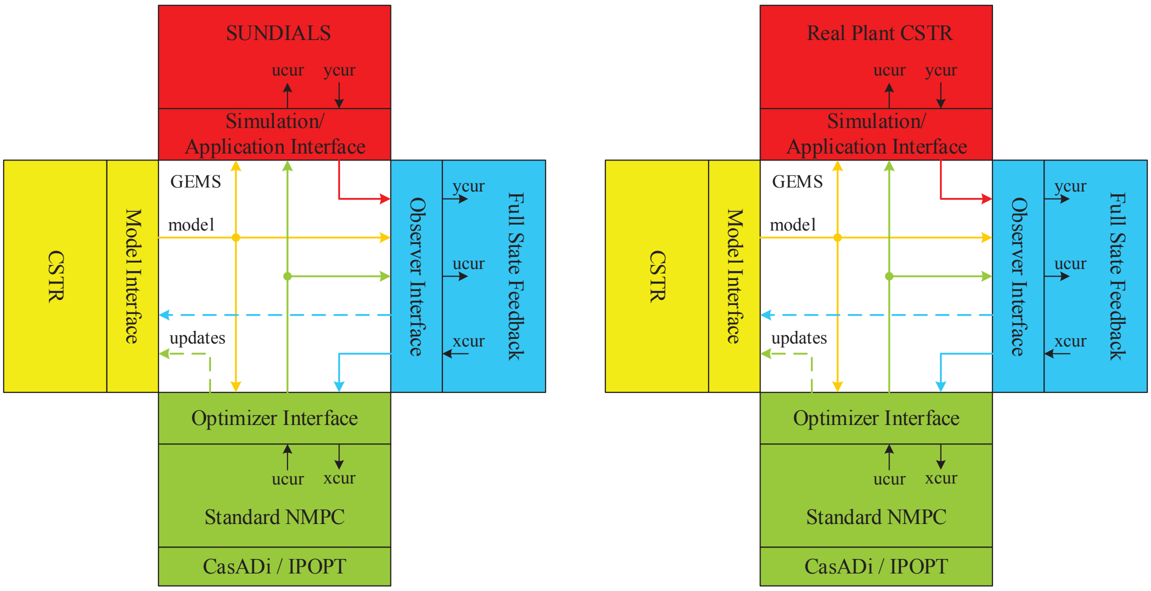

Figure 10.5: DO-MPC configurations used to obtain the simulation results (left) and the experimental results (right) of standard NMPC applied to the CSTR under consideration.

If standard NMPC is applied when the uncertain parameters differ from the nominal values considered in the model, violations of the constraint on the reactor temperature T_{R} occur as it can be seen in the experimental results shown in Fig. 10.6. In this case it is considered that the reaction parameters k and ΔH, which are used to perform the parallel simulation of the system and to calculate the heat of reaction Q_{rea}, have values that are a 25% higher than their nominal values.

The violations of the constraints can be avoided if multi-stage NMPC is used. This can be observed in Fig. 10.7 where the results for multi-stage NMPC of the CSTR are shown. The left plot shows the simulation results and the right plot the results obtained experimentally. Again, the simulation results match the experimental results very well. A longer batch time is obtained when compared to the use of standard NMPC shown in Fig. 10.6 because the feed is reduced to avoid the violations of the constraints. Fig 10.8

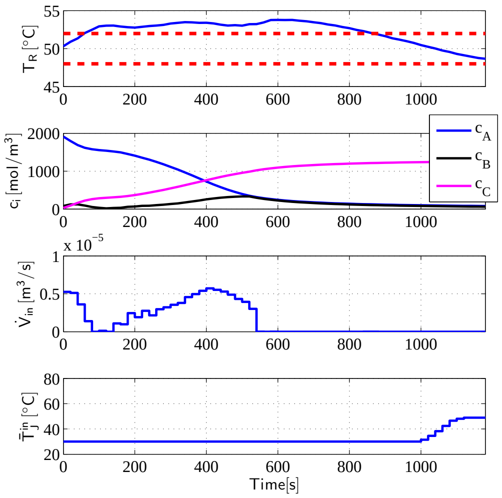

Figure 10.6: Experimental results for the CSTR using multi-stage NMPC. The actual reaction parameters ΔH and k are considered to be 25% higher than the nominal values.

shows further experimental results of multi-stage NMPC for different values of the uncertain parameters. The case when both uncertain parameters are chosen to be a 25% smaller than their nominal value is shown in Fig 10.8 (left) and the case when uncertain parameters have their nominal value can be seen in Fig 10.8 (right). It can be seen that multi-stage NMPC satisfies the constraints for all the scenarios also in the experiments.

The average computation time needed for the solution of each optimization problem is 0.073 s for the standard NMPC case and 0.65 s for the multi-stage case. The standard NMPC problem has around 500 variables and constraints and the multi-stage problem around 4500. In terms of iterations of the interior point algorithm, IPOPT needs in average around 22 iterations to find an optimal solution for the standard case, and around 26 for the multi-stage case. Since the needed computation time is small in comparison to the sampling time ($t_{\text{step}} = 20$ s), no additional measures are applied to counteract the effect of the computation delay. For other cases where the delay is significant, it can be taken into account by simulating the system for the expected computation time and then by using this state as initial condition of the NMPC controller. More advanced techniques to cope with this problem include the use of the real-time iteration algorithm described in (Diehl et al., 2005).

This chapter described the experimental results obtained for a nonlinear CSTR with

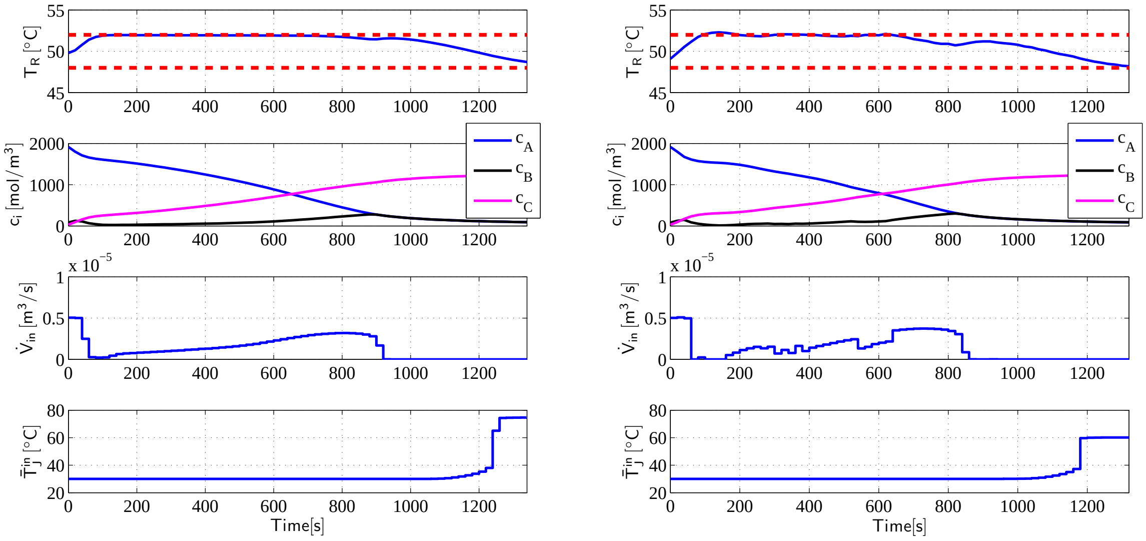

Figure 10.7: Simulation (left) and experimental (right) results for the CSTR using multi-stage NMPC. The actual reaction parameters ΔH and k are considered to be 25% higher than the nominal values.

virtual reaction kinetics using multi-stage NMPC. The results show that it is possible to apply to a real system the methods developed in this thesis and that the use of the tool DO-MPC for its implementation facilitates the task of transferring a controller that has a satisfactory performance in simulation to the real plant. The experimental setup is chosen in such a way that the optimal operation resembles the one obtained for the polymerization reactor problems considered in Chapter 8. Thus it is reasonable to conclude that the application of multi-stage NMPC to the real reactors is possible and would achieve the expected results that have been suggested by the simulation studies. For that case, it is also necessary to include an observer to estimate the states that cannot be measured, since in this case study it was assumed that all the states are measurable.

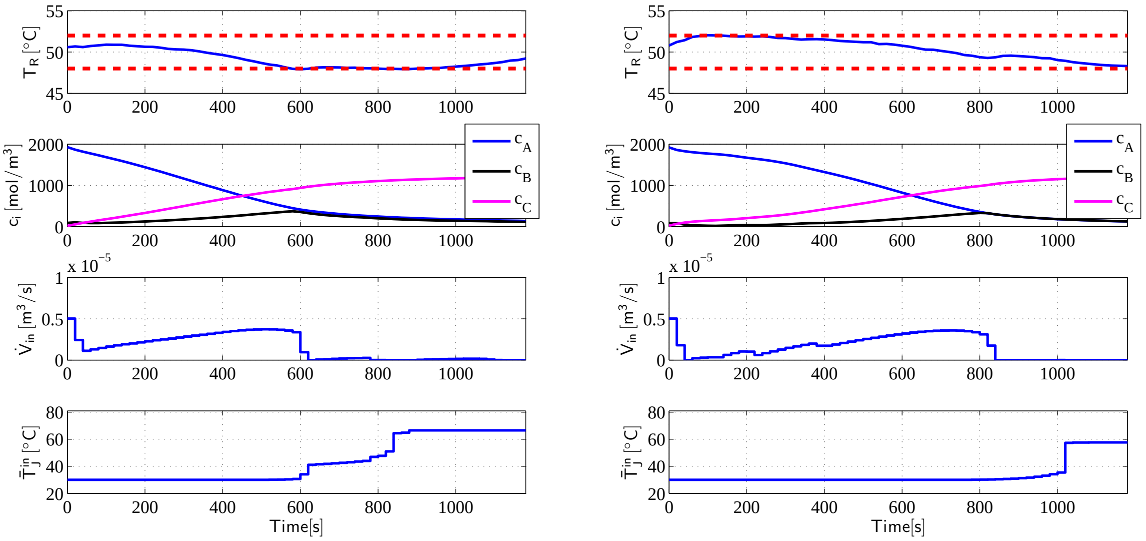

Figure 10.8: Experimental results for multi-stage NMPC of the CSTR with reaction parameters 25% smaller than the nominal values (left) and with the nominal values (right).

Part IV

Extended Algorithms and Enhancements

Chapter 11

Multi-stage NMPC with Reduced Variability

From a practical point of view, it can be beneficial to operate a system that behaves consistently in the same manner despite the presence of changes in the plant dynamics or of disturbances that act on the plant. For this reason, some control approaches try to achieve a similar control performance of an uncertain system for all the possible cases of the uncertainty. In some of the formulations of tube based-MPC, as the one presented in (Mayne et al., 2011), the variability of the different trajectories can be influenced by tuning a parameter that controls how strong the influence of the ancillary controller is. Using a high value of this parameter leads to less variability, but also to a reduction of the average performance.

This chapter presents results, which have been published in (Lucia et al., 2014b), in which the idea of the reduction of the variability is followed by introducing a novel formulation of the multi-stage NMPC approach. The general cost function defined in (3.5) is modified by introducing an additional term which penalizes the deviations between the scenarios in the state space. The new optimization problem can be written as:

$$\min_{x_k^j, u_k^j \forall (j,k) \in I} \quad \sum_{i=1}^{N} \omega_i J_i(X_i, U_i) + k_{\text{var}} \left(\sum_{k=1}^{N_P} \sum_{j=1}^{T_k - 1} (x_k^j - x_k^{j+1})^2 \right) \tag{11.1a}$$

subject to:

$$x_{k+1}^j = f(x_k^{p(j)}, u_k^j, d_k^{r(j)}), \qquad \forall\, (j, k+1) \in I, \tag{11.1b}$$

$$g(x_{k+1}^j, u_k^j) \leq 0\ , \qquad \forall\,(j, k+1) \in I, \tag{11.1c}$$

$$u_k^j = u_k^l \text{ if } x_k^{p(j)} = x_k^{p(l)} \qquad \forall\,(j, k), (l, k) \in I, \tag{11.1d}$$

where $k_{\text{var}} \in \mathbb{R}$ is a tuning parameter that controls the trade-off between robust economic performance and variability of the different scenarios. The number of nodes in the scenario tree at the stage k is denoted by T_k. The new term introduced in (11.1a) penalizes (with a quadratic term) the deviations between two *neighboring* scenarios along the prediction horizon N_P for all the scenarios. Note that the presented approach is not a tube-based approach, although it was inspired by tube-based ideas. There are three main differences of the presented approach compared to the usual tube-based MPC approaches. Firstly, tube-based approaches guarantee that the real system remains in a robust positively invariant set around some trajectory, which is not guaranteed by the proposed method. Secondly, the use of the typical structure consisting of a nominal MPC and an ancillary controller is substituted by a single optimizing controller with a multi-objective cost function (11.1a). Thirdly, the trajectories of the system are not forced to track the nominal trajectory, but they are free to stay close around any trajectory which is optimal in average for the different scenarios in the tree.

This formulation is comparable to minimizing an approximation of the sensitivities of the states with respect to the uncertain parameters. It is important to note that in the multi-stage approach the sensitivities are indirectly computed when predicting over all the scenarios and therefore it is not necessary to compute the sensitivities explicitly, which might be computationally expensive. In this way it is possible to use the scenarios for a two-fold purpose. Firstly, to guarantee robust constraint satisfaction, and secondly to reduce the variability of the system in the presence of uncertainties.

This new approach was applied to the industrial batch polymerization reactor presented in Section 8.2 using the same tuning parameters and the same solution method. The results obtained with the approach proposed in (11.1) are shown in Fig. 11.1 for different values of k_{var}. The results show that by increasing k_{var} the state trajectories are closer to each other for the different values of the uncertainties, obtaining also smoother control inputs, but the batch times become much higher.

We define the variability obtained by the controller as the difference of each trajectory to the average trajectory summed up over the batch time

$$\text{variability} = \sum_{i=1}^{N} \frac{1}{K_i^{\text{f}}} \sum_{k=1}^{K_i^{\text{f}}} \frac{|x_k(i) - x_k^{\text{av}}|}{x_k^{\text{av}}}, \tag{11.2}$$

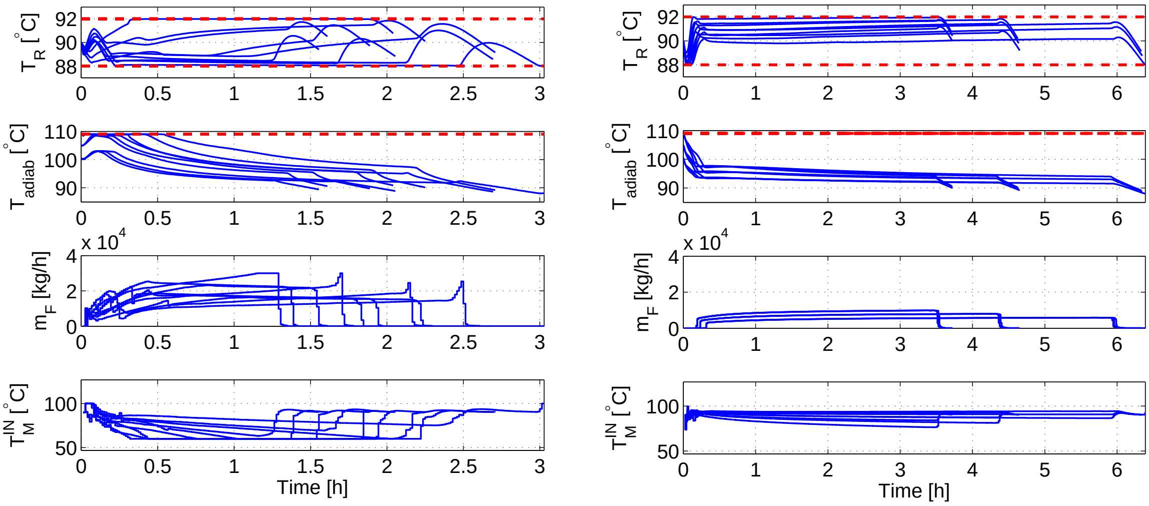

Figure 11.1: Reactor temperature, safety temperature (with constraints indicated), monomer feed and jacket temperature for multi-stage tube-based NMPC with $k_{\text{var}} = 0.1$ (left) and $k_{\text{var}} = 1$ (right) in the economic cost function (11.1a).

where $x_k(i)$ represents the real state of the plant at sampling time k for the scenario (realization of the uncertainty) i and all the operators are applied element-wise. The number of sampling times that are necessary to end the batch for each scenario are denoted as K_i^{f} and x_k^{av} is the average state of the real system at sampling time k over all the scenarios, that is:

$$x_k^{\text{av}} = \frac{1}{N} \sum_{i=1}^{N} x_k(i). \tag{11.3}$$

Using this definition it is possible to perform an analysis of the closed-loop performance by comparing the variability obtained by the controller and the resulting batch times for different values of k_{var}. This analysis is shown in Fig. 11.2. As it can be seen, trying to reduce the variability (increasing k_{var}) of the system causes a significant loss of the economic performance of the process.

Reducing the variability of the system with respect to a certain trajectory might be a good idea if the cost function is a classical tracking term, since both goals go in the same direction, that is, if the nominal trajectory tracks the set-point well and if the uncertain trajectories are near the nominal one, then the average performance will be good. However, if an economic cost function is used, very often the objective of a reduced variability and of an optimal economic performance are contradictory. This can be clearly

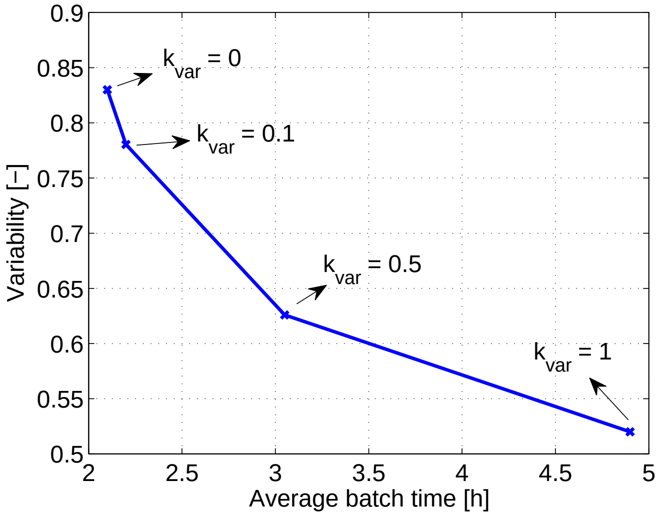

Figure 11.2: Average batch time and variability over the different scenarios for different values of k_{var}.

illustrated with a simple example used often in the MPC literature. If a driver wants to track the center of a road and the driver wants to minimize the variability of his trajectory despite the uncertain conditions of the road, the solution of this problem is to drive very slowly such that the center of the road can be tracked well for all the conditions of the road. This result will be good if the ultimate goal is just to track the center of the road. However, this solution will provide very poor results if the real goal is to drive a certain distance in the minimum time possible fulfilling the constraints for all the cases of the uncertainty. The same happens in the presented case study, where the result with a high value of k_{var} (Fig. 11.1 (right)) leads to slow feeding of the monomer $\dot{m}_{\mathrm{F}}$ in order to achieve low variability, leading to very long batch times.

Note that it is also possible to choose k_{var} such that only the variability of certain states (or other algebraic variables) is penalized. It is also possible to choose $k_{\mathrm{var}} < 0$ in order to excite the system. This can be useful in the context of Optimal Experiment Design (OED) to increase the identifiability of certain parameters with respect to the available measurements. OED can be also used to reduce the range of the uncertainty and to generate a new scenario tree accordingly that leads to improved performance as it will be shown in Chapter 13.

The goal of this chapter is not to compare different tube-based approaches, but to illustrate with simulation results the fact that, in the same way as the concepts and tools that are needed for stability guaranteeing formulations of economic MPC are different to those of classical tracking MPC (Amrit et al., 2011), some typical approaches used in robust tracking MPC that try to minimize the variability of the system may not be suitable in the case of economic MPC because they might lead to significant losses of performance.

The powerful tools of set theory used in some tube-based methods for establishing stability and constraint guarantees could be combined with the multi-stage approach in order to make use of the advantages of both approaches by considering, for example, *slim* tubes centered around the different scenarios of the scenario tree.

Chapter 12

Multi-stage NMPC with Verified Robust Constraint Satisfaction

One of the challenges of the multi-stage NMPC approach is that the use of a discrete set of scenarios does not provide robust guarantees if the actual realization of the uncertainty is not explicitly contained in the scenario tree. This chapter presents an approach to deal with this challenge in an efficient way based on the computation of reachable sets. The proposed strategy is similar to the robust NMPC methods based on reachable sets presented e.g. in (Limon et al., 2005) using interval arithmetics or the one presented in (Bravo et al., 2006) using zonotopic inclusions. The proposed approach possesses two main advantages with respect to the cited works thanks to the use of multi-stage NMPC. Firstly, due to the introduction of recourse on the scenario tree, a nonlinear closed-loop control law without a fixed structure is obtained, which reduces the conservativeness of the approach. Secondly, the partition of the parameter space that can be achieved by adding more branches to the scenario tree helps to reduce the over-approximation introduced by the different bounding techniques (at the cost of a higher computational effort) that are necessary for the computation of the reachable sets. Using an efficient implementation it is shown that it is possible to solve the multi-stage NMPC problem with guaranteed robust constraint satisfaction in real-time for the industrial batch polymerization reactor presented in Section 8.2.

The main contribution of this chapter is the introduction of a guarantee of robust constraint satisfaction for all values of the uncertainty, including those that are not present in the scenario tree. The focus of this work is on the guaranteed robust constraint satisfaction and not the analysis of the stability and recursive feasibility of the approach which are only checked by simulations. To simplify the notation during this chapter, polytopic constraints on the states and on the inputs are considered for the NMPC problem:

$$\min_{x_k^j, u_k^j \forall (j,k) \in I} \quad \sum_{i=1}^{N} \omega_i J_i(X_i, U_i), \tag{12.1a}$$

subject to:

$$x_{k+1}^j = f(x_k^{p(j)}, u_k^j, d_k^{r(j)}), \qquad \forall\, (j, k+1) \in I, \tag{12.1b}$$

$$x_k^j \in \mathbb{X}\,, u_k^j \in \mathbb{U}\,, \qquad \forall\, (j,k) \in I, \tag{12.1c}$$

$$u_k^j = u_k^l \text{ if } x_k^{p(j)} = x_k^{p(l)} \qquad \forall\, (j,k), (l,k) \in I. \tag{12.1d}$$

The work presented in this chapter has been done in collaboration with Radoslav Paulen and has been published in (Lucia et al., 2014d).

12.1 Computing the Reachable Sets

The computation of reachable sets for dynamic nonlinear systems has been an active field of research in the last years. Reachable sets have been used in the context of many different fields such as MPC (Bravo et al., 2006), global optimization (Singer and Barton, 2006), hybrid systems verification (Althoff and Krogh, 2014), verified solution of ODEs (Sahlodin and Chachuat, 2011) or for parameter estimation (Jaulin and Walter, 1993), (Paulen et al., 2013b).

The techniques presented in this section are defined and applied in a continuous-time setting in order to avoid any issues connected with validated numerical integration of the ODEs (Sahlodin and Chachuat, 2011) at this stage. Consider an initial value problem (IVP) for $t \geq t_k$ for the set of parametric ODEs

$$\dot{x}(t) = f^c(x(t), u_k, d_k), \qquad x(t_k) \in \mathcal{X}_k. \tag{12.2}$$

The reachable set of the state variables over the time horizon $[t_k, t_{k+1}]$ is defined as

$$\mathcal{X}_{k+1} := \{x(t_{k+1}) \,|\, \text{Eq. (12.2)}, u_k \in \mathbb{U}, \forall d_k \in \mathcal{D}\}, \tag{12.3}$$

where $\mathcal{D}$ represents a continuous set of possible uncertainties in contrast to the discrete set of realizations used in the formulation of the scenario tree $\mathbb{D}$. In general it is not possible to obtain an exact characterization of the set $\mathcal{X}_{k+1}$. Consequently, one must rely on approximation techniques that compute an over-approximation $\bar{\mathcal{X}}_{k+1}$ that is guaranteed to contain the reachable set, $\mathcal{X}_{k+1} \subseteq \bar{\mathcal{X}}_{k+1}$. Such techniques exploit the propagation-in-time of convex or non-convex envelopes that enclose the state-space evolution of $x(t)$ over the possible realizations of d.

Many different methods for the computation of the over-approximation of reachable sets have been presented in the literature. This section describes two techniques for enclosing the solution set of parametric ODEs that are exploited later in this chapter.

12.1.1 Interval Bounds

Interval bounds for (12.3) can be computed by application of the following classical results from the theory of differential inequalities presented in (Walter, 1970).

Consider the IVP (12.2), where $f^c : \mathbb{Z} \times \mathbb{U} \times \mathcal{D} \to \mathbb{R}^{n_x}$ is a continuous vector function that satisfies a uniqueness condition on $\mathbb{Z} \times \mathbb{U} \times \mathcal{D}$, with $\mathbb{Z} \subset \mathbb{R}^{n_x}$. Let the functions $x^L, x^U : \mathbb{R} \to \mathbb{R}^{n_x}$ be continuous on some open set containing $[t_k, t_{k+1}]$ and satisfy $[x^L(t), x^U(t)] \subset \mathbb{Z}$ for all $t \in [t_k, t_{k+1}]$. If for all $\mathcal{X}_k \subseteq [x^L(t_k), x^U(t_k)]$ and

$$\dot{x}_i^{\mathrm{L}}(t) = \min_{z,d} \left\{ f_i^c(z, u_k, d_k) \,\middle|\, \begin{matrix} z_i = x_i^{\mathrm{L}}(t), d \in \mathcal{D} \\ z \in [x^{\mathrm{L}}(t), x^{\mathrm{U}}(t)] \end{matrix} \right\}, \tag{12.4}$$

$$\dot{x}_i^{\mathrm{U}}(t) = \max_{z,d} \left\{ f_i^c(z, u_k, d_k) \,\middle|\, \begin{matrix} z_i = x_i^{\mathrm{U}}(t), d \in \mathcal{D} \\ z \in [x^{\mathrm{L}}(t), x^{\mathrm{U}}(t)] \end{matrix} \right\}, \tag{12.5}$$

for some $u \in \mathbb{U}$, for almost all $t \in [t_k, t_{k+1}]$ and $i = 1, \ldots, n_x$, then $x(t) \in [x^L(t), x^U(t)]$ for all $(t, d) \in [t_k, t_{k+1}] \times \mathcal{D}$.

In practice, bounds on the right-hand sides of the differential inequalities can be obtained via natural interval extensions. The main idea is to decompose any function in elementary operations as for the case of automatic differentiation presented in Chapter 5, replacing the variables for their corresponding intervals and applying the rules of interval arithmetic. Although simple to implement, the bounds obtained using natural interval extensions are usually very conservative, i.e., the over-approximation of the true reachable set is large. One of the main reasons for this is the so called wrapping effect, which occurs because the actual reachable set is approximated by a box-shaped set and wrapped through time. The wrapping effect can be mitigated in several ways, for example by representing the reachable set using zonotopes instead of intervals, as presented in (Kühn, 1998) and applied in the context of MPC in (Bravo et al., 2006).

Another important challenge of these methods is the dependency problem that occurs when a variable enters several times in an expression, which cannot be detected by interval arithmetic resulting in unnecessarily large over-approximations. A possibility to avoid this problem which has been extensively studied in the last years is the use of Taylor models as described in (Makino and Berz, 1996) and in (Lin and Stadtherr, 2007).

A Taylor model can be constructed by a truncated Taylor series, and it is composed of the polynomial part and the interval remainder. The interval remainder is then bounded using techniques such as interval arithmetic. An alternative to the use of interval arithmetics, is to use convex/concave relaxations as presented for example in (Sahlodin and Chachuat, 2011) for the computation of tight bounds of the remainder of the Taylor models.

A different possibility for the computation of reachable sets is the use of ideas of ellipsoidal calculus as presented in (Houska et al., 2012), which are shortly described in the next section.

12.1.2 Ellipsoidal-based Bounds

Given a positive semi-definite matrix $Q \in \mathbb{S}_+^{n_x}$ and a vector $c \in \mathbb{R}^{n_x}$, the set

$$\mathcal{E}(Q,c) = \{c + Q^{\frac{1}{2}} v \mid \exists v \in \mathbb{R}^{n_x} : v^\mathsf{T} v \leq 1\}, \tag{12.6}$$

defines an n_x-dimensional ellipsoid centered at c and with shape matrix Q. As presented in (Houska et al., 2012), an ellipsoidal reachable set of a nonlinear parametric ODE can be calculated as explained below.

Consider the IVP (12.2), where $f : \mathbb{Z} \times \mathbb{U} \times \mathcal{D} \rightarrow \mathbb{R}^{n_x}$ is a $\mathcal{C}^2$ vector function, with $\mathbb{Z} \subset \mathbb{R}^{n_x}$. Let $Q : [t_k, t_{k+1}] \rightarrow \mathbb{S}_+^{n_x}$, $S : [t_k, t_{k+1}] \rightarrow \mathbb{R}_+^{n_x \times n_d}$ and $x^* : [t_k, t_{k+1}] \rightarrow \mathbb{R}^{n_x}$, and define $\mathtt{X}(t)$ such that

$$\mathtt{X}_i(t) = \sqrt{Q_{ii}(t)}[-1,1] \oplus S_i(t)[\mathcal{D} - d^*] \oplus x_i^*(t), \tag{12.7}$$

for each $i = \{1, \ldots, n_x\}$ and all $t \in [t_k, t_{k+1}]$ where $\oplus$ is the Minkowski sum. Suppose that Q, S and x^* satisfy

$$\dot{Q} \succeq \frac{\partial f}{\partial x}(x^*, u_k, d^*)\, Q + Q\, \frac{\partial f}{\partial x}(x^*, u_k, d^*)^\mathsf{T}$$

$$+ \frac{\mathbf{1}^{\mathsf{T}}\mathrm{rad}\, R_t}{\sqrt{\mathrm{tr}\, Q}}\, Q + \mathrm{diag}(\mathrm{rad}\, R_t)\, \sqrt{\mathrm{tr}\, Q}, \tag{12.8}$$

$$\dot{S} = \frac{\partial f}{\partial x}(x^*, u_k, d^*)\, S + \frac{\partial f}{\partial d}(x^*, u_k, d^*), \tag{12.9}$$

$$\dot{x}^* = f(x^*, u_k, d^*), \tag{12.10}$$

for some $u_k \in \mathbb{U}$, for every $t \in [t_k, t_{k+1})$, as well as the initial conditions $Q(t_k) = \mathrm{diag}(\mathrm{rad}\, \bar{\mathcal{X}}_k)^2$, $S(t_k) = S_k$ and $x(t_k) = \mathrm{mid}\, \bar{\mathcal{X}}_k$, where R_t is the remainder term of a 1st-order Taylor model (Sahlodin and Chachuat, 2011) of $f(\cdot)$ on $\mathtt{X}(t) \times \mathbb{U} \times \mathcal{D}$ at $(x^*(t), u_k, d^*)$, $\mathrm{rad}(\cdot)$ is the radius of an interval and $\mathrm{mid}(\cdot)$ is the center point of an interval. Then, $x(t) \in \mathtt{X}(t)$, $\forall (t, d) \in [t_k, t_{k+1}] \times \mathcal{D}$ and the over-approximated reachable set is $\bar{\mathcal{X}}_{k+1} = \mathtt{X}(t_{k+1})$. For the presented method, the influence of the wrapping effect is mitigated as compared to the method of differential inequalities that results in generally tighter enclosures of computed reachable sets. This is obtained at the price of increased computational time that arises from the quadratical growth of the number of auxiliary states (Q, S) for increasing n_x.

12.2 Verified Robust Constraint Satisfaction using Multi-stage NMPC

Usually rigorous state bounding techniques are embedded into an optimization problem in order to provide envelopes of the realizations of the state trajectories under the uncertainty. These envelopes, propagated in time, are required to satisfy the constraints along the prediction horizon as presented in (Limon et al., 2005; Bravo et al., 2006). Note that such state bounds represent non-differentiable functions which makes the convergence of the resulting optimization problem computationally very demanding. Moreover the reachable sets are constructed throughout the solution of the optimization problem even for the sub-optimal control inputs which further and unnecessarily increases the computational time. In order to avoid the dramatic increase of the computational burden, the multi-stage NMPC problem is solved recursively introducing time- and scenario-dependent back-off constants whose values are updated based on the actual form of the reachable set.

The construction of the scenario tree is done under certain assumptions to make the optimization problem tractable. First, the realization of the uncertainty is considered to be constant over a sampling period. Moreover, the notion of robust horizon explicitly imposes that the realization of the uncertainty remains constant after a certain point in time. Such assumptions are inconsistent within the framework of rigorous state bounding techniques. Two modifications in the scenario tree are introduced here to achieve a more realistic representation of the uncertainty so that the scenario tree representation can make use of the information provided by the employed bounding techniques in a more efficient manner.

First, to avoid the optimistic assumption that the control inputs can be adapted after the robust horizon to each uncertainty realization independently, all control inputs coming from the same parent node at the stage at which the tree stops branching are constrained to be equal at each sampling time until the end of the prediction horizon. This constraint is imposed by (12.11e). Secondly, for the calculation of the reachable sets it is necessary to provide the bounding algorithm with a parameter box $\mathcal{D}$. As the values of the uncertainty are considered as sampled in the scenario tree, there exist a non-unique parameter box $\mathcal{D}^i$ that contains the realizations of the uncertainty from $\mathcal{D}$ around a particular scenario. Consequently, this parameter box can be chosen such that $\mathcal{D} \subseteq \cup_{\forall i} \mathcal{D}^i$. We incorporate this degree of freedom to the optimization problem by the introduction of new optimization variables $d_k^{\mathrm{c}(j)}$ that determine the branching points between two neighboring realizations ($d_k^{r(j)}$ and $d_k^{r(j+1)}$) and give the ranges of the uncertainty that will be used to compute the over-approximated reachable sets. An illustration of this modification can be seen in Fig 12.1. The new states ($x_{k+1}^{\mathrm{c}(j)}$) are calculated with the inputs of both neighbor (original) branches (denoted as $u_k^{n(j)}$) for the realization $d_k^{\mathrm{c}(j)}$. The newly occurring indices are summarized in I^{c} and they are also subject to constraints as indicated in (12.11c) and (12.11g). The modified optimization problem solved at each sampling time for a multi-stage NMPC with verified constraint satisfaction reads as:

$$\min_{\substack{x_{k+1}^j, u_k^j, d_k^{\mathrm{c}(j)}, \forall (j,k) \in I \\ x_{k+1}^{\mathrm{c}(j)}, \forall (j,k) \in I^c}} \quad \sum_{i=1}^{N} \omega_i J_i(X_i, U_i), \tag{12.11a}$$

subject to:

$$\begin{aligned}
x_{k+1}^j &= f(x_k^{p(j)}, u_k^j, d_k^{r(j)}) \,, & \forall\, (j, k+1) \in I, \quad & (12.11b)\\
x_{k+1}^{\mathrm{c}(j)} &= f(x_k^{p(c(j))}, u_k^{n(j)}, d_k^{\mathrm{c}(j)}), & \forall (j, k+1) \in I^{\mathrm{c}}, \quad & (12.11c)\\
u_k^j &= u_k^l \text{ if } x_k^{p(j)} = x_k^{p(l)} \,, & \forall\, (j,k), (l,k) \in I, \quad & (12.11d)
\end{aligned}$$

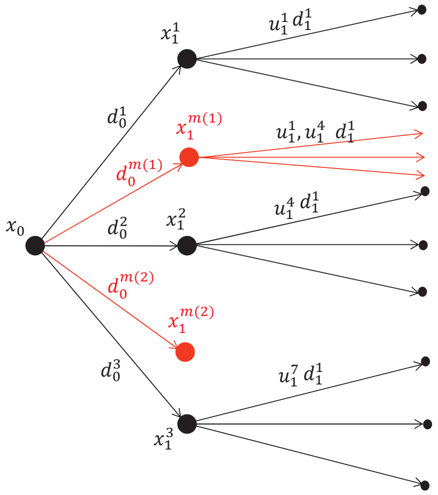

Figure 12.1: Illustration of the addition of a middle branch to the scenario tree (indicated in red) which is added as an additional optimization variable. The constraints are checked at the next stage using the neighboring control inputs. The same procedure is applied to all the nodes in the tree.

$$u_k^j = u_k^l \text{ if } x_{N_r}^{p(j)} = x_{N_r}^{p(l)}\ , \qquad \forall\,(j,k),(l,k) \in I, \tag{12.11e}$$

$$x_k^j \oplus \mathcal{B}_0(\epsilon_k^j) \subseteq \mathbb{X}, u_k^j \in \mathbb{U}\ , \qquad \forall\,(j,k) \in I, \tag{12.11f}$$

$$x_k^{c(j)} \oplus \mathcal{B}_0(\epsilon_k^j) \subseteq \mathbb{X}, \qquad \forall\,(j,k) \in I^c, \tag{12.11g}$$

$$d_k^{r(j)} \leq d_k^{c(j)} \leq d_k^{r(j+1)}, \qquad \forall\,(j,k) \in I, \tag{12.11h}$$

where $\mathcal{B}_0(r) = \{b \in \mathbb{R}^{n_x} | \|b\|_\infty \leq r\}$ and ϵ_k^j denotes a tightening factor of the original constraints. We propose to use a systematic approach to the simultaneous tightening and solution of problem (12.11). For this purpose we make use of the information contained in the scenario tree. It is possible to approximate the parametric sensitivities of the states $\frac{\partial x}{\partial d}$ by comparing *neighbor* states in the scenario tree, i.e.:

$$s_{k+1}^j := \frac{\partial x_{k+1}^j}{\partial d_k^{r(j)}} \approx \frac{x_{k+1}^j - x_{k+1}^{j+1}}{d_k^{r(j)} - d_k^{j+1}} = \tilde{s}_{k+1}^j. \tag{12.12}$$

Once the approximated sensitivities are obtained, this information can be used to approximate the reachable set, exploiting the choice of the range of uncertainty at the same time at which the problem (12.11) is solved, by computing

$$\gamma_k^j := \tilde{s}_{k+1}^j (d_k^{r(j)} - d_k^{c(j)}), \tag{12.13}$$

which is an approximation of the second term on the right-hand side of (12.7). We use $\epsilon_k^j = \rho\,\gamma_k^j$ as the automatic tightening of problem (12.11), where ρ is a tuning parameter. If after solving (12.11) the reachable sets of any scenario i at any time stage k ($\bar{\mathcal{X}}_k^i$) violate the original constraints, we increase ρ and solve problem (12.11) again. We introduce the following assumptions to prove that it is possible to achieve an NMPC controller with verified robust constraint satisfaction.

Assumption 5 (Boundedness of $\bar{\mathcal{X}}_k^i$)**.** *We assume that the distance between the over-approximated reachable set $\bar{\mathcal{X}}_k^i$ and any node of the scenario tree x_k^j which belongs to scenario S_i is bounded for all $k = 1, \ldots, N_p$ and for any sequence of admissible control inputs so that:*

$$\bar{\mathcal{X}}_k^i \subseteq x_k^j \oplus \mathcal{B}_0(L(rad\,\mathcal{D}^i)), \tag{12.14}$$

where $L(rad\,\mathcal{D}^i)$ is the radius of the ball which bounds the over-approximated reachable set. The radius of the ball depends on the size of the parameter box $\mathcal{D}^i$.

Assumption 6 (Feasible solution)**.** *For a given set of parameter boxes $\mathcal{D}^i$ with $i = 1, \ldots, N$, the optimization problem* (12.11) *remains feasible using a tightening of the constraints $\mathcal{B}_0(\epsilon_k^j) = \mathcal{B}_0(L(rad\,\mathcal{D}^i))$.*

The complete algorithm for multi-stage NMPC with verified robust constraint satisfaction is described in Algorithm 2.

Algorithm 2 Multi-stage NMPC with verified robust constraint satisfaction

```
REQUIRE Initial condition $x_0^1$; $\rho > 0$;
1. Solve problem (12.11) with $\epsilon_k^j = \rho\gamma_k^j$.
    1.1 if (12.11) is feasible: GOTO 2.
    1.2 else: reduce $\rho$ and GOTO 1.
2. Calculate the reachable sets until the end of the prediction horizon $(\bar{\mathcal{X}}_1^i, \bar{\mathcal{X}}_2^i, ..., \bar{\mathcal{X}}_{N_p}^i)$
   for each scenario $S_i$ given the optimal control inputs and the ranges of the uncer-
   tainty calculated in 1.
3. if $\bar{\mathcal{X}}_k^i$ satisfy the constraints with $k = 1, \ldots, N_p$ and $i = 1, \ldots, N$:
    3.1 Apply $u_0^1$ and wait until next sampling time.
    3.2 Take the new measurement $x_0^1$ and GOTO 1.
4. else: Increase $\rho$ and GOTO 1.
```

Theorem 4. *If Assumptions 5 and 6 hold, then Algorithm 2 provides an NMPC algorithm that guarantees the satisfaction of the constraints for all the possible realizations of the uncertainty $d \in \mathcal{D}$.*

Proof. If problem (12.11) is feasible then for any node x_k^j in the tree belonging to any scenario S_i, it follows that $x_k^j \oplus \mathcal{B}_0(\epsilon_k^j) \subseteq \mathbb{X}$. Dilating both sets we have: $x_k^j \oplus \mathcal{B}_0(\epsilon_k^j) \oplus \mathcal{B}_0(L(\mathrm{rad}\,\mathcal{D}^i)) \subseteq \mathbb{X} \oplus \mathcal{B}_0(L(\mathrm{rad}\,\mathcal{D}^i))$. Then using Assumption 5, it follows that $\bar{\mathcal{X}}_k^i \oplus \mathcal{B}_0(\epsilon_k^j) \subseteq \mathbb{X} \oplus \mathcal{B}_0(L(\mathrm{rad}\,\mathcal{D}^i))$. By increasing ϵ_k^j (and decreasing if (12.11) becomes infeasible) at each iteration of Algorithm 2, a value of $\epsilon_k^j \leq L(\mathrm{rad}\,\mathcal{D}^i)$ (element-wise) can be found so that the optimization problem (12.11) is feasible because of Assumption 6 and all the resulting reachable sets satisfy the original constraints. Then the constraints are satisfied for all values of the uncertainty and Theorem 4 is proven. □

Remark 4. *Assumption 5 is a mild assumption that only requires that the bounding procedure does not explode, giving thus bounded over-approximations. Assumption 6 might be difficult to verify in general. However, the bound $L(rad\,\mathcal{D}^i)$ can be reduced arbitrarily* *by adding more branches to the scenario tree and thus reducing the size of the parameter boxes $\mathcal{D}^i$, which will facilitate the fulfillment of Assumption 6. Note that it is not necessary to know explicitly the bound, but only to assume its existence.*

12.3 Case Study

The algorithm presented in the previous section is applied to the industrial batch polymerization reactor described in Section 8.2. For simplicity, only one parameter, the reaction enthalpy ΔH_{R}, is considered to be uncertain and it takes values that vary by $\pm 30\%$ with respect to its nominal value $\Delta H_{\mathrm{R}}^{\mathrm{nom}} = 950$ kJ/kg. It is checked by simulations that the bounding procedure provides tight bounds that satisfy Assumptions 5 and 6 for this example.

The optimization problems are solved using CasADi to provide first and second order derivatives to IPOPT. The dynamics of the system are discretized using orthogonal collocation on finite elements as explained in Chapter 5. The reachable sets are calculated using the library MC++ (`http://projects.coin-or.org/MCpp`), a C++ library employing operator overloading for bounding of factorable functions. Moreover, the numerical integrator available through the GNU Scientific Library (GSL) is used. The C++ class implementing the various bounding techniques based on differential inequalities, ellipsoidal-based bounds and other techniques is freely available at: `http://www3.imperial.ac.uk/environmentenergyoptimisation/software`.

Fig. 12.2 shows the predicted scenario trees (solid black lines) and the obtained bounds for each scenario for a comparison of the different methods studied in this paper using a prediction horizon $N_P = 8$ and a robust horizon $N_R = 3$. Fig. 12.2a shows the predicted trees and the bounds over the prediction horizon at the initial time for the case of the modified scenario tree obtained by solving (12.11) with $\epsilon_k^j = 0$ (i.e. no initial tightening of the constraints); the reachable sets are calculated using interval bounds. As it can be seen the scenario tree is a good approximation of the reachable set but the over-approximation introduced by the bounding technique causes constraints violations on the reactor temperature constraint (T_{R}). The same situation occurs when ellipsoidal bounds are used for the calculation of the reachable sets, but as expected the over-approximation is smaller as it can be seen in Fig. 12.2b. In contrast, if the simultaneous shrinking of the constraints using the approximated sensitivities obtained in the scenario tree are

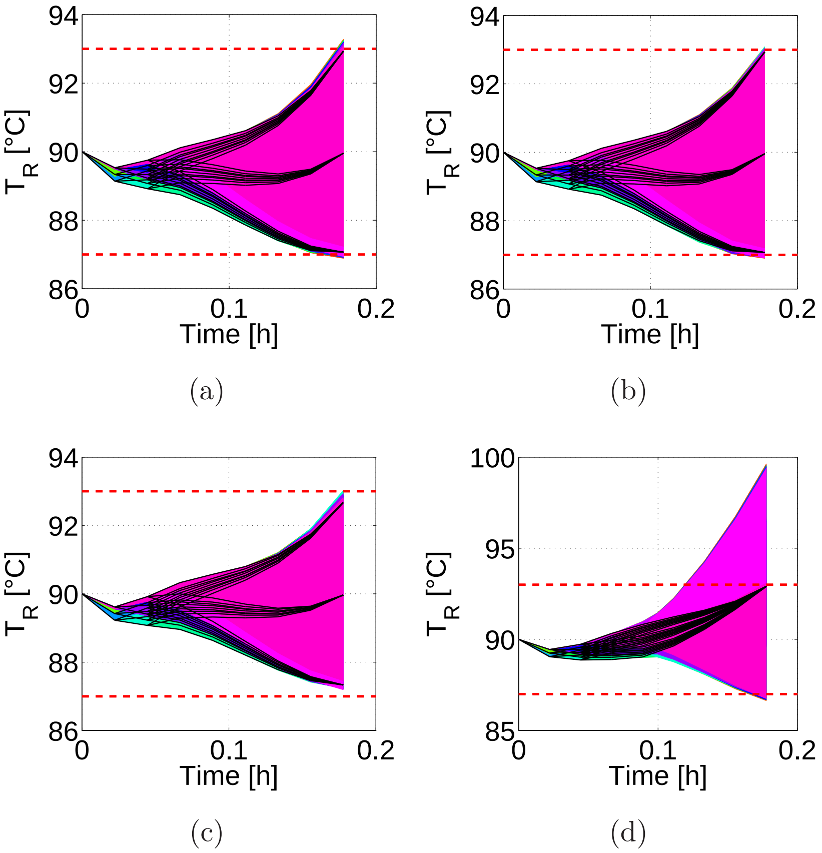

Figure 12.2: Tree predictions and bounds obtained for the reactor temperature T_{R} at the initial time for the modified scenario tree ($N_P = 8$, $N_R = 3$) with interval bounds and $\epsilon = 0$ (Fig. 12.2a), with ellipsoidal bounds and $\epsilon = 0$ (Fig. 12.2b) and with interval bounds and ϵ based on the approximated sensitivities (Fig. 12.2c). Fig. 12.2d is obtained with the original scenario tree and interval bounds.

used together with interval bounds as bounding technique, no constraint violations occur even in the iteration 0 of Algorithm 2 (see Fig. 12.2c). Fig. 12.2d shows the same results obtained by solving the original multi-stage problem (3.5). As it can be seen, the scenario tree is a good approximation of the reachable sets until the robust horizon $N_R = 3$ but then the computed over-approximation and the scenario tree are very different. It is important to note that this does not mean that the real use of this controller will result in constraint violations. It rather shows that the assumption made after the robust horizon – the uncertainty remains constant – causes that the control inputs are independently adapted for each scenario. When they are used to calculate the reachable sets for all the possible values of the uncertainty, they produce very different bounds compared to the scenario tree because they were calculated for a single and constant value of the uncertainty. This problem can be significantly reduced by increasing the robust horizon N_R. This analysis can be useful to test the quality of a given scenario tree.

Next, the scenario tree used in Fig. 12.2c (with $N_P = 6$) is chosen and the controller is tested for random values of the uncertainty parameters. 100 batch runs are considered for random values of $\Delta H_\mathrm{R} \sim \mathcal{N}(\mu, \sigma) = \mathcal{N}(\Delta H_\mathrm{R}^\mathrm{nom}, 0.15 \cdot \Delta H_\mathrm{R}^\mathrm{nom})$ truncated at 2σ. The uncertain parameter is maintained constant during each run, and the obtained batch times are reported in Fig. 12.3. As it can be seen, multi-stage NMPC with verified constraint satisfaction achieves a significantly better performance than open-loop NMPC with verified constraint satisfaction and worse performance than multi-stage NMPC without robust constraint satisfaction guarantee. It is important to note that if the ellipsoidal bounds are used or especially if the robust horizon N_R is increased, the conservativeness of the approach can be further reduced. In this way, N_R provides a trade-off between computational effort and conservativeness w.r.t the non-guaranteed case, but even for small values of N_R, the proposed method performs significantly better than an open-loop NMPC approach. The reactor temperature trajectory T_R as well as the control inputs $\dot{m}_\mathrm{F}$, T_M^IN for the case of $\Delta H_\mathrm{R} = 1142.05$ kJ/kg are presented in Fig. 12.4. The average computation time per NMPC iteration is of 3.52 seconds. The sampling time is $t_\mathrm{step} = 80$ s which makes it possible to apply the proposed scheme in real-time. Only 0 or 1 iterations of the proposed Algorithm 2 were needed at each sampling time in the presented case study.

This chapter presented an extension of multi-stage NMPC for the verification of robust constraint satisfaction for all possible values of the uncertainty within given bounds. The use of a scenario tree makes it possible to introduce feedback in the prediction and to decrease the parameter range that the bounding techniques have to take into account. This leads to a less conservative solution. The algorithm is computationally demanding, and might be difficult to solve for a high cardinality of the set of uncertain parameters, or for problems where bounding techniques provide very conservative results. On the other hand, the use of efficient tools to solve the optimization and bounding problems made it possible to solve in real-time an industrially relevant problem with a rigorous guarantee of robust constraint satisfaction.

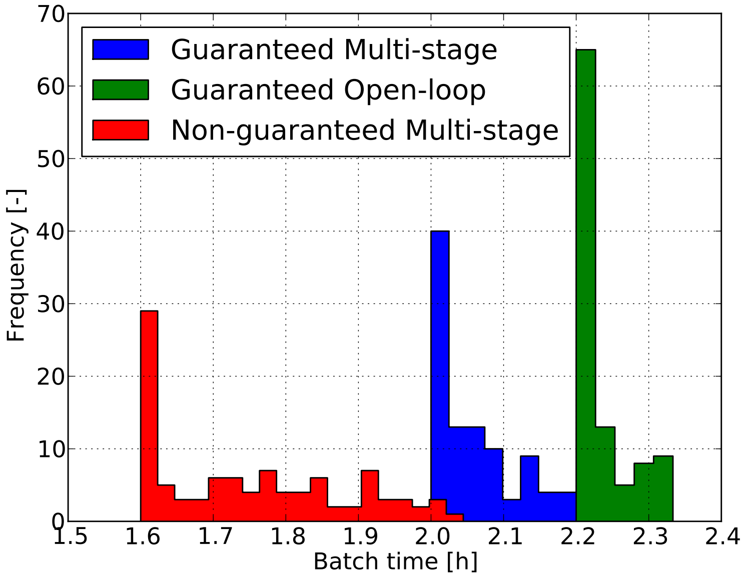

Figure 12.3: Distribution of the batch times obtained for 100 simulations with random values of ΔH_{R} for different controllers.

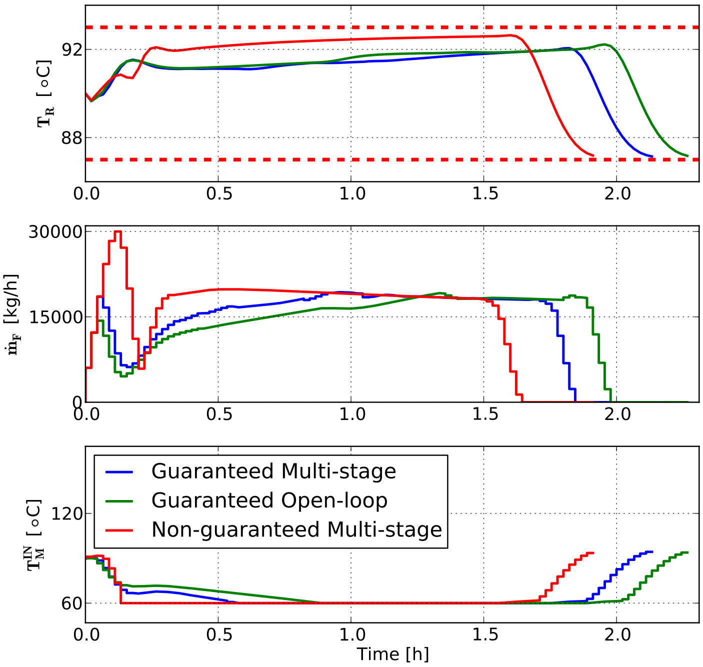

Figure 12.4: Reactor temperature T_{R}, and control inputs $\dot{m}_{\mathrm{F}}$, $T_{\mathrm{M}}^{\mathrm{IN}}$ for the case of $\Delta H_{\mathrm{R}} = 1142.05$ kJ/kg for different controllers.

Chapter 13

Multi-stage NMPC with Reduction of the Uncertainty

The control strategy presented in this thesis is based on the use of a scenario tree which is generated using different realizations $d_k^{r(j)}$ of the uncertainty. A usual strategy to choose the realizations included in the tree is to use the combinations of the lower, upper and middle (or nominal) values of the assumed range of the uncertain parameters. The bounds of the parameters are denoted in this chapter as $D := [d^L, d^U]$. The superscripts ^{L} and ^{U} represent the lower and upper bounds of an interval box and are understood component-wise.

By reducing the range of the parametric uncertainty D, i.e. narrowing the employed scenario tree, a significant improvement can be achieved in terms of the conservativeness of the resulting robustly optimal control input. Dual control, originally proposed in (Feldbaum, 1960) and also studied in (Åström and Wittenmark, 1971), tackles a similar problem.

The aim of dual control is to strike the balance between finding the optimizing inputs for the real control goal (e.g. an economic cost function) and inputs that excite the process sufficiently to reduce the (a posteriori estimated) bounds on the parameter values D. The dynamic programming formulation of the problem (Bertsekas, 2000) is often found computationally intractable for nonlinear systems, but the minimization of the uncertainty can also be achieved via means of optimal design of dynamic experiments.

The goal of this chapter is to study possible improvements of the multi-stage NMPC scheme via a combination of the inputs that optimize a chosen criterion of optimal experiment design and the inputs that optimize an economic criterion in the framework of multi-stage NMPC. The work presented in this chapter has been done in collaboration with Radoslav Paulen and has been published in (Lucia and Paulen, 2014).

13.1 Robust Optimal Dynamic Experiment Design

Optimal dynamic experiment design has been widely used since the seventies of the last century, especially in the field of system identification (see (Gevers et al., 2011) for a review). In general, it can be formulated as the problem of designing the input trajectories to the system (13.1) that generate measurements from which parameters can be identified with the smallest possible uncertainty. As done in the rest of the thesis, a discrete-time nonlinear system is considered. Here also the measurement equations are used, defined by $h : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_y}$ being n_y the number of measurements. The system can be written as:

$$x_{k+1}^{j} = f(x_k^{p(j)}, u_k^j, d_k^{r(j)}), \tag{13.1a}$$

$$y_k^j = h(x_k^j). \tag{13.1b}$$

By applying the chain rule of differentiation to the system equations it is possible to obtain the equations that define the $n_x \times n_p$ matrix of the parametric state sensitivities $s_k^j = \frac{\partial x_k^j}{\partial d_k^{r(j)}}$. The parametric state sensitivity equations at each node of the tree can be calculated for each parameter $m = 1, ..., n_d$ can be written as:

$$s_{k+1}^{j}(m) = \left(\frac{\partial f}{\partial x} s_k^{p(j)}(m) + \frac{\partial f}{\partial d_k^{r(j)}(m)} \right), \tag{13.2}$$

where $s_k^{p(j)}(m)$ denotes the m-th row of the sensitivity matrix and $d_k^{r(j)}(m)$ is the m-th element of the uncertainty vector. In practice the system model is usually given by a set of continuous ODEs, which implies that the sensitivity equations are also continuous. To discretize them, they are collocated using orthogonal collocation on finite elements in the same manner as the system equations. In order to take into account the possible different magnitudes of the parameters, the sensitivities used within this chapter are defined in the fully relative form as suggested by (Munack, 1991). That is, the sensitivities are scaled according to the values of the states and parameters as:

$$\hat{s}_{k+1}^{j}(m) = \frac{d_k^{r(j)}(m)}{x_k^{p(j)}} \left(\frac{\partial f}{\partial x} s_k^{p(j)}(m) + \frac{\partial f}{\partial d_k^{r(j)}(m)} \right), \tag{13.3}$$

where the division by the state $x_k^{p(j)}$ is understood element-wise. For the common case in which the measurement function $h(\cdot)$ is linear, the $n_y \times n_p$ matrix of parametric output sensitivities ($s_{y,k}^j$) can be calculated directly for each row as: $s_{y,k}^j(m) = h(s_k^j(m))$.

The robust optimal dynamic experiment design problem can be formulated via minimization of an appropriate measure of the Fisher information matrix defined for each scenario i of the scenario tree as:

$$F_{N_e}^i(X_i, U_i) := \sum_{k=0}^{N_e-1} \hat{s}_{y,k}^{j\,T} Q \hat{s}_{y,k}^j,\ \forall \hat{s}_{y,k}^j \in S_i \tag{13.4}$$

where N_e stands for the horizon for which the optimal experiment is realized, and Q is the inverse of the covariance matrix of the measurement noise.

Among the several different possible experiment design criteria (Munack, 1991), a modified E-design criterion is chosen here:

$$\phi_{\mathrm{mE}}(F) = \left(\max_l \lambda_l(F)\right) / \left(\min_l \lambda_l(F)\right), \tag{13.5}$$

where λ_l represents l-th eigenvalue of F. This criterion provided good results during the simulation studies even though it is nondifferentiable and therefore it is used in this chapter. Other typical criteria can also be chosen (Franceschini and Macchietto, 2008), some of which are differentiable and can make the resulting optimization problem easier, facilitating the convergence of the NLP solver to a local solution.

The robust optimal experiment design problem can be formulated using the multi-stage formulation:

$$\min_{x_k^j, u_k^j \forall (j,k) \in I} \quad \sum_{i=1}^{N} \omega_i J_i(X_i, U_i), \tag{13.6a}$$

subject to:

$$x_{k+1}^j = f(x_k^{p(j)}, u_k^j, d_k^{r(j)}), \qquad \forall\,(j, k+1) \in I, \tag{13.6b}$$

$$g(x_{k+1}^j, u_k^j) \leq 0\ , \qquad \forall\,(j, k+1) \in I, \tag{13.6c}$$

$$u_k^j = u_k^l \text{ if } x_k^{p(j)} = x_k^{p(l)} \qquad \forall\,(j,k), (l,k) \in I, \tag{13.6d}$$

in which the cost of each scenario (J_i) for the case of robust OED with modified E-design is the one defined in (13.5).

The main contribution of this chapter is the proposal of a new optimal experiment design criterion, which takes into account the effect that a reduction in the uncertainty range has on the robust economic performance. For this purpose, the sensitivities of the optimal robust economic cost in (13.6) ($\tilde{J}^*$) with respect to the uncertainty range ($w(D)$) are calculated when an economic cost function is used as stage cost for each scenario. The uncertainty range $w(D) \in \mathbb{R}^{n_d}$ is a vector that contains the difference between the maximum and the minimum value of each uncertain parameter, so that the scenario tree can be formulated as a function of $w(D)$. These sensitivities can be obtained by computing the NLP sensitivities of the multi-stage problem with an economic cost function, which can be calculated by a linearization of the KKT conditions with a very low computational effort (Pirnay et al., 2012) if no active set changes occur. In the case of active set changes other procedures can be used as described in (Wolbert et al., 1994; Pirnay et al., 2012). This information can be used to design a new OED criterion that tries to achieve a better estimation (reduction of the range of the uncertainty) of those parameters which have a higher influence in the robust economic operation of the system. We propose to use the sensitivities described above $\left(\frac{\partial \tilde{J}^*}{\partial w(D)}\right)$ to scale the Fisher information matrix such that the new criterion is:

$$\phi(F) = \phi_{\mathrm{mE}} \left(\mathrm{diag}^{-1} \left[\frac{\partial \tilde{J}^*}{\partial w(D)} \right] F \; \mathrm{diag}^{-1} \left[\frac{\partial \tilde{J}^*}{\partial w(D)} \right] \right), \tag{13.7}$$

A similar scaling was proposed in (Recker et al., 2013) where the sensitivity of the economic cost is considered w.r.t. a parametric uncertainty. In contrast to their approach, here it is directly taken into account that if parametric uncertainty is present, a robust operation will be needed. As mentioned above, the conservativeness of the robust operation directly depends on the range of the uncertainty. Therefore the potential gain in the robust economic operation w.r.t. reduction in the parameter uncertainty range is used as scaling factor.

The next section presents the strategy used in this chapter for the calculation of the uncertainty range D given noisy measurements of the process output.

13.2 Guaranteed Parameter Estimation

Given a set of output measurements y^{m} at N_e time points $1, \ldots, N_e$, *classical* parameter estimation seeks for *one* particular instance d_{e} of the parameters for which a (possibly weighted) norm of the difference between measurements and the corresponding model outputs y_k at each stage k is minimized. This optimization problem, for instance using the l_2-norm, i.e. in the least-square sense, is given by:

$$d_\mathrm{e} \in \arg\min_{d\in D} \sum_{k=1}^{N_e} \|y_k^\mathrm{m} - y_k\|_2^2, \tag{13.8a}$$

$$\text{subject to:} \tag{13.8b}$$

$$x_{k+1} = f(x_k, u_k, d), \tag{13.8c}$$

$$y_k = h(x_k). \tag{13.8d}$$

The confidence of the parameter estimates subject to measurement noise can then be approximated via an ellipsoidal set. The shaping matrix (variance-covariance matrix of the estimates) of such an ellipsoidal set can be approximated by the inverse of the Fisher information matrix defined in (13.4).

In contrast, *guaranteed* (bounded-error) parameter estimation accounts explicitly for the fact that the true process outputs, y^p, are known to be corrupted by some bounded measurement errors $e \in E := [e^L, e^U]$, so that

$$y_k^\mathrm{p} \in y_k^\mathrm{m} + [e^L, e^U] =: Y_k. \tag{13.9}$$

Here, the main objective is to estimate the set D_e of *all possible* parameter values d such that $y_k \in Y_k$ for every $k = 1, \ldots, N_e$; that is,

$$D_\mathrm{e} := \left\{ d \in D_0 \;\middle|\; \begin{array}{l} \exists x, y : \\ x_{k+1} = f(x_k, u_k, d), \\ y_k = h(x_k), \\ y_k \in Y_k, \forall k \in \{1, \ldots, N_e\} \end{array} \right\}. \tag{13.10}$$

Depicted in red in Fig. 13.1 (left) is the set of parameters D_e projected in the (d_1, d_2) space that generate trajectories satisfying $y_k \in Y_k$, $k = 1, \ldots, N_\mathrm{e}$ (Fig. 13.1 (right)).

Obtaining an exact characterization of the set D_e is not possible in general, and one has to resort to approximation techniques to make the problem computationally tractable. Here, a variant of the Set Inversion Via Interval Analysis (SIVIA) algorithm by (Jaulin and Walter, 1993) is used in order to approximate the solution set D_e as closely as possible. More concretely, the set D_e is approximated using the union of parameter sub-boxes that approximate its interior ($\mathbb{D}_\mathrm{int}$) and over-approximate its boundary ($\mathbb{D}_\mathrm{bnd}$). An illustration of such parameter sub-boxes is shown in Fig. 13.1 where $\mathbb{D}_\mathrm{out}$ stands for one partition of the parameter space which is guaranteed to have an empty intersection with

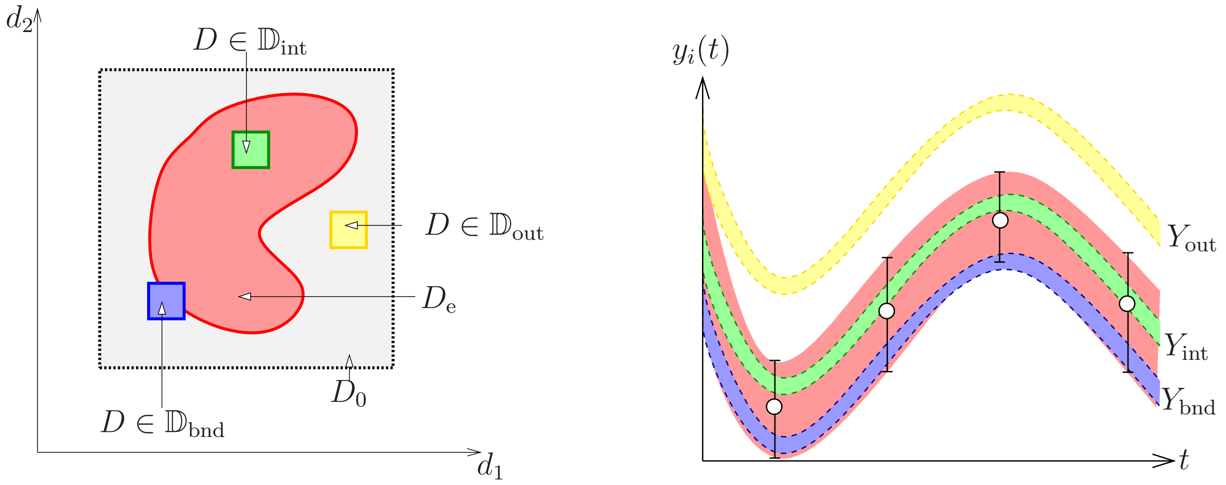
d_2
$D \in \mathbb{D}_{\text{int}}$
$D \in \mathbb{D}_{\text{out}}$
D_{e}
D_0
$D \in \mathbb{D}_{\text{bnd}}$
d_1
$y_i(t)$
Y_{out}
Y_{int}
Y_{bnd}
t

Figure 13.1: Illustration of the concept of guaranteed parameter estimation in the parameter space (left) and the corresponding output trajectories in the time domain (right).

$\mathbb{D}_{\text{e}}$. For each parameter box in Fig. 13.1 (left), bounding techniques as the ones described in Chapter 12 are used so that the parameter box can be classified as belonging to the interior (if the bounds are inside the measurement error for all steps), to the exterior (if the reachable sets are outside the measurement error for all steps) or to the boundary in an intermediate case. This is illustrated in Fig. 13.1 (right). Upon termination, this algorithm returns partitions $\mathbb{D}_{\text{int}}$ and $\mathbb{D}_{\text{bnd}}$ such that

$$\bigcup_{D \in \mathbb{P}_{\text{int}}} D \;\subseteq\; D_{\text{e}} \;\subseteq\; \left(\bigcup_{D \in \mathbb{D}_{\text{int}}} D \right) \cup \left(\bigcup_{D \in \mathbb{D}_{\text{bnd}}} D \right) =: D_{N_e}. \tag{13.11}$$

That is, the union of the interior boxes $\mathbb{D}_{\text{int}}$ and the boxes of the boundary $\mathbb{D}_{\text{bnd}}$ defines the over-approximation of the true set of parameters that explain the obtained process outputs given a bounded measurement noise and a model of the process. Further details on possible implementation variants of the described procedure can be found in (Paulen et al., 2013a,b).

13.3 Proposed Algorithm

The central idea of this chapter is the presentation of a novel algorithm for the reduction of the uncertainty of the model by applying robust optimal design of experiments and thus reducing the conservativeness introduced by a robust NMPC approach. The main motivation for proposing the novel OED criterion (13.7) is to estimate better those parameters that have a higher impact on the robust economic operation of the system. Since the Fisher information matrix provides only an approximation of the variance-covariance matrix of the estimated parameters, the use of guaranteed parameter estimation is proposed to ensure more accurate approximation of the possible parameter values taking into account the measurement noise.

We propose to divide the operation of the system into two different stages. In the first stage, an identification procedure is used, via the proposed robust OED criterion defined in (13.7). In order to be robust during this identification stage, a scenario tree is formulated using some a priori known bounds of the uncertainty (D_0). Then the multi-stage NMPC problem (13.6) is solved at each sampling time in which the cost of each scenario J_i is the OED criterion defined in (13.7). For this, it is necessary to previously solve the multi-stage NMPC problem (13.6) with an economic cost function in order to obtain the sensitivities of the optimal robust cost with respect to the range of the uncertainty $\left(\dfrac{\partial \tilde{J}^*}{\partial w(D)}\right)$. We consider a fixed length for the identification stage of N_e sampling times. Since this length is known a priori, the multi-stage NMPC problems are formulated in a shrinking horizon fashion, in which the prediction horizon at the start of the operation is $N_P = N_e$ and it is reduced at each sampling time.

Once the identification stage is finished, guaranteed parameter estimation is used to determine the new set of uncertain parameters D_{N_e}, which is used to build a new scenario tree using the combination of the maximum, minimum, and center values of the new parameter set. Then the second stage of the operation starts, in which multi-stage NMPC (13.6) is solved with the new scenario tree based on D_{N_e} and using an economic cost function as stage cost inside the cost of each scenario J_i until the end of the control problem (e.g. the end of the batch). The complete procedure is described in Algorithm 3.

The proposed algorithm is applied to a case study that is described in the next section. Its performance is compared to two different algorithms. First, the proposed algorithm is compared to the use of pure economic multi-stage NMPC (i.e., no OED), in which after N_e steps, parameter estimation is performed to generate a new tree, as in the proposed Algorithm 3. After that, multi-stage NMPC with the same economic cost function is applied (using the new scenario tree) until the end of the control problem. The proposed Algorithm 3 is also compared with a method in which instead of using the proposed scaling of the Fisher information matrix (13.7), the original modified E-design (13.5) is used as cost for each scenario. In order to have a fair comparison (as in the previous

Algorithm 3 Robust NMPC with uncertainty reduction

```
Input: $i_{\text{iter}} = 0; N_e > 0; x_0; D_0$

1. while $i_{\text{iter}} < N_e$ do
    1.1 Calculate $\frac{\partial \tilde{J}^*}{\partial w(D)}$ with $D = D_0$. For this it is necessary to solve (13.6) with an economic cost function as stage cost inside the cost of each scenario $J_i$.
    1.2 Solve the robust OED problem by solving (13.6) using as cost for each scenario the proposed OED criterion defined in (13.7) with a prediction horizon $N_e - i_{\text{iter}}$.
    1.3 Increment $i_{\text{iter}}$. Measure (or estimate) the new states $x_0$ at the next sampling time.
    end of while
2. Run guaranteed parameter estimation using the obtained measurements, getting $D_{N_e}$ as a result.
3. Generate a new scenario tree using the guaranteed maximum, middle and minimum values of the parameters from $D_{N_e}$.
4. Run multi-stage NMPC by solving (13.6) with an economic cost function inside the cost of each scenario until the end of the control problem (e.g. until the end of the batch).
```

case) the robust OED with modified E-design is performed for N_e steps, then guaranteed parameter estimation is used to generate a new tree and the rest of the control problem is done using multi-stage NMPC with an economic cost function. That is, the three controllers that will be compared only differ in the way that the identification phase (the first N_e steps) is performed. After the identification phase, the three controllers perform guaranteed parameter estimation and generate a new scenario tree which is used to apply multi-stage NMPC with an economic cost function until the end of the batch.

All the optimization problems reported in this Chapter are solved using IPOPT via CasADi as explained in Chapter 5. The sensitivities entering in (13.7) are calculated using sIPOPT (Pirnay et al., 2012) and the guaranteed parameter estimation is implemented using GOLIB (`http://www3.imperial.ac.uk/environmentenergyoptimisation/software`) and the library MC++ (`http://projects.coin-or.org/MCpp`).

13.4 Case study

In order to illustrate the advantages of the proposed approach, the following problem of the optimal control of a chemical reactor is considered. An exothermic chemical reaction $\mathrm{A} + \mathrm{B} \rightarrow \mathrm{C}$ is run in a fed-batch reactor equipped with a cooling jacket. This example has been adapted from (Srinivasan et al., 2003) and (Ubrich et al., 1999).

The reaction system is described by the following set of ODEs:

$$\frac{\mathrm{d}c_{\mathrm{A}}}{\mathrm{d}t} = -kc_{\mathrm{A}}c_{\mathrm{B}} - \frac{u}{V}c_{\mathrm{A}}, \qquad c_{\mathrm{A}}(0) = c_{\mathrm{A},0}, \tag{13.12a}$$

$$\frac{\mathrm{d}c_{\mathrm{B}}}{\mathrm{d}t} = -kc_{\mathrm{A}}c_{\mathrm{B}} + \frac{u}{V}(c_{\mathrm{B,in}} - c_{\mathrm{B}}), \qquad c_{\mathrm{B}}(0) = c_{\mathrm{B},0}, \tag{13.12b}$$

$$\frac{\mathrm{d}c_{\mathrm{C}}}{\mathrm{d}t} = kc_{\mathrm{A}}c_{\mathrm{B}} - \frac{u}{V}c_{\mathrm{C}}, \qquad c_{\mathrm{C}}(0) = 0, \tag{13.12c}$$

$$\frac{\mathrm{d}V}{\mathrm{d}t} = u, \qquad V(0) = V_0, \tag{13.12d}$$

where c_i represents concentration of the substance i, k stands for the reaction rate, V is the volume of the reactor, and u represents the feed flowrate of reactant B with concentration $c_{\mathrm{B,in}}$.

The reaction is run under isothermal conditions where the inlet cooling jacket temperature is assumed to be adjusted to maintain the temperature in the reactor at $T = 70°\mathrm{C}$. The evolution of the temperature of the cooling medium inside the jacket obeys:

$$T_j(t) = T - \frac{(-\Delta H) k c_{\mathrm{A}}(t) c_{\mathrm{B}}(t) V(t)}{\alpha A(t)}, \tag{13.13}$$

where ΔH is the reaction enthalpy, α is a heat transfer coefficient and A is the contact area between the jacket and the reactor content.

In order to prevent an uncontrollable behavior of the reaction under a cooling failure, the maximum attainable temperature is restricted to:

$$T_{cf} = T(t) + \min_{i \in \{A,B\}} c_i \frac{(-\Delta H)}{\rho c_p} \leq T_{\max}, \tag{13.14}$$

where ρ denotes the density and c_p the heat capacity of the reaction mixture. Additionally, the volume of the reactor is bounded by its maximum value, $V \leq V_{\max}$ and the control input is bounded ($u_{\min} \leq u \leq u_{\max}$) as well.

The control task is to achieve a desired mass of the product C as fast as possible, $n_{\mathrm{C}} = c_{\mathrm{C}} V \geq n_{\mathrm{C,des}}$. The minimum time problem is approximated by the maximization of the mass of product C (n_{C}) over a finite prediction horizon, since simulation studies showed that the results obtained are almost equivalent. The initial conditions, values of the parameters and constraints are given in Table 13.1.

It is considered that the parameters k and ΔH are uncertain and have constant but unknown values in the range ±30% with respect to their nominal values. The measured quantities (c_{A}, c_{B} and T_j) are subject to bounded noise $E_{c_i} = [-0.05, 0.05]$ and $E_{T_j} = [-0.5, 0.5]$ which is simulated using rounding of the values of the simulated outputs for the purpose of reproducibility of the results obtained here.

The results of applying Algorithm 3 to the presented case study are shown in the remainder of this chapter. The first step is the robust optimal dynamic experiment design. In order to achieve robust performance and satisfaction of the constraints for all the possible values of the uncertainty (±30%), the OED problem is formulated using a scenario tree that contains the maximum, minimum and nominal value of the uncertain parameters with a robust horizon equal to 1. The OED problem is solved in a shrinking horizon fashion for $N_e = 10$ steps and a sampling time $t_{\mathrm{step}} = 0.1$ h. The results for three algorithms described before are shown in Fig. 13.2. The purely economic design (no OED) uses as

Table 13.1: Parameter values, initial conditions and bounds.

Parameter	**Value**	**Units**
k	0.0482	$\mathrm{L\,mol^{-1}\,h^{-1}}$
ΔH	-60000	$\mathrm{J\,mol^{-1}}$
T	70	°C
ρ	900	$\mathrm{g\,L^{-1}}$
c_p	4.2	$\mathrm{J\,g^{-1}\,K^{-1}}$
$c_{\mathrm{B,in}}$	10	$\mathrm{mol\,L^{-1}}$
$c_{\mathrm{A,0}}$	2	$\mathrm{mol\,L^{-1}}$
$c_{\mathrm{B,0}}$	0.46	$\mathrm{mol\,L^{-1}}$
$c_{\mathrm{B,0}}$	0	$\mathrm{mol\,L^{-1}}$
V_0	0.7	L
u_{min}	0	$\mathrm{L\,h^{-1}}$
u_{max}	0.1	$\mathrm{L\,h^{-1}}$
T_{max}	80	°C
V_{max}	1	L
$n_{\mathrm{C,des}}$	0.6	mol

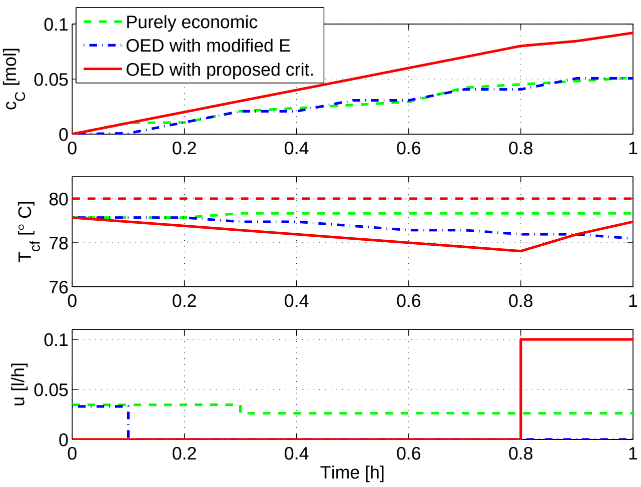

Figure 13.2: Concentration c_{C}, temperature T_{cf} and control input u obtained from running economic multi-stage NMPC and robust OED for different design criteria.

objective function the maximization of the product C (n_{C}), the OED with modified E-design uses as cost for each scenario the equation (13.5) and the OED with the proposed criterion minimizes (13.7) for each scenario.

After $N_e = 10$ steps, guaranteed parameter estimation is run. The obtained sets of guaranteed parameter estimates are shown in Fig. 13.3. The projections of the resulting sets on the parameter axes determine the generation of the new scenario tree for the solution of the multi-stage NMPC problem. As expected, the use of a purely economic cost yields the biggest ranges of the parameters which justifies the utilization of OED information in this case study. The use of OED with modified E-design gives smaller parameter ranges. The proposed algorithm with the OED criterion (13.7) yields a smaller range for ΔH and a larger range for the parameter k in comparison to the ranges obtained for the modified E-design. All the sets contain the real values of the parameters that in this case were assumed to be 20% larger (in absolute value) than the nominal values.

Due to the novel scaling of the Fisher matrix introduced in (13.7), ΔH is estimated with a higher accuracy when the proposed algorithm is used and this results in a superior performance compared to the other algorithms (see Fig. 13.4) when multi-stage NMPC is run with the new scenario tree based on the parameter estimates and with an economic cost function (maximization of n_{C}). The reason for this is that ΔH has a higher impact on the robust economic operation of the plant, since it influences directly the

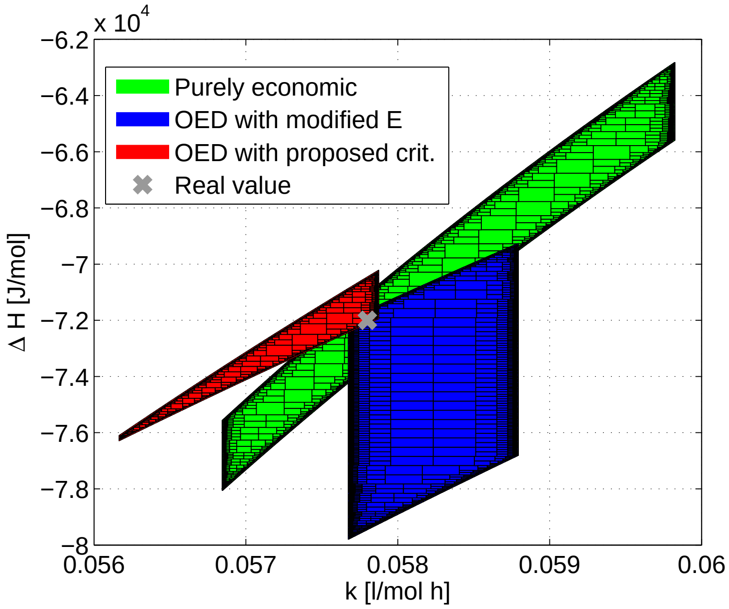

Figure 13.3: Sets of guaranteed parameter estimates resulting from the measurements and control inputs obtained from running economic multi-stage NMPC and robust OED for different design criteria.

constraint on the temperature T_{cf}. This information is given by the sensitivity of the robust economic cost with respect to the range of the uncertainty, which is used to scale the Fisher information matrix. For all the algorithms the same economic cost function is used (maximization of product C (n_{C})), with a scenario tree given by the guaranteed parameter estimates using a prediction horizon $N_P = 10$ and a robust horizon $N_R = 1$.

As it is shown in Fig. 13.4, the economic optimal operation of the plant consists in driving the system as close as possible to the temperature constraint T_{cf}. Multi-stage NMPC calculates automatically a back-off from the constraint to ensure that the constraint is not violated for any value of the uncertainty. If the range of the uncertainty that has to be taken into account is wider, then the necessary back-off is larger and the resulting economic performance decreases. In this case, the lower bound on the uncertain parameter ΔH is the most important factor for the back-off. Therefore the dual control-like procedure that uses the OED with modified E design criterion gives a performance similar to the performance achieved by pure robust economic cost optimization (without OED) despite yielding a narrower range of parameter uncertainty. The algorithm proposed in this chapter achieves a batch time reduction by 1.5 hours which stands for a 7.5 % im-

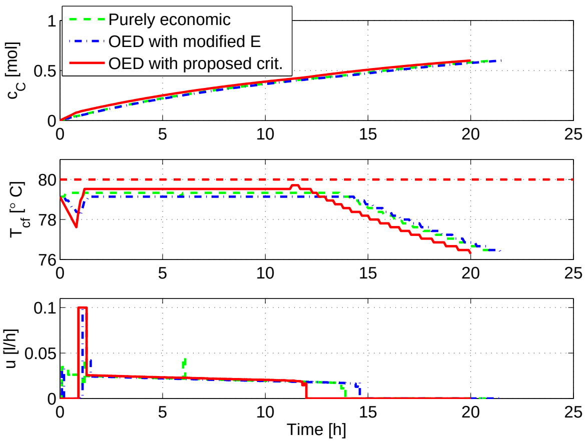

Figure 13.4: Concentration c_C, temperature T_{cf} and control input u obtained from running multi-stage NMPC with a scenario tree generated with parameter bounds obtained by the guaranteed parameter estimation for the different algorithms. During the first hour the inputs used are the ones obtained during the identification phase.

provement over running robust NMPC with economical cost and the same procedure for estimation of the uncertainty.

13.5 Using the Correct Sensitivity Information

Recently, the work presented in (Recker et al., 2013), proposed the use of sensitivity information of the economic cost with respect to the uncertain parameters to scale the Fisher information matrix, in a similar manner as proposed in this chapter. This section studies the difference between both approaches and shows an example that illustrates that relying on the sensitivity of the cost w.r.t the parameter as proposed in (Recker et al., 2013) can lead to wrong conclusions.

To illustrate this concept, both sensitivities are shown in Fig. 13.5 for the case study presented above when applying multi-stage NMPC with an economic cost function assuming an uncertainty range of ±30% with respect to their nominal values. If the sensitivity of the optimal robust cost with respect to the parameters is used, as proposed in (Recker et al., 2013), the sensitivity of the optimal cost with respect to the reaction rate k is 5 times higher than the sensitivity with respect to the reaction enthalpy ΔH as shown in Fig. 13.5 (left). The same happens if standard NMPC is used instead of multi-stage

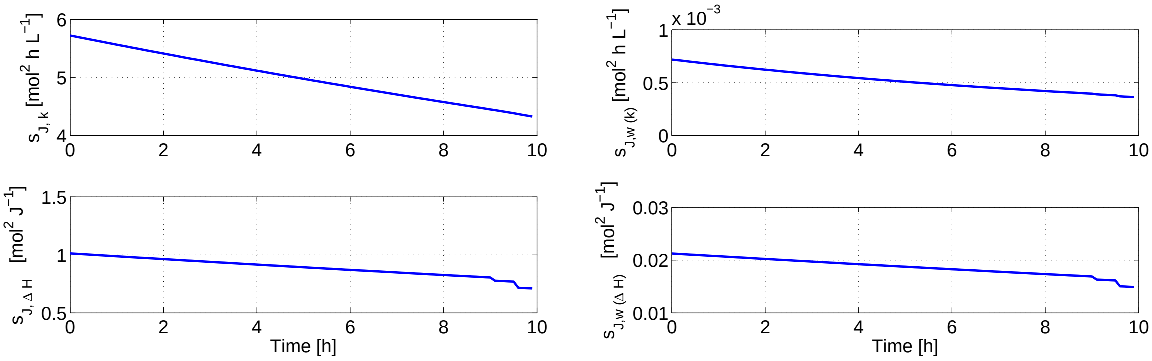

Figure 13.5: Sensitivity of the optimal robust cost with respect to the uncertain parameters k and ΔH (left) and sensitivity of the optimal robust cost with respect to the range of uncertain parameters $w(k)$ and $w(\Delta H)$ (right).

NMPC. This occurs because the economic cost function is to maximize n_{C}, and if the reaction occurs faster (higher k) then more n_{C} will be produced. This sensitivity analysis would lead to the conclusion that it is better to estimate k with a higher accuracy in order to maximize the cost.

On the other hand, if the sensitivities of the parameters with respect to the uncertainty range are calculated, as proposed in this thesis, the sensitivity of the uncertainty range in ΔH is 30 times larger than the sensitivity of the uncertainty range in k as shown in Fig. 13.5 (right), clearly contradicting the conclusions from the parametric sensitivities.

Three different simulation studies are conducted in order to illustrate the effects of the different conclusions. Firstly, one batch is simulated assuming that there is no estimation of any of the parameters and that the uncertainty range remains $\pm 30\%$ with respect to the nominal values of both parameters. Secondly, following the conclusions extracted from the sensitivities of the optimal robust cost with respect to the parameters it is assumed that the parameter k can be perfectly estimated (no uncertainty) and that ΔH remains uncertain in a range $\pm 30\%$ with respect to its nominal value because the sensitivity analysis suggests that estimating k leads to a better performance than estimating ΔH. Thirdly, following the information of the sensitivity of the cost with respect to the parameter ranges, a batch is run assuming that ΔH can be perfectly estimated and that k remains uncertain.

Relying on the information of the sensitivity of the optimal cost with respect to the parameters (assuming that k can be perfectly and instantaneously estimated) leads to a minor improvement (0.04% shorter batch time) with respect to the case where no estimation is performed at all as shown in Fig. 13.6. The reason is that using the sensitivity of the robust optimal cost with respect to the parameter is misleading. That is, the fact that one parameter has a higher influence on the robust optimal cost does not mean that estimating it with a better accuracy would lead to an increased performance. The information which should be used to decide which parameter to estimate better is a measure of how much can the robust economic performance be improved if a better estimation is achieved. The proposed sensitivity of the robust optimal cost with respect to the range of the uncertainty provides such information. Following the conclusions that the proposed sensitivities provide (assuming here as a consequence that ΔH can be perfectly and instantaneously estimated) leads to a smaller back-off, which makes possible to increase the performance as it can be seen in Fig. 13.6. The obtained batch time is 14% shorter compared to the ones obtained when none of the parameters is estimated or when k is perfectly estimated. The main reason for this improvement is that the performance is strongly influenced by how close to the constraints the process can be operated, and therefore the best way to operate the uncertain system is to estimate better the parameter that makes it possible to reduce this back-off. This information is included in the proposed sensitivities with respect to the range of the uncertainty, but not in the sensitivity of the optimal robust cost with respect to the uncertain parameters. In this case the sensitivity with respect to the range of the uncertainty is used as a measure of the knowledge of the uncertainty, but it is also possible to use other measures such as the area of the confidence ellipsoids, or the upper and lower bounds of each parameter.

If standard NMPC is used, calculating the sensitivities of the optimal robust cost with respect to the ranges of the uncertainty is not trivial. However, the use of multi-stage NMPC on scenario trees makes this process straightforward since they can be obtained with very cheap computations when solving the multi-stage problem, using e.g. sIPOPT.

This chapter shows how the performance of the multi-stage NMPC approach can be enhanced by reducing the uncertainty range. A new criterion for robust OED is proposed in order to prioritize the more accurate estimation of the parameters that have a higher influence on the robust economic operation of the system. In order to avoid the unreliable approximation of the parameter ranges associated with typical OED approaches, a guaranteed parameter estimation approach is used to obtain the bounds of the uncertain parameters. The obtained bounds are used to build a new scenario tree with reduced

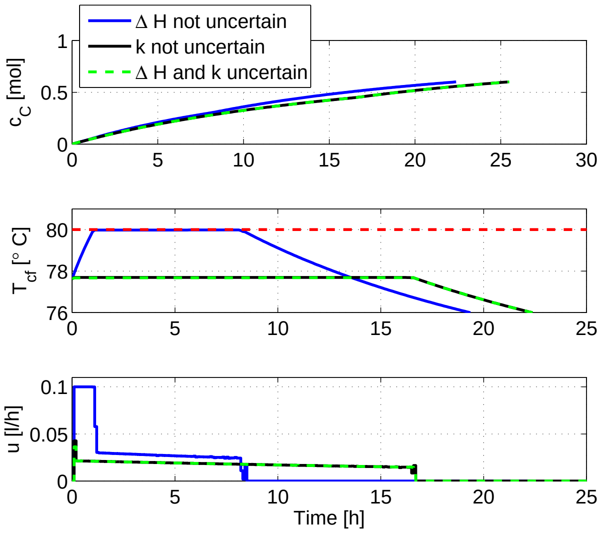

Figure 13.6: Concentration c_C, temperature T_{cf} and control input u obtained from running multi-stage NMPC with a scenario tree generated for different knowledge of the uncertain parameters.

uncertainty and a better economic performance is achieved. Simulation results of a chemical reactor example show the potential of the approach and the possible improvements compared to a typical OED design and to a standard robust economic operation of the plant.

Part V

Summary, Conclusions, and Future Work

Chapter 14

Conclusions and Future Work

14.1 Conclusions and Guidelines for the Use of Multi-stage NMPC

The main goal of this thesis was to develop a new approach for the explicit consideration of uncertainty in the field of optimization-based control of nonlinear systems which is not overly conservative and is real-time implementable. This problem has been addressed in the framework of nonlinear model predictive control using a novel formulation called multi-stage NMPC. The theoretical basis, the efficient implementation and different simulation and experimental results have been presented in the four parts of this thesis.

The first part of the thesis presented multi-stage NMPC as a general framework that includes other approaches for robust NMPC, such as min-max NMPC, or open-loop robust NMPC. Here, the main advantage of the multi-stage approach was discussed: the tree structure makes it possible to consider explicitly that new information will become available in the future. This implies that the control inputs can act as recourse variables, reducing the conservativeness of the approach significantly. This comes at the price of an exponential growth of the scenario tree with the length of the prediction horizon and with the number of uncertainties taken into account. Furthermore, a formulation of multi-stage NMPC has been described which guarantees a priori stability and recursive feasibility.

The second part presented the efficient implementation of the approach that has been developed in this thesis which is necessary to achieve a real-time solution of the large optimization problems that result from the multi-stage NMPC formulation. In particular, the use of orthogonal collocation on finite elements with exact first and second order derivative information calculated using automatic differentiation has been proven to achieve excellent computational performance. This has been implemented using the tools CasADi and IPOPT, which provide a very high performance with a low implementation effort. Additionally, a new tool called DO-MPC has been developed, by which MPC can be implemented in a modular manner, using the four main parts of any model-based optimizing control implementation: model, optimizer, observer and simulator. DO-MPC provides an environment for the easy, simple and efficient use of multi-stage NMPC, including the communication and control with real plants.

In the third part of this thesis it was shown that multi-stage NMPC with an efficient implementation achieves very promising results for challenging problems, satisfying the constraints for all the assumed possible values of the uncertainty. In particular the approach has been extensively tested for the control of highly nonlinear polymerization processes. It has been shown that multi-stage NMPC achieves superior performance compared to standard NMPC or to other robust approaches. Furthermore, multi-stage NMPC has been tested successfully on a laboratory plant using DO-MPC, which illustrates that the necessary computations can be solved in real-time and thus can be applied to real systems.

The fourth part of the thesis shows how the flexibility of the multi-stage NMPC approach enables the extension of the method by combining it with other ideas to further enhance its capabilities and performance. For example, it is possible to use multi-stage NMPC to achieve a trade-off between variability of the system under uncertainty and its economic performance. Furthermore, the approach can be extended by integrating it with reachability analysis tools for the rigorous guarantee of constraint satisfaction and with robust optimal experiment design to reduce the uncertainty and improve performance.

Overall it can be concluded that multi-stage NMPC provides excellent results for medium size problems with tight constraints on inputs and on states. It has been shown that this can be especially beneficial if an economic cost function is chosen, and that the resulting optimization problems can be solved in real-time using the efficient implementation proposed in this thesis.

The main limitations of the approach can be summarized in the following two points.

Firstly, the consideration of high dimensions of the uncertainty space poses an important challenge because the size of the scenario tree grows exponentially with the number of uncertainties considered. Secondly, although it has been shown that it is possible to provide rigorous guarantees for stability and constraint satisfaction, the necessary assumptions and tools needed for these guarantees limit the applicability of the approach, as it occurs with any MPC approach.

14.2 When and How to Use Multi-stage NMPC?

This section presents some guidelines that provide insight on the kind of problems that can be tackled by multi-stage NMPC and on the problems where its use can be more advantageous with respect to other approaches.

Once it has been decided that multi-stage NMPC is a suitable approach for the problem under consideration the controller has to be designed. This section also presents some guidelines for the design of the multi-stage NMPC controller and the use of the different extensions and modifications that have been presented in this thesis. Parts of this section have been published in (Lucia and Engell, 2015).

14.2.1 When to Use Multi-stage NMPC

Multi-stage NMPC is a robust NMPC approach and as such, it is in general suboptimal when compared to a standard NMPC that uses the perfect model of the system, because the multi-stage approach accounts for possible uncertainties. Therefore, the best option to achieve a robust NMPC scheme is to estimate the uncertainties if it is possible, removing the uncertainty of the problem. However, if the uncertainties vary over time, the application of standard NMPC (even in the case of exact and instantaneous estimation) may result in constraint violations. On the other hand, the use of multi-stage NMPC can prevent the constraint violations because it takes into account in advance that the uncertainty might change as illustrated in Chapter 8.

For the cases when the uncertainty cannot be exactly estimated, the use of multi-stage NMPC is beneficial in comparison with standard NMPC or with other robust approaches that do not take feedback into account. The benefits can be seen in the form of increased average performance over the different possible scenarios of the uncertainty or in the form of robust constraint satisfaction. Furthermore, the flexibility of the approach makes it possible to integrate it with estimation techniques (see Chapter 8) or with optimal experiment design (see Chapter 13) to enhance the performance based on measurement information.

For the case of an economic cost function and tight constraints, an important improvement in the performance is expected by using multi-stage NMPC as it has been shown in several results (see Chapter 8). If the control task is the unconstrained tracking of a pre-defined setpoint the benefits of multi-stage NMPC can be very small or nonexistent (depending on the problem) compared to the use of standard NMPC with a bias term to achieve steady-state accuracy as illustrated in Appendix A. In that case, it is necessary to analyze for each problem if the possible improvements justify the increase in the complexity of the controller.

Stochastic information about the uncertainty can be incorporated in the multi-stage formulation using the weights for each scenario. However, it is not possible to handle directly stochastic formulations of the constraints (chance constraints) with the tools presented in this thesis. If such constraints are necessary, other approaches can be considered such as the scenario approach in the convex case (Calafiore and Campi, 2006) or the use of polynomial chaos expansions (Mesbah et al., 2014).

For simple cases where it is possible to find the invariant sets that are necessary for the design of tube-based controllers, they can be preferred to the multi-stage NMPC controller because they can be implemented with the same computational complexity as standard NMPC (see (Yu et al., 2011)) and they provide set-theoretic guarantees about the possible trajectories of the controlled system. However, for general nonlinear systems it is very difficult to find the necessary elements for the design of tube-based NMPC. Furthermore, these controllers ignore the question of optimal performance under uncertainty.

For a real-world example in which the most important uncertainties can be summarized in only a few parameters (or disturbances) with known bounds, multi-stage NMPC represents a very promising strategy that provides excellent performance satisfying tight constraints if enough computation power is available. Industrially relevant case studies can be solved in real-time if an efficient implementation of multi-stage NMPC is used, as described in Chapter 5.

14.2.2 How to Use Multi-stage NMPC

Once multi-stage NMPC has been chosen as the control approach for a given system, the following steps must be taken to design the controller.

The first step consists in designing the scenario tree. An easy rule to generate a suitable scenario tree is to consider the combination of the maximum, minimum, and optionally also the nominal values of the different uncertain parameters as scenarios. This is the main limiting factor of the approach. If there are many uncertain parameters the resulting scenario tree might be intractable, and it is therefore necessary to lump the effect of several uncertainties into a few critical uncertainties.

Then the robust horizon has to be chosen. In this thesis it has been shown that branching the tree only in the first stage (robust horizon $N_R = 1$) results in very good results with low computational effort. Nevertheless, it has to be kept in mind that this assumes that at the next sampling time different control inputs can be taken depending on the uncertainty. This is not true because at the next sampling time a new scenario tree (shifted in time) will be solved which imposes that all the control inputs have to be the same in the first stage. This can potentially lead to recursive infeasibility of the controller, although it has not been encountered during the case studies solved in this thesis.

Then it is possible to make use of the different enhancements and extensions that have been provided in this thesis to improve the performance of multi-stage NMPC.

If some information about the uncertainty is available, it can be introduced on-line into the scenario tree. This can be done either by adjusting the probabilities of the different scenarios of the tree (see Chapter 8), or by *narrowing* the tree based on the new bounds of the uncertainty that are provided by confidence ellipsoids or other estimation techniques (see Chapter 13).

If theoretical guarantees are required it is possible to have an a priori guarantee of stability and recursive feasibility under the usual assumptions (terminal set and terminal penalty term) as shown in Chapter 4 for the case of a tracking cost function. If only a guarantee of robust constraint satisfaction is needed, the reachable sets of each scenario can be computed as shown in Chapter 12.

The multi-stage formulation can also be used to achieve a trade-off between performance and variability of the controlled system under uncertainty by penalizing the distance between the different scenarios of the tree in the state space. However, enforcing a low variability might result in a significant loss of performance as shown in Chapter 11.

14.3 Future Work

In the following, several ideas for future work are described which can enhance the capabilities of the multi-stage approach.

Throughout this thesis, it has been usually assumed for the design of the controller that the states of the system can be exactly measured, which is in general not true. Although it has been shown in simulations and in real experiments that in some cases ignoring measurement and estimation errors can lead to good results, it is important to study more rigorously the interplay between the estimation strategy, which is necessary for any MPC implementation in which some of the states cannot be measured, and the controller itself. The propagation of the uncertainty introduced by measurement and estimation errors into the future poses an extraordinary challenge due to the exponential growth of the scenario tree. First steps in this direction have been published in (Subramanian et al., 2014).

The presence of many uncertainties is the most important challenge of the proposed approach. In this case the enumeration of all the possible combinations of the extreme values might lead to an intractable optimization problem and the multi-stage NMPC approach should be combined with other methods. One possibility is to use reachability analysis combined with the scenario tree, so that only a few critical uncertainties are taken into account in the tree and the rest are considered by the calculation of the reachable sets.

The flexibility of the approach facilitates its integration with other methods. Especially, multi-stage NMPC is suitable for the formulation of dual-control approaches, because the scenario tree enables the explicit consideration of the future reduction of the uncertainty that could be achieved thanks to future measurements by considering different values of the uncertainty at each time stage.

The stability guaranteeing formulation of multi-stage NMPC that has been presented in this thesis requires typical assumptions for the stability of NMPC which are necessary to prove that the optimal value of the cost function is a Lyapunov function. For the general case of economic MPC, this is not always the case. Recent work on stability of economic MPC (Angeli et al., 2012) can be used to extend the stability proof for the case of an economic cost function.

Finally, this thesis presented simulation results for an industrial batch polymerization reactor provided by BASF SE. The application of the approach to a real industrial reactor is an important step for the validation of multi-stage NMPC as a powerful strategy to deal with uncertainty in the framework of nonlinear model predictive control in industrial practice.

Appendix A

Multi-stage NMPC for Setpoint Tracking

Most of the results presented in this thesis were obtained with the use of an economic cost function. This problem is particularly interesting because it leads very often to an operation of the process close to the constraints, leading to constraint violations in the case of plant-model mismatch or disturbances. However, the classical goal of tracking a predefined setpoint under uncertainty is also an important control task. This chapter shows how multi-stage NMPC can be used for tracking setpoints using the same strategies employed for standard NMPC. An important part of this appendix has been published in (Lucia and Engell, 2015).

A.1 Case Study

The results of this appendix are illustrated using a nonlinear CSTR benchmark problem adapted from (Klatt and Engell, 1998). The dynamics of the CSTR are described by the following differential equations:

$$\dot{c}_{\mathrm{A}} = F(c_{\mathrm{A0}} - c_{\mathrm{A}}) - k_1 c_{\mathrm{A}} - k_3 c_{\mathrm{A}}^2, \tag{A.1a}$$

$$\dot{c}_{\mathrm{B}} = -F c_{\mathrm{B}} + k_1 c_{\mathrm{A}} - k_2 c_{\mathrm{B}}, \tag{A.1b}$$

$$\dot{T}_{\mathrm{R}} = F(T_{\mathrm{in}} - T_{\mathrm{R}}) + \frac{k_{\mathrm{W}} A}{\rho c_{\mathrm{p}} V_{\mathrm{R}}}(T_{\mathrm{K}} - T_{\mathrm{R}}) - \frac{k_1 c_{\mathrm{A}} \Delta H_{\mathrm{AB}} + k_2 c_{\mathrm{B}} \Delta H_{\mathrm{BC}} + k_3 c_{\mathrm{A}}^2 \Delta H_{\mathrm{AD}}}{\rho c_{\mathrm{p}}}, \tag{A.1c}$$

$$\dot{T}_{\mathrm{K}} = \frac{1}{m_{\mathrm{K}} c_{\mathrm{pK}}}(\dot{Q}_{\mathrm{K}} + k_{\mathrm{W}} A (T_{\mathrm{R}} - T_{\mathrm{K}})), \tag{A.1d}$$

where the reaction rates k_i follow the Arrhenius law:

$$k_i = k_{0,i} e^{\frac{-E_{A,i}}{R(T_{\mathrm{R}}+273.15)}}. \tag{A.2}$$

The ODEs are derived from component balances for the concentration of component A (c_{A}) and for the the concentration of component B (c_{B}). The energy balances for the temperature of the reactor T_{R} and for the coolant temperature (T_{K}) form the last two differential equations. The control inputs are the inflow (F) normalized by the volume of the reactor and the heat removed by the coolant ($\dot{Q}_K$). The parameters that appear in the model equations are described in Table A.1.

Table A.1: Parameter values of the CSTR.

Parameter	**Value**	**Unit**
$k_{0,1}$	1.287 10^{12}	h^{-1}
$k_{0,2}$	1.287 10^{12}	h^{-1}
$k_{0,3}$	9.043 10^{9}	$\mathrm{l\,mol^{-1}h^{-1}}$
$E_{A,1}/R$	9758.3	K
$E_{A,2}/R$	9758.3	K
$E_{A,3}/R$	8560.0	K
ΔH_{AB}	4.2	KJ mol^{-1}
ΔH_{BC}	-11.0	KJ mol^{-1}
ΔH_{AD}	-41.85	KJ mol^{-1}
ρ	0.9342	$\mathrm{kg\,l^{-1}}$
c_{p}	3.01	$\mathrm{kJ\,kg^{-1}\,K^{-1}}$
c_{pK}	2.0	$\mathrm{kJ\,kg^{-1}\,K^{-1}}$
A	0.215	m^2
V_{R}	10.01	l
m_{k}	5.0	kg
T_{in}	130.0	°C
k_{W}	4032	KJ h^{-1} m^{-2} K^{-1}

The initial conditions of the states together with the constraints on the states are described in Table A.2. The constraints for the control inputs are shown in Table A.3.

Table A.2: Initial conditions and state constraints.

State	Init. cond.	Min.	Max.	Unit
c_{A}	0.8	0.1	5.0	$\mathrm{mol\ l^{-1}}$
c_{B}	0.5	0.1	5.0	$\mathrm{mol\ l^{-1}}$
T_{R}	134.14	50.0	180.0	°C
T_{J}	134.0	50.0	180.0	°C

Table A.3: Bounds on the manipulated variables.

Control	Min.	Max.	Unit
F	5	100	$\mathrm{h^{-1}}$
$\dot{Q}_{\mathrm{K}}$	-8500	0	$\mathrm{kJ\,h^{-1}}$

A.2 Setpoint Tracking under uncertainty

The control task of the presented case study is to track a predefined setpoint for the concentration of component B (c_{B}). It is considered that the activation energy $E_{A,3}$ is uncertain and it is assumed that it varies by ±10% with respect to the nominal value described in Table A.1.

The stage cost that is minimized at each time stage and for each scenario is chosen as:

$$L = (c_{\mathrm{B}} - c_{\mathrm{B}}^{\mathrm{ref}})^2 + r_1 \Delta F^2 + r_2 \Delta \dot{Q}_{\mathrm{K}}^2, \tag{A.3}$$

where the penalty terms for the control movements are chosen as $r_1 = 10^{-5}$ and $r_2 = 10^{-7}$. The setpoint is chosen to be $c_{\mathrm{B}}^{\mathrm{ref}} = 0.5$ for $t \leq 0.3$ h and $c_{\mathrm{B}}^{\mathrm{ref}} = 0.7$ for $t > 0.3$ h. The prediction horizon is $N_P = 40$ steps and the sampling time of the controller is $t_{\mathrm{step}} = 0.005$ h. For the multi-stage case, a scenario tree is generated using the maximum, minimum and nominal value of the uncertainty and a robust horizon $N_R = 1$.

The results of the tracking problem for standard and for multi-stage NMPC are shown in Fig. A.1. The state constraints are chosen such that they remain inactive to analyze only the tracking performance. Standard NMPC results in a steady state error (Fig. A.1 (left)) for all the cases of the uncertainty except when a perfect model is used (0% variation with respect to the nominal value of the parameter). If multi-stage NMPC is used, the steady state error cannot be completely avoided, but the reason for this offset is different than in

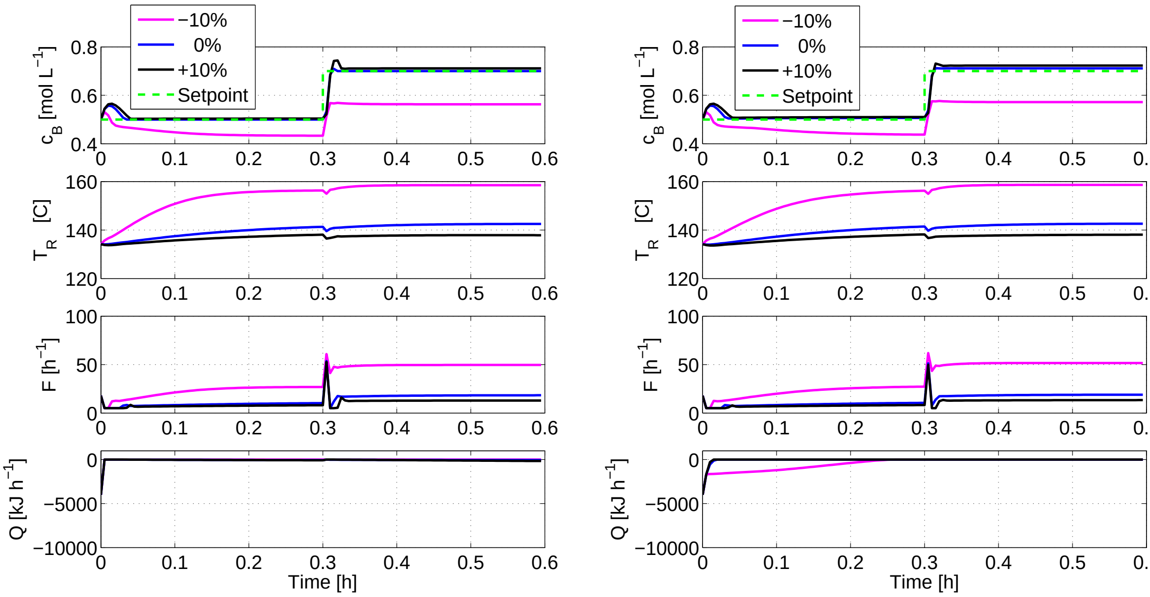

Figure A.1: Concentration of component B, reactor temperature, and control inputs obtained by standard NMPC (left) and by multi-stage NMPC (right) tracking a predefined setpoint for different values of the uncertain parameter (±10% w.r.t. the nominal value).

the standard NMPC case. In standard NMPC the steady-state error occurs because the controller minimizes the tracking cost function using a wrong model for the predictions. According to this model the calculated input would drive the system to the setpoint but once the control is applied to the system, it remains in the same position. For multi-stage NMPC the controller calculates sequences of control inputs (one for each scenario) that minimizes the distance to the setpoint on the average. Since the first control input is common due to the non-anticipativity constraints, it is not possible to drive the system to the setpoint for all the scenarios in the first stage and this results in the steady-state error that can be seen in Fig. A.1 (right). Table A.4 shows that multi-stage NMPC achieves a better performance in average (~11%) comparing the average accumulated cost over the three scenarios. The accumulated cost is calculated by integrating the tracking error over the whole time period. Standard NMPC has better performance when the model is perfect, but it has no control about the loss of performance for the rest of the cases. Multi-stage NMPC calculates the inputs that result on minimum average performance. If the performance of standard NMPC is very similar to the average performance (e.g. in the linear case), standard NMPC and multi-stage will provide almost identical solutions.

Table A.4: Performance comparison between standard NMPC and multi-stage NMPC.

	Accumulated cost	
Unc. in $E_{\mathbf{A},3}$	**Standard NMPC**	**Multi-stage NMPC**
+10%	0.0602	0.0827
+0%	0.0389	0.0467
-10%	1.3041	1.1272
Average	1.403	1.256

A common strategy to achieve steady-state accuracy is to add a bias term to the cost function so that the output of the system is corrected based on the difference between the predicted output and the measured one. The reference (or equivalently the output) is updated as:

$$\tilde{c}_{\mathrm{B}}^{\mathrm{ref}} = c_{\mathrm{B}}^{\mathrm{ref}} + (c_{\mathrm{B}}^{j} - c_{\mathrm{B}}^{\mathrm{meas}}), \tag{A.4}$$

where $\tilde{c}_{\mathrm{B}}^{\mathrm{ref}}$ is the actual reference used in the cost function, $c_{\mathrm{B}}^{\mathrm{meas}}$ is the measured concentration and c_{B}^{j} is the predicted concentration. The bias is assumed to be constant over the prediction horizon. Such a simple correction improves the performance but fails to achieve steady-state accuracy as can be seen in Fig. A.2. The reason for this is that due to the high nonlinearity of the problem (the uncertain parameter enters in the exponential term in (A.2)), the inputs that the NMPC calculates for an adapted reference with the wrong model are very different compared to the inputs that are necessary to drive the real system to the original reference, resulting in an offset. The same occurs for the multi-stage case. Augmenting the system with a disturbance model and an observer to estimate its state makes it possible to achieve for offset-free tracking as shown e.g. in (Morari and Maeder, 2012; Huang et al., 2012; Rajamani et al., 2009).

Another very simple strategy to achieve steady state accuracy is to adapt the reference based on the integrated tracking error. This continuous update is given by:

$$\tilde{c}_{\mathrm{B}}^{\mathrm{ref}} \leftarrow \tilde{c}_{\mathrm{B}}^{\mathrm{ref}} + k_{\mathrm{bias}}(c_{\mathrm{B}}^{\mathrm{ref}} - c_{\mathrm{B}}^{\mathrm{meas}}). \tag{A.5}$$

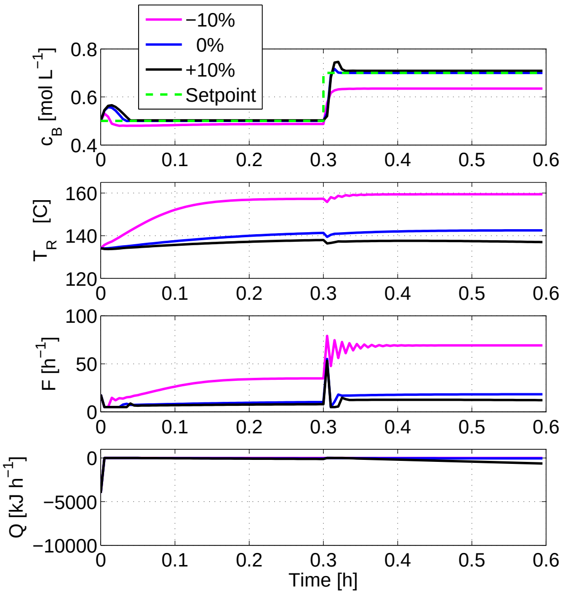

Figure A.2: Concentration of component B, reactor temperature (with constraint), and control inputs obtained by standard NMPC with bias term tracking a pre-defined setpoint for different values of the uncertain parameter (±10% w.r.t. the nominal value).

Using $k_{\text{bias}} = 0.2$ steady-state accuracy is achieved for all scenarios for both standard NMPC (Fig. A.3 (left)) and for multi-stage NMPC (Fig. A.3 (right)). There is no significant difference between the performance of standard and multi-stage NMPC. The reason for this is that the bias term adapts the output of the model based on the measurement information. In particular when the plant reaches the steady state the input-output behavior is perfectly corrected for all the cases of the uncertainty. During the dynamic part of the trajectory the bias update is an approximation of the exact correction, but this does not have to be worse than optimizing an average performance for several scenarios – where only one scenario is the real one – as it is done in multi-stage NMPC.

It is possible to improve the performance of the controllers using a combination of both presented bias updates by using a varying value of k_{bias}. This can be achieved by scaling the bias term with the accuracy of the prediction of the model in the last step when compared to the obtained measurement. The idea of this scaling is that if the effect of the uncertainty on the output is large, then the corrections should also be large. On the other hand, if the prediction given by the model coincides with the measurement or if the plant is already at the desired setpoint, no correction should be performed. The bias

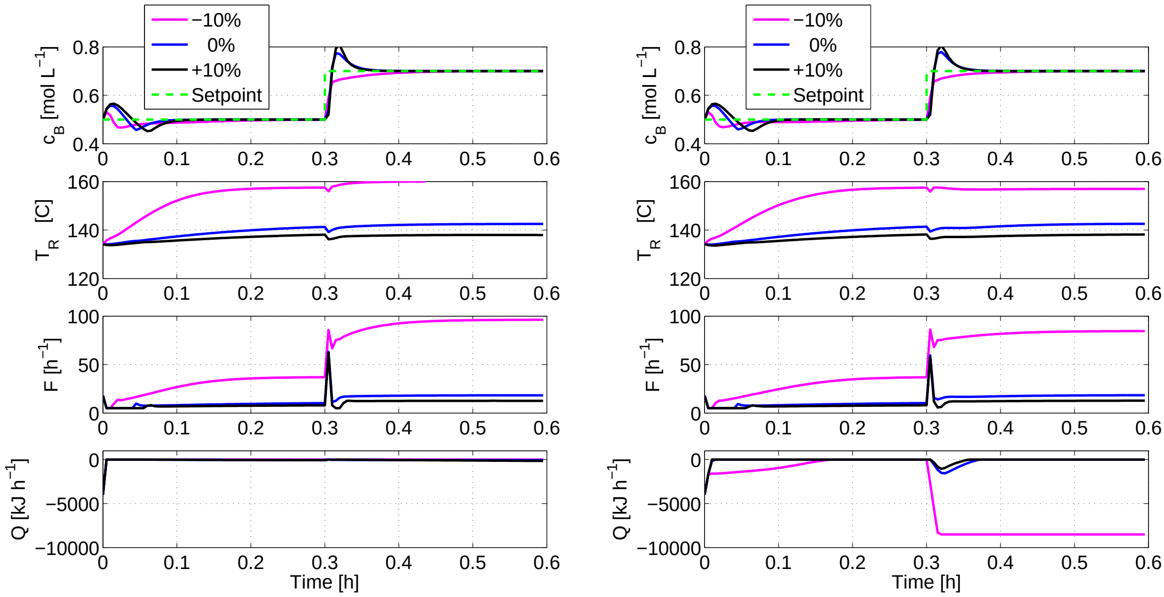

Figure A.3: Concentration of component B, reactor temperature, and control inputs obtained by standard NMPC (left) and by multi-stage NMPC (right) with bias term tracking a pre-defined setpoint for different values of the uncertain parameter (±10% w.r.t. the nominal value).

correction is calculated as:

$$\tilde{c}_{\mathrm{B}}^{\mathrm{ref},j} \leftarrow \tilde{c}_{\mathrm{B}}^{\mathrm{ref},j} + k_{\mathrm{bias}}(c_{\mathrm{B}}^{\mathrm{ref}} - c_{\mathrm{B}}^{\mathrm{meas}})\,|c_{\mathrm{B}}^{j} - c_{\mathrm{B}}^{\mathrm{meas}}|, \tag{A.6}$$

where $\tilde{c}_{\mathrm{B}}^{\mathrm{ref},j}$ is the reference that is used in the optimization problem. In this case there is a different bias term for each scenario j, which is obtained when comparing the predictions of the output at the last step c_{B}^{j} with the obtained measurement $c_{\mathrm{B}}^{\mathrm{meas}}$. As it is shown in Fig. A.4 both standard NMPC and multi-stage NMPC achieve a better performance with this modification compared to the bias update in (A.5). Again, there is no significant difference between standard and multi-stage NMPC.

The importance of using a robust approach can be seen in this case only if some constraints are active. Now it is considered that there is an upper bound on the temperature of the reactor that should not be violated, $T_{\mathrm{R}} \leq 155$. If standard NMPC with the modified bias term (A.6) is used, tracking is achieved but the constraint is violated (for the case when the uncertainty is 10% smaller than the nominal value). In contrast, multi-stage NMPC with the modified bias term realizes that the defined setpoint is unreachable and stays as close as possible without violating the temperature constraint as can be seen in Fig. A.5.

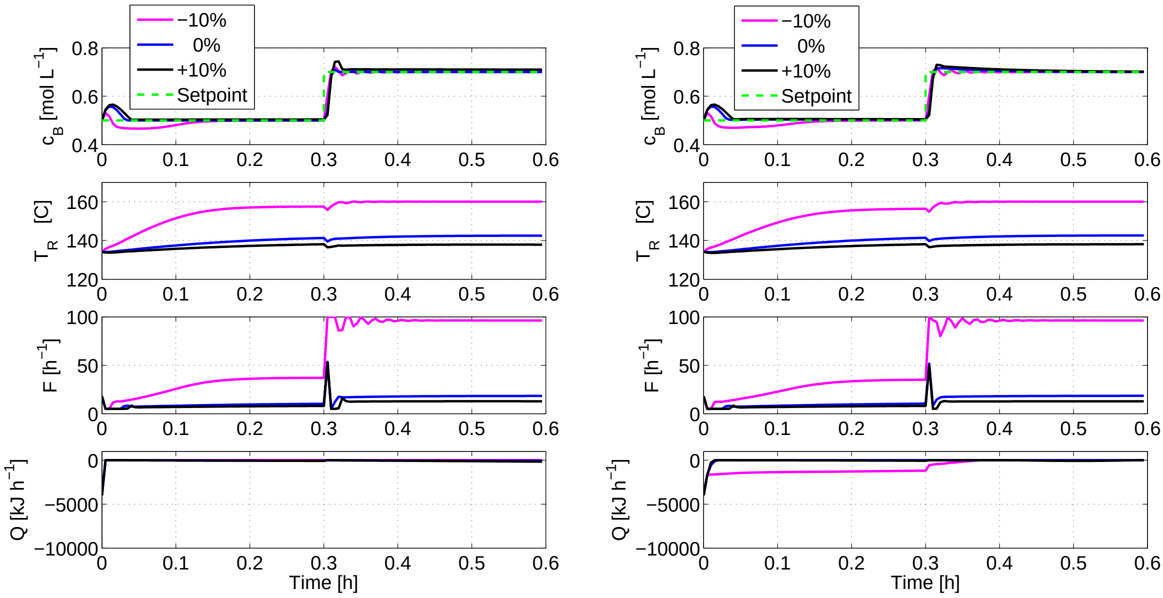

Figure A.4: Concentration of component B, reactor temperature, and control inputs obtained by standard NMPC (left) and by multi-stage NMPC (right) with modified bias term tracking a pre-defined setpoint for different values of the uncertain parameter (±10% w.r.t. the nominal value).

The main goal of this appendix is to show that the classical techniques applied to achieve steady-stage accuracy for standard NMPC can be applied in a straightforward manner to multi-stage NMPC. The rigorous analysis of the offset-free behavior is out of the scope of this appendix and the reader is referred to (Morari and Maeder, 2012) for an overview on the theoretical assumptions required to guarantee offset-free tracking in NMPC. If the use of measurement information results in a correction of the nominal model or its adaptation (at least approximately) the performance of standard NMPC can be in some cases comparable to the multi-stage case because uncertainty is (partially) removed. This has been illustrated for the very simple case of bias updates, but the central idea is applicable to other techniques used to achieve offset-free NMPC. Nevertheless, the use of multi-stage NMPC can be beneficial in the case of active state constraints. In that case, a modification of the cost function of the standard NMPC controller will not result in robust constraint satisfaction unless additional measures are taken while multi-stage NMPC achieves robustness without any modifications.

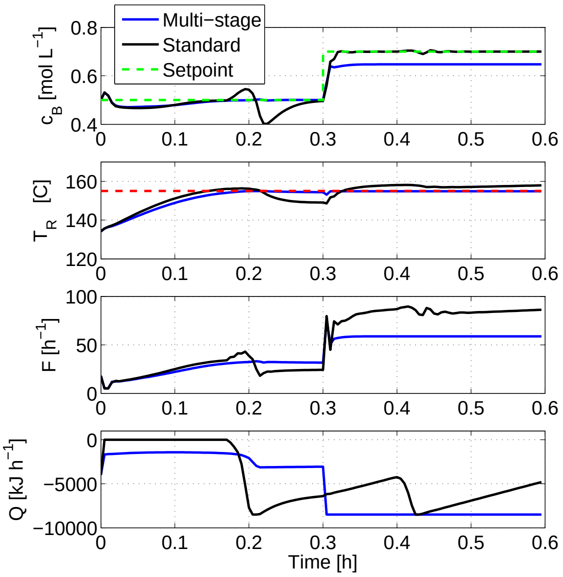

Figure A.5: Concentration of component B, reactor temperature (with constraint), and control inputs obtained by standard NMPC (left) and by multi-stage NMPC (right) with modified bias term tracking a pre-defined setpoint for a value of the uncertain parameter 10% smaller than the nominal value).

Appendix B

List of Symbols

Symbol	**Description**
x_k^j	Node of a tree at stage k and position j
$x_k^{p(j)}$	Parent node of the node x_{k+1}^j
u_k^j	Control input associated with the node x_k^j
y_k^j	Output of the system associated with node x_k^j
$\hat{u}_k^j$	Fictitious control input associated with the node x_k^j used in decomposition
$d_k^{r(j)}$	Uncertainty realization r at stage k and position j
I	Set containing all the occurring indices (j, k) for a given scenario tree
S_i	i-th scenario of a given scenario tree
X_i	Set of states of the scenario tree that belong to the i-th scenario
U_i	Set of control inputs of the scenario tree that belong to the i-th scenario
ω_i	Weight or probability of scenario S_i
π_k^j	Accumulated probability of node x_k^j
J_i	Cost of scenario S_i
L	Stage cost of the scenario-wise formulation of multi-stage NMPC
ℓ	Stage cost of the stage-wise formulation of multi-stage NMPC
V_f^r	Terminal penalty term associated with the realization r of the uncertainty
V_{N_P}	Optimal value of the cost function with prediction horizon N_P
$\mathbb{X}_f$	Terminal set
$\mathbb{X}_A(N_P)$	Feasibility region of an MPC controller with prediction horizon N_P

Symbol	Description
$\mathbb{X}$	Set defining the state constraints
$\mathbb{U}$	Set defining the input constraints
$\mathbb{D}$	Set of uncertainty realizations considered in the scenario tree
f	Discrete-time nonlinear dynamic model
g	Constraints on inputs and states
h	Output function (map of the state space to the output space)
n_x	Dimension of the state vector
n_u	Dimension of the control vector
n_d	Dimension of the uncertainty vector
N_P	Prediction horizon
N_R	Robust horizon
N	Number of scenarios
t_{step}	Sampling time of the NMPC controller
k_{var}	Tuning parameter to control variability of multi-stage NMPC
$\mathcal{X}_k^i$	Exact reachable set at time k for scenario i
$\bar{\mathcal{X}}_k^i$	Over-approximation of the reachable set at time k for scenario i
ϵ_k^j	Tightening of the constraints for the node x_k^j
s_k^j	Sensitivities of the states with respect to the parameters at node x_k^j
$s_{y,k}^j$	Sensitivities of the output with respect to the parameters at node x_k^j
$\mathcal{D}^i$	Uncertainty box associated to scenario i
D	Bounds of the uncertainty $D := [d^L, d^U]$
$w(D)$	Range of the uncertainty
$F_{N_e}^i$	Fisher Information Matrix associated to scenario i with a horizon N_e

Bibliography

V. Adetola, D. DeHaan, and M. Guay. Adaptive model predictive control for constrained nonlinear systems. *Systems & Control Letters*, 58:320–326, 2009.

A. AlGhazzawi and B. Lennox. Model predictive control monitoring using multivariate statistics. *Journal of Process Control*, 19:314–327, 2009.

M. Althoff and B.H. Krogh. Reachability analysis of nonlinear differential-algebraic systems. *IEEE Transactions on Automatic Control*, 59:371–383, 2014.

R. Amrit and J. Rawlings. Nonlinear model predictive control tools (NMPC tools). Online report, 2008. URL `http://jbrwww.che.wisc.edu/software/mpctools/mpc-tools-doc.pdf`.

R. Amrit, J. Rawlings, and D. Angeli. Economic optimization using model predictive control with a terminal cost. *Annual Reviews in Control*, 35:178–186, 2011.

J. Andersson. *A General-Purpose Software Framework for Dynamic Optimization.* PhD thesis, KU Leuven, 2013.

J. Andersson, J. Åkesson, and M. Diehl. CasADi – A symbolic package for automatic differentiation and optimal control. In S. Forth, P. Hovland, E. Phipps, J. Utke, and A. Walther, editors, *Recent Advances in Algorithmic Differentiation*, Lecture Notes in Computational Science and Engineering, pages 297–307, Berlin, 2012a. Springer.

J. Andersson, J. Åkesson, and M. Diehl. Dynamic optimization with CasADi. In *Proc. of the 51st IEEE Conference on Decision and Control*, pages 681–686, 2012b.

D. Angeli, R. Amrit, and J. Rawlings. On average performance and stability of economic model predictive control. *IEEE Transactions on Automatic Control*, 57:1615–1626, 2012.

H. Arellano-Garcia, T. Barz, M. Wendt, and G. Wozny. Real-time feasibility of nonlinear predictive control for semi-batch reactors. *Computer Aided Chemical Engineering*, 20: 967–972, 2005.

K.J. Åström and B. Wittenmark. Problems of identification and control. *Journal of Mathematical Analysis and Applications*, 34:90–113, 1971.

E.M.L. Beale. On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society*, 17(2):173–184, 1955.

R. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

A. Bemporad. Reducing conservativeness in predictive control of constrained systems with disturbances. In *Proc. of the 37th IEEE Conference on Decision and Control*, pages 1384–1389, 1998.

A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23:769–805, 1998.

A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99:351–376, 2004.

A. Ben-Tal, L. El-Ghaoui, and A. Nemirovski. *Robust Optimization.* Princeton University Press, 2009.

D. Bernardini and A. Bemporad. Scenario-based model predictive control of stochastic constrained linear systems. In *Proc. of the 48th IEEE Conference on Decision and Control, 2009.*, pages 6333–6338, 2009.

D. P. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, 2nd edition, 2000.

M. Beyer, W. Grote, and G. Reinig. Adaptive exact linearization control of batch polymerization reactors using a sigma-point Kalman filter. *Journal of Process Control*, 18: 663–675, 2008.

L.T. Biegler. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes.* SIAM, 2010.

J.R. Birge. Stochastic programming computation and applications. *INFORMS Journal on Computing*, 9(2):111–133, 1997.

R.R. Bitmead, M. Gevers, and V. Wertz. *Adaptive optimal control - The thinking man's GPC*. Prentice-Hall, 1990.

J. Bjornberg and M. Diehl. Approximate robust dynamic programming and robustly stable MPC. *Automatica*, 42:777–782, 2006.

H. Bock and K. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proc. of the 9th IFAC World Congress Budapest*, pages 242–247, 1984.

W. Bosarge and O. Johnson. Direct method approximation to the state regulator control problem using a Ritz-Trefftz suboptimal control. *IEEE Transactions on Automatic Control*, 15:627–631, 1970.

J. Bravo, T. Alamo, and Camacho E.F. Robust MPC of constrained discrete-time nonlinear systems based on approximated reachable sets. *Automatica*, 42:1745–1751, 2006.

A.E. Bryson and Y.C. Ho. *Applied Optimal Control*. Taylor & Francis, 1975.

G. Calafiore and M.C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51:742–753, 2006.

P.J. Campo and M. Morari. Robust model predictive control. In *Proc. of the American Control Conference*, pages 1021–1026, 1987.

M. Cannon, B. Kouvaritakis, S. Rakovic, and Q. Cheng. Stochastic tubes in model predictive control with probabilistic constraints. *IEEE Transactions on Automatic Control*, 56:194–200, 2011.

Y. Cao, S. Li, L.R. Petzold, and R. Serban. Adjoint Sensitivity Analysis for Differential-Algebraic Equations: The Adjoint DAE System and its Numerical Solution. *SIAM Journal on Scientific Computing*, 24(3):1076–1089, 2003.

B. Chachuat, B. Srinivasan, and D. Bonvin. Adaptation strategies for real-time optimization. *Computers and Chemical Engineering*, 33:1557–1567, 2009.

C. Y. Chen and B. Joseph. On-line optimization using a two-phase approach: An application study. *Industrial & Engineering Chemistry Research*, 26:1924–1930, 1987.

W. Chen, J. O'Reilly, and Ballance D.J. On the terminal region of model predictive control for non-linear systems with input/state constraints. *International Journal of Adaptive Control and Signal Processing*, 17:195–207, 2003.

R. Chylla and D. Haase. Temperature control of semi-batch polymerization reactors. *Computers & Chemical Engineering*, 17:257–264, 1993.

J. Cui and S. Engell. Medium-term planning of a multiproduct batch plant under evolving multi-period multi-uncertainty by means of a moving horizon strategy. *Computers & Chemical Engineering*, 34:598–619, 2010.

J.E. Cuthrell and L.T. Biegler. On the optimization of differential-algebraic process systems. *AIChE Journal*, 33:1257–1270, 1987.

C. R. Cutler and B.L. Ramaker. Dynamic matrix control - a computer control algorithm. In *Proc. of the Automatic Control Conference*, 1980.

Cybernetica. Cybernetica AS. http://www.cybernetica.no, 2014.

K. Dadhe and S. Engell. Robust nonlinear model predictive control: A multi-model nonconservative approach. In *Book of Abstracts, Int. Workshop on NMPC, Pavia*, page 24, 2008.

G. Dantzig. Linear programming under uncertainty. *Management Science*, 3/4:197–206, 1955.

M. Diehl, D. Leineweber, and A. Schäfer. *MUSCOD-II Users Manual.* IWR Preprint 2001-25, 2001.

M. Diehl, H.G. Bock, and J. Schlöder. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on Control and Optimization*, 43: 1714–1736, 2005.

J.C. Doyle. Guaranteed margins for LQG regulators. *IEEE Transactions on Automatic Control*, 23:756–757, 1978.

M. Ellis, M. Heidarinejad, and P. D. Christofides. Economic model predictive control of nonlinear singularly perturbed systems. *Journal of Process Control*, 23:743–754, 2013.

EMBOCON. EMBOCON EU Project (2014). http://www.embocon.org, 2014.

S. Engell. Feedback control for optimal process operation. *Journal of Process Control*, 17:203–219, 2007.

S. Engell. Online optimizing control: The link between plant economics and process control. In *10th International Symposium on Process Systems Engineering*, volume 27 of *Computer Aided Chemical Engineering*, pages 79 – 86. Elsevier, 2009.

L. Fagiano and M. Khammash. Nonlinear stochastic model predictive control via regularized polynomial chaos expansions. In *Proc. of the 51st IEEE Conference on Decision and Control*, pages 142–147, 2012.

M. Farina, L. Giulioni, L. Magni, and R. Scattolini. A probabilistic approach to model predictive control. In *Proc. of the 52nd IEEE Conference on Decision and Control*, pages 7734–7739, 2013.

A. A. Feldbaum. Dual control theory I. *Automation Remote Control*, 21:874–880, 1960.

M. Fiacchini, T. Alamo, and Camacho E.F. On the computation of convex robust control invariant sets for nonlinear systems. *Automatica*, 46:1334–1338, 2010.

T. Finkler, S. Lucia, M. Dogru, and S. Engell. Simple control scheme for batch time minimization of exothermic semibatch polymerizations. *Industrial & Engineering Chemistry Research*, 52:5906–5920, 2013.

G. Franceschini and S. Macchietto. Model-based design of experiments for parameter precision: State of the art. *Chem. Eng. Sci.*, 63(19):4846–4872, 2008. ISSN 0009-2509.

W. Gao and S. Engell. Iterative set-point optimization of batch chromatography. *Computers & Chemical Engineering*, 29:1401–1409, 2005.

M. Gevers, X. Bombois, R Hildebrand, and G. Solari. Optimal experiment design for open and closed-loop system identification. *Communications in Information and Systems*, 11:197–224, 2011.

J. D. Glover and F.C. Schweppe. Control of linear dynamic systems with set constrained disturbances. *IEEE Transactions on Automatic Control*, pages 411–423, 1971.

A. Gopalakrishnan and L. Biegler. Economic nonlinear model predictive control for periodic optimal operation of gas pipeline networks. *Computers & Chemical Engineering*, 52:90–99, 2013.

P. Goulart, E. C. Kerrigan, and J. M. Maciejowski. Optimization over state feedback policies for robust control with constraints. *Automatica*, 42:523–533, 2006.

K. Graichen, V. Hagenmeyer, and M. Zeitz. Feedforward control with online parameter estimation applied to the Chylla-Haase reactor benchmark. *Journal of Process Control*, 16:733–745, 2006.

S. Grammatico, X. Zhang, K. Margellos, P.J. Goulart, and J. Lygeros. A scenario approach for non-convex control design. *IEEE Transactions on Automatic Control (submitted)*, 2014.

A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, 2008.

G. Grimm, M.J. Messina, S.E. Tuna, and A.R. Teel. Examples when nonlinear model predictive control is nonrobust. *Automatica*, 40:1729–1739, 2004.

M.J. Hadjiyiannis, P.J. Goulart, and D. Kuhn. An efficient method to estimate the suboptimality of affine controllers. *IEEE Transactions on Automatic Control*, 56:2841–2853, 2011.

M. Herceg, M. Kvasnica, C.N. Jones, and M. Morari. Multi-parametric toolbox 3.0. In *Proc. of the European Control Conference*, pages 502–510, 2013.

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,*, 31:363–396, 2005.

B. Houska, H. Ferreau, and M. Diehl. An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. *Automatica*, 47:2279–2285, 2011a.

B. Houska, H.J. Ferreau, and M. Diehl. ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. *Optimal Control Applications and Methods*, 32:298–312, 2011b.

B. Houska, F. Logist, J. Van Impe, and M. Diehl. Robust optimization of nonlinear dynamic systems with application to a jacketed tubular reactor. *Journal of Process Control*, 22:1152–1160, 2012.

R. Huang, S.C. Patwardhan, and L.T. Biegler. Multi-scenario-based robust nonlinear model predictive control with first principle models. In *Proc of the 10th International Symposium on Process Systems Engineering*, volume 27, pages 1293 – 1298, 2009a.

R. Huang, V. Zavala, and L.T. Biegler. Advanced step nonlinear model predictive control for air separation units. *Journal of Process Control*, 19:678–685, 2009b.

R. Huang, S.C. Patwardhan, and L.T. Biegler. Robust stability of nonlinear model predictive control based on extended kalman filter. *Journal of Process Control*, 22: 82–89, 2012.

E. Idris and S. Engell. Economics-based NMPC strategies for the operation and control of a continuous catalytic distillation process. *Journal of Process Control*, 22:1832–1843, 2012.

IPCOS. IPCOS NV. http://www.ipcos.com, 2014.

L. Jaulin and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4):1053–1064, 1993.

Z.-P. Jiang and Y. Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37:857–869, 2001.

R.E. Kalman. Contributions to the theory of optimal control. *Bol. Soc. Mat. Mex.*, 5: 102–119, 1960.

H. Khalil. *Nonlinear Systems*. Prentice Hall, 2002.

V.L. Kharitonov. Asymptotic stability of an equilibrium position of a family of systems of linear differential equations. *Differentialnye Uravneniya*, 14:2086–2088, 1978.

K.U. Klatt and S. Engell. Gain-scheduling trajectory control of a continuous stirred tank reactor. *Computers & Chemical Engineering*, 22:491–502, 1998.

W. Kühn. Rigorously computed orbits of dynamical systems without the wrapping effect. *Computing*, 61:47–67, 1998.

M. Lazar, D. Muñoz de la Peña, W.P.M.H. Heemels, and T. Alamo. On input-to-state stability of min-max nonlinear model predictive control. *Systems & Control Letters*, 57(1):39 – 48, 2008.

J. H. Lee. From robust model predictive control to stochastic optimal control and approximate dynamic programming: A perspective gained from a personal journey. *Computers & Chemical Engineering*, 70:114–121, 2014.

J. H. Lee and Z. H. Yu. Worst-case formulations of model predictive control for systems with bounded parameters. *Automatica*, 33(5):763–781, 1997.

D. Limon, J. Bravo, T. Alamo, and Camacho E.F. Robust MPC of constrained nonlinear systems based on interval arithmetic. *IEE Proc.-Control Theory Appl.*, 152:325–332, 2005.

D. Limon, T. Alamo, F. Salas, and Camacho E.F. Input to state stability of min-max MPC controllers for nonlinear systems with bounded uncertainties. *Automatica*, 42: 797–803, 2006.

D. Limon, T. Alamo, D. Raimondo, D. de la Peña, J. Bravo, A. Ferramosca, and E. Camacho. Input-to-state stability: A unifying framework for robust model predictive control. In *Nonlinear Model Predictive Control*, volume 384 of *Lecture Notes in Control and Information Sciences*, pages 1–26. Springer Berlin / Heidelberg, 2009.

Y. Lin and M. A. Stadtherr. Validated solutions of initial value problems for parametric ODEs. *Appl. Numer. Math.*, 57(10):1145–1162, 2007.

J Löfberg. Approximations of closed-loop minimax MPC. In *Proc. of the 42nd IEEE Conference on Decision and Control*, pages 1438–1442, 2003.

S. Lucia and S. Engell. Multi-stage and two-stage robust nonlinear model predictive control. In *Proc. of the IFAC Conference on Nonlinear Model Predictive Control*, pages 181–186, 2012.

S. Lucia and S. Engell. Robust nonlinear model predictive control of a batch bioreactor using multi-stage stochastic programming. In *Proc. of the 12th European Control Conference, Zurich*, pages 4124–4129, 2013.

S. Lucia and S. Engell. Control of towing kites under uncertainty using robust economic nonlinear model predictive control. In *Proc. of the 13th European Control Conference*, pages 1158–1163, 2014.

S. Lucia and S. Engell. Potential and limitations of multi-stage nonlinear model predictive control. In *Proc. of the International Symposium on Advanced Control of Chemical Processes (accepted)*, 2015.

S. Lucia and R. Paulen. Robust nonlinear model predictive control with reduction of uncertainty via robust optimal experiment design. In *Proc. of the 19th IFAC World Congress*, pages 1904–1909, 2014.

S. Lucia, T. Finkler, D. Basak, and S. Engell. A new robust NMPC scheme and its application to a semi-batch reactor example. In *Proc. of the International Symposium on Advanced Control of Chemical Processes*, pages 69–74, 2012.

S. Lucia, T. Finkler, and S. Engell. Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty. *Journal of Process Control*, 23:1306–1319, 2013a.

S. Lucia, S. Subramanian, and S. Engell. Non-conservative robust nonlinear model predictive control via scenario decomposition. In *Proc. of the 2013 IEEE Multi-Conference on Systems and Control*, pages 586–591, 2013b.

S. Lucia, J. Andersson, H. Brandt, A. Bouaswaig, M. Diehl, and S. Engell. Efficient robust economic nonlinear model predictive control of an industrial batch reactor. In *Proc. of the 19th IFAC World Congress*, pages 11093–11098, 2014a.

S. Lucia, J. Andersson, H. Brandt, M. Diehl, and S. Engell. Handling uncertainty in economic nonlinear model predictive control: a comparative case-study. *Journal of Process Control*, 24:1247–1259, 2014b.

S. Lucia, D. Limon, and S. Engell. Stability properties of multi-stage robust nonlinear model predictive control. *Automatica (Submitted)*, 2014c.

S. Lucia, R. Paulen, and S. Engell. Multi-stage nonlinear model predictive control with verified robust constraint satisfaction. In *Proc. of the 54th IEEE Conference on Decision and Control*, pages 2816–2821, 2014d.

S. Lucia, A. Tatulea-Codrean, C. Schoppmeyer, and S. Engell. An environment for the efficient testing and implementation of robust NMPC. In *Proc. of the 2014 IEEE Multi-Conference on Systems and Control*, pages 1843–1848, 2014e.

S. Lucia, A. Tatulea-Codrean, and S. Engell. DO-MPC. https://github.com/sergiolucia/DO-MPC, 2015.

J. Maestre, L Raso, P. Van Overloop, and .B. De Schuuter. Distributed tree-based model predictive control on an open water system. In *Proc. of the American Control Conference*, pages 1985–1990, 2012.

L. Magni and R. Scattolini. Robustness and robust design of MPC for nonlinear discrete-time systems. In *Assessment and Future Directions of Nonlinear Model Predictive Control*, volume 358 of *Lecture Notes in Control and Information Sciences*, pages 239–254. Springer Berlin / Heidelberg, 2007.

K. Makino and M. Berz. *Remainder Differential Algebras and Their Applications*, chapter 5, pages 63–75. SIAM, 1996.

A. Marchetti, B. Chachuat, and D. Bonvin. Modifier-adaptation methomethod for real-time optimization. *Industrial & Engineering Chemistry Research*, 48:6022–6033, 2009.

T. E. Marlin and J. B. Hrymak. Real-time operations optimization of continous processes. In *AIChE Symposium Series-CPC-V*, 1997.

R. Marti, D. Sarabia, D. Navia, and C. De Prada. Coordination of distributed model predictive controllers using price driven coordination and sensitivity analysis. In *Proc. of the 10th IFAC International Symposium on Dynamics and Control of Process Systems*, 2013.

D. Q. Mayne, E. C. Kerrigan, and P. Falugi. Robust model predictive control: advantages and disadvantages of tube-based methods. In *Proc. of the 18th IFAC World Congress Milano*, pages 191–196, 2011.

D.Q. Mayne. Control of constrained dynamic systems. *European Journal of Control*, 7: 87–99, 2001.

D.Q. Mayne and E.C. Kerrigan. Tube-based robust nonlinear model predictive control. In *Proc. of the 7th IFAC Symposium on Nonlinear Control Systems*, pages 110–115, 2007.

D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789 – 814, 2000.

D.Q. Mayne, M.M. Seron, and S.V. Rakovic. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41:219 – 224, 2005.

A. Mesbah, S. Streif, R. Findeisen, and R.D. Braatz. Stochastic nonlinear model predictive control with probabilistic constraints. In *Proc. of the American Control Conference*, pages 2413–2419, 2014.

M. Morari and U. Maeder. Nonlinear offset-free model predictive control. *Automatica*, 48:2059–2067, 2012.

D. Muñoz de la Peña, T. Alamo, and A. Bemporad. A decomposition algorithm for feedback min-max model predictive control. In *Proc. of the 44th IEEE Conference on Decision and Control*, pages 5126–5131, 2005a.

D. Muñoz de la Peña, A. Bemporad, and T. Alamo. Stochastic programming applied to model predictive control. In *Proc. of the 44th IEEE Conference on Decision and Control*, pages 1361–1366, 2005b.

A. Munack. Optimization of sampling. *Measuring, Modelling and Control, Biotechnology*, 4:252–264, 1991.

Z.K. Nagy. OptCon - an efficient tool for rapid prototyping of nonlinear model predictive control applications. In *Proc. of the AIChE Annual Meeting*, pages 16–21, 2008.

J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 2006.

C.J. Ong, D. Sui, and E.G. Gilbert. Enlarging the terminal region of nonlinear model predictive control using the support vector machine method. *Automatica*, 42:1011–1016, 2006.

R. S. Patwardhan, S. L. Shah, and K. Z. Qi. Assessing the performance of model predictive controllers. *Canadian Journal of Chemical Engineering*, 80:954–966, 2002.

R. Paulen, M. Villanueva, and B. Chachuat. Optimization-based domain reduction in guaranteed parameter estimation of nonlinear dynamic systems. In *9th IFAC Symposium on Nonlinear Control Systems*, pages 564–569, 2013a.

R. Paulen, M. Villanueva, M. Fikar, and B. Chachuat. Guaranteed parameter estimation in nonlinear dynamic systems using improved bounding techniques. In *European Control Conference, Zurich, Switzerland*, pages 4514–4519, 2013b.

H. Pirnay, R. Lopez-Negrete, and L. Biegler. Optimal sensitivity based on IPOPT. *Mathematical Programming Computation*, 4:307–331, 2012.

B. Pluymers, J. Ludlage, L. Ariaans, and W. Van Brempt. An industrial implementation of a generic NMPC controller with application to a batch process. In *Proc. of the 17th IFAC World Cogress*, 2008.

L.S. Pontryagin, V.G. Boltyanskii, and E.F. Gamkrelidze, R.V. andMischenko. *The Mathematical Theory of Optimal Processes*. Wiley, 1962.

W.B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley, 2007.

W.B. Powell. A unified framework for stochastic and dynamic programming. *Informs Computing Society Newsletter*, 2012.

J. Qian, P. Dufour, and M. Nadri. Observer and model predictive control for on-line parameter identification in nonlinear systems. In *Proc. of the 10th IFAC International Symposium on Dynamics and Control of Process Systems*, 2013.

S. J. Qin and T.A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003.

Qt5.1. Qt5.1 framework. http://www.qt-project.org, 2014.

D. Raimondo, T. Alamo, D. Limon, and Camacho E.F. Towards the practical implementation of min-max nonlinear model predictive control. In *Proc. of the 46th IEEE Conference on Decision and Control*, 2007.

D. Raimondo, D. Limon, M. Lazar, L. Magni, and E. Camacho. Min-max model predictive control of nonlinear systems: A unifying overview on stability. *European Journal of Control*, 15(1):5–21, 2009.

M.R. Rajamani, J. Rawlings, and S. J. Qin. Achieving state estimation equivalence for misassigned disturbances in offset-free model predictive control. *AIChE Journal*, 2009.

S. Rakovic, B. Kouvaritakis, M. Cannon, C. Panos, and R Findeisen. Fully parameterized tube MPC. In *Proc. of the 18th IFAC World Congress Milano*, pages 197–202, 2011.

J. Rawlings and R. Amrit. Optimizing process economic performance using model predictive control. In *Nonlinear Model Predictive Control*, pages 119–138. Springer Berlin - Heidelberg, 2009.

J.B. Rawlings and D.Q. Mayne. *Model Predictive Control Theory and Design*. Nob Hill Pub, 2009.

S. Recker, N. Kerimoglu, A. Harwardt, O. Tkacheva, and W. Marquardt. On the integration of model identification and process optimization. In *Proc. of the 23rd European Symposium on Computer Aided Process Engineering*, pages 1021–1026, 2013.

J. Richalet, A. Rault, J.L. Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14:413–428, 1978.

P. Roberts. An algorithm for steady-state system optimization. *International Journal on System Science*, 10:719–734, 1979.

R.T. Rockafellar and R.J. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16:119–147, 1991.

A. Ruszczynski. Decomposition methods in stochastic programming. *Mathematical Programming*, 79:333–353, 1997.

N. Sahinidis. Optimization under uncertainty: state-of-the-art and opportunities. *Computers & Chemical Enggineering*, 28:971–983, 2004.

A.M. Sahlodin and B. Chachuat. Convex/concave relaxations of parametric ODEs using Taylor models. *Computers & Chemical Engineering*, 35(5):844 – 857, 2011. ISSN 0098-1354.

G. Sand and S. Engell. Modeling and solving real-time scheduling problems by stochastic integer programming. *Computers & Chemical Eng.*, 28:1087 – 1103, 2004.

R.W.H. Sargent. Optimal control. *Journal of Computational and Applied Mathematics*, 124:361–371, 2000.

R.W.H. Sargent and G.R. Sullivan. The development of an efficient optimal control package. In J. Stoer, editor, *Optimization Techniques*, volume 7 of *Lecture Notes in Control and Information Sciences*, pages 158–168. Springer Berlin Heidelberg, 1978.

C. Schoppmeyer. GEMS - Generic EMBOCON Minimal Supervisor. https://github.com/EMBOCONcs/GEMS, 2013.

P.O.M. Scokaert and D.Q. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic Control*, 43(8):1136–1142, 1998.

A. Shapiro. *Lectures on Stochastic Programming: Modeling and Theory.* SIAM, 2009.

A.B. Singer and P.I. Barton. Global optimization with nonlinear ordinary differential equations. *Journal of Global Optimization*, 2006.

S. Skogestad. Self-optimizing control: The missing link between steady-state optimization and control. *Computers & Chemical Engineering*, 24:569–575, 2000.

A.L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5):1154–1157, 1973.

B. Srinivasan, D. Bonvin, E. Visser, and S. Palanki. Dynamic optimization of batch processes II. role of measurements in handling uncertainty. *Computers & Chemical Engineering*, 27:27–44, 2002.

B. Srinivasan, S. Palanki, and D. Bonvin. Dynamic optimization of batch processes: I. Characterization of the nominal solution. *Computers & Chemical Engineering*, 27(1): 1–26, 2003.

M.C. Steinbach. Hierarchical sparsity in multistage convex stochastic programs. In *Stochastic Optimization: Algorithms and Applications*, pages 363–388. Kluwer Academic Publishers, 2000.

S. Subramanian. Evaluation of scenario decomposition approaches for the solution of robust nmpc problems. Master's thesis, TU Dortmund, 2012.

S. Subramanian, S. Lucia, and S. Engell. Economic multi-stage output nonlinear model predictive control. In *Proc. of the 2014 IEEE Multi-Conference on Systems and Control*, pages 1837–1842, 2014.

T.H. Summers, K. Kunz, N. Kariotoglou, M. Kamgarpour, S. Summers, and J. Lygeros. Approximate dynamic programming via sum of squares programming. In *Proc. of the European Control Conference*, pages 191–197, 2013.

A. Tatulea-Codrean. Development and evaluation of a configurable software platform for robust NMPC. Master's thesis, TU Dortmund, 2014.

T.H. Tsang, D.M. Himmelblau, and T.F. Edgar. Optimal control via collocation and nonlinear programming. *International Journal of Control*, 21:763–768, 1975.

O. Ubrich, B. Srinivasan, F. Stoessel, and D. Bonvin. Optimization of a semi-batch reaction system under safety constraints. In *European Control Conference, Karlsruhe, Germany*, pages F306.1–F306.6, 1999.

A. Wächter and L. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.

W. Walter. *Differential and integral inequalities.* Springer-Verlag, 1970.

H.S. Witsenhausen. A minimax control problem for sampled linear systems. *IEEE Transactions on Automatic Control*, 13:5–21, 1968.

D. Wolbert, X. Joulia, B. Koehret, and L.T. Biegler. Flowsheet optimization and optimal sensitivity analysis using exact derivatives. *Computers & Chemical Engineering*, 18: 1083, 1994.

S. Yu, H. Chen, and F. Allgöwer. Tube MPC scheme based on robust control invariant sets with application to Lipschitz nonlinear systems. In *Proc. of the 50th IEEE Conference on Decision and Control*, pages 2650–2655, 2011.

N. Zagrobelny, L. Ji, and J.B. Rawlings. Quis custodiet ipsos custodes? *Anual Reviews in Control*, 37:260–270, 2013.

G. Zames. Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Transactions on Automatic Control*, 26:301–320, 1981.

K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control.* Prentice Hall, 1996.