Doctoral theses at NTNU, 2024:197

Zawadi Mdoe

Nonlinear Model Predictive Control under Uncertainty

Enhancing efficiency, stability and robustness

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Natural Sciences
Department of Chemical Engineering

NTNU
Norwegian University of
Science and Technology

Zawadi Mdoe

Nonlinear Model Predictive Control under Uncertainty

Enhancing efficiency, stability and robustness

Thesis for the Degree of Philosophiae Doctor

Trondheim, May 2024

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Chemical Engineering

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Natural Sciences
Department of Chemical Engineering

ISBN 978-82-326-7986-7 (printed ver.)
ISBN 978-82-326-7985-0 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2024:197

Printed by NTNU Grafisk senter

To Naomi & Chantel

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy (PhD) at the Norwegian University of Science and Technology (NTNU). The work performed during this PhD was carried out at the Department of Chemical Engineering from September 2019 to August 2023, under the supervision of Prof. Johannes Jäschke (supervisor, Department of Chemical Engineering) and Onsager Prof. Lorenz Biegler (co-supervisor, Carnegie Mellon University, USA).

The doctoral thesis is a compilation of four research papers that I have authored in the duration of my PhD studies. I coauthored two other articles that are not included in this thesis. I also attended conferences, workshops, and meetings to present my findings. Thus, I acknowledge inputs from reviewers and editors of the channels where my research was presented, submitted, and published.

My motivation for writing this thesis is the growing interest in robust multi-stage nonlinear MPC in the process systems engineering (PSE) community. Also, industrial counterparts have begun to adopt nonlinear MPC to control processes. This thesis strives to bridge the gap between robust multi-stage nonlinear MPC theory and its practical implementation. I have suggested, in this thesis, new approaches to improve computational efficiency and robust performance while maintaining closed-loop stability. All experiments are numerical simulations and the limitations of the findings are clearly outlined.

The financial support for my research was granted by NTNU. I was also an associated PhD to FME HighEFF project that is financed by the Research Council of Norway. I had the opportunity to practice teaching duties for 25% of my time in the Department's courses. This enriched not only my research skills but it instilled a pedagogical perspective into my work.

Declaration of Compliance

I hereby declare that the thesis is an independent work in agreement with the Norwegian University of Science and Technology (NTNU) exam rules and regulations. This work is original and my own work and the sources I used were properly cited and acknowledged.

Trondheim, 15th May 2024
Zawadi Mdoe

Acknowledgment

I write this acknowledgment with a strange feeling being more than six years after I came to Norway to pursue a degree in Chemical engineering. This is the culmination of the most challenging experience of my life thus far, yet it has been extremely rewarding. There are certainly many people to thank at this point but I first give thanks, glory, and praise to my Lord and Savior, Jesus Christ.

I would like to express my deepest gratitude to my advisor, Prof. Johannes Jäschke, for his unwavering support, guidance, and encouragement throughout the entirety of my doctoral journey. His expertise, patience, and insightful feedback have been invaluable in shaping this dissertation. I also thank my co-supervisor, Onsager Prof. Lorenz Biegler from Carnegie Melon University for our discussions, which inspired the objectives of this research at its early stages. I am especially grateful to Mandar Thombre, Dinesh Krishnamoorthy, and Evren Turan for their fruitful collaborations leading to several co-authorships.

I also extend my gratitude to Prof. Sigurd Skogestad, who was the head of our research group when I started. He had a strong belief in me that encouraged me to embark on this journey. Thanks to Prof. Heinz Preisig, Prof. Nadav Bar, and Prof. Idelfonso Nogueira, for your valuable input from your experiences, especially in the weekly group meetings. My heartfelt appreciation goes to the Department of Chemical Engineering (IKP) and the Faculty of Natural Sciences (NV) at NTNU, for providing the resources and environment conducive to academic research and learning. Many thanks to the FME HighEFF – where I was affiliated – and Mo Fjernvarme AS for providing data used in this work. Together, they gave me opportunities for collaboration and growth that have been instrumental in my development as a scholar.

I am deeply grateful to my colleagues and peers for their camaraderie, support, and intellectual exchange. A special mention to those I shared office K4-239 with; Mandar, Cristina, David, Håkon, Allyne, Evren, Mauricio, Erbet, and Lena-Marie – Thank you, for all the spontaneous chats and coffee breaks! I cannot forget Adriaen, Ana, Andrea, Bahareh, Carine, Carol, Fabienne, Halvor, Igor, José, Leonardo, Lucas B., Lucas C., Marius, Md, Pedro, Peter, Rafael, Risvan, Robert, Saket, Simen, and Timur. Together they created a dependable social circle, with joyful chats at lunch and the famous Friday happy hours. Their diverse perspectives and friendships have made this academic journey all the more enriching and enjoyable.

Since September 2023, I've been employed at ABB Efficiency Services while simultaneously working on my thesis during after-work hours. I feel fortunate to collaborate with talented and amiable individuals as my new coworkers, and I'm thrilled to tackle exciting process control challenges with them. I express my gratitude to them for their support during the concluding months of this endeavor.

I am immensely thankful to my PhD evaluation committee, Prof. Radoslav Paulen and Prof. Roshan Sharma, for giving their time and effort to read and evaluate this thesis. I am grateful for their valuable insights, constructive criticism, and dedication to ensuring the quality of this work.

To my family to which this work is dedicated, especially to my beloved wife, Naomi, thank you for being my anchor and for your unwavering love. And my bundle of joy, Chantel, thank you for your understanding and patience when I chose to write rather than spend time with you. To my mom, Justina, thank you for your love, prayer, and encouragement. And to my dad, Ntengua, thank you for paving the way and passing on to me the best qualities. Your belief in me has been a constant source of strength and motivation.

Lastly, I want to express my sincere gratitude to all those whose names may not appear here but have contributed in ways both seen and unseen to the completion of this dissertation. Your support, whether in the form of encouragement, assistance, or inspiration, has been deeply appreciated.

Thank you all for being part of this incredible journey.

Zawadi Mdoe,
Trondheim, Norway
May 2024

Abstract

Model predictive control is an optimization-based control strategy that has become increasingly popular in industrial applications. This popularity stems from its inherent capacity to handle complex control problems, such as those found in multiple-input-multiple-output systems with inequality constraints. The current actions of the MPC control strategy are obtained by optimizing an objective for a certain look-ahead period. The constraints and objectives of the MPC problem can be either linear or nonlinear, resulting in linear MPC and nonlinear MPC, respectively. Nonlinear MPC is sought due to the greater accuracy offered by nonlinear process models. Moreover, robust MPC schemes have garnered widespread attention to ensure that the control problem accounts for uncertainty. However, many of these methods exhibit a high degree of conservativeness and have significant limitations in practical implementation. This doctoral thesis focuses on multi-stage MPC, a robust MPC approach that transcribes uncertainty into the optimization problem using a scenario tree. The approach considers that recourse actions can be taken, thus achieving robustness with lower conservativeness.

One significant drawback of multi-stage MPC is poor scalability with increasing dimensionality of uncertainty. Assembling the scenario tree results in a computational burden that grows exponentially for larger problems. Furthermore, conventional scenario selection approaches are based on heuristics that may be unsuitable for the specific application, potentially resulting in unnecessary conservativeness. This thesis addresses the aforementioned issues for multi-stage MPC, particularly for robust nonlinear control, by proposing new methods and algorithms.

The first part of the thesis tackles computational delay in multi-stage MPC by approximation strategies. In this part, a strategy is proposed to reduce the prediction horizon as the controlled system approaches its optimal equilibrium. This approach is based on performing approximate solution predictions of the subsequent iterations and determining an updated horizon that will maintain closed-loop stability. The stability criterion is based on an approximation of terminal cost and a common terminal region for the scenarios. This adaptive horizon approach in multi-stage MPC demonstrates comparable robust control performance to conventional fixed horizon multi-stage MPC, albeit at a significantly reduced computational cost. Finally, a theoretical analysis shows the conditions for recursive feasibility and robust stability of the proposed methodology.

The second part of the thesis addresses the issue of conservativeness by proposing scenario selection techniques that are representative of the actual process. In this part, an approach combining multivariate data analysis and nonlinear program sensitivity analysis is proposed. The proposed method aims to utilize existing correlations in available uncertainty data and assemble a more compact scenario tree that closely aligns with the process operation domain, thus reducing conservativeness. Additionally, sensitivity analysis is employed to identify data-informed scenarios that are most likely to lead to constraint violations. The findings show that the data-driven sensitivity-assisted multi-stage MPC shrinks the optimization problem size and has a reduced conservativeness while being robust and computationally efficient. This approach is also applied to a detailed case study on thermal energy storage, demonstrating a reduction in peak heating requirements and an enhanced robustness to uncertainty in energy supply and demand.

Contents

List of Tables

List of Figures

Acronyms

CRCQ	Constant Rank Constraint Qualification
CSTR	Continuously-Stirred Tank Reactor
DAEs	Differential and Algebraic Equations
GSSOSC	General Strong Second-Order Sufficient Conditions
IPOPT	Interior Point OPTimizer
ISpS	Input-to-State practical Stability
KF	Kalman Filter
KKT	Karush-Kuhn-Tucker
LICQ	Linear Independent Constraint Qualification
LQR	Linear Quadratic Regulator
LQG	Linear Quadratic Gaussian
MFCQ	Mangasarin-Fromovitz Constraint Qualification
MINLP	Mixed-Integer Nonlinear Program
MPC	Model Predictive Control
MSMPC	Multi-stage Model Predictive Control
NACs	Non-Anticipativity Constraints

NLP	Nonlinear Program
NMPC	Nonlinear Model Predictive Control
ODEs	Ordinary Differential Equations
PC	Principal Component
PCA	Principal Component Analysis
PID	Proportional Integral Derivative
PSE	Process Systems Engineering
RPI	Robust Positively Invariant
SAMNMPC	Sensitivity-Assisted Multi-stage NMPC
SC	Strict complementarity
SSOSC	Strong Second-Order Sufficient Conditions
TES	Thermal Energy Storage

1 Introduction

"This attitude of mind – this attitude of uncertainty – is vital to the scientist, and it is this attitude of mind which the student must first acquire. It becomes a habit of thought. Once acquired, we cannot retreat from it anymore"

Richard P. Feynman (1918-1988)

1.1 Motivation

Industrial processes are subject to different operating conditions, safety limits, and product specifications, often involving complex dynamics. As global energy demands continue to rise, optimizing these processes from both environmental and economic standpoints is crucial. However, a significant challenge in achieving this is that most real-world processes lack complete information and are affected by considerable uncertainty. Advanced process control methods are employed to optimize processes despite such uncertainty.

Model predictive control (MPC) is a powerful tool commonly used for control and optimization in the chemical process industry due to its ability to handle complex multivariable systems with process constraints by computing an optimal control trajectory that minimizes a specific cost function over a prediction horizon [1]. Assuming a good systems model and disturbance predictions, MPC proves more advantageous than classical control structures for rejecting disturbances, particularly in frequently disturbed processes, for example, energy systems. This is due to its ability to optimize transient operations effectively. However, in industrial applications, MPC is commonly applied in the upper layer of the control hierarchy with a slower time scale, above the classical control structures (PID controllers) that make faster adjustments to stabilize the process locally.

As plant dynamics are often nonlinear, the nonlinear counterpart of MPC, known as nonlinear MPC (NMPC), has recently gained considerable attention. Nevertheless, the performance of model-based controllers is impacted by the accuracy of the model in describing the real system and the process disturbances affecting the system dynamics. Plant-model mismatch can cause the controlled system to violate constraints or even become unstable. While the conventional MPC, also known as standard MPC, provides some inherent robustness against uncertainty, it becomes insufficient when the uncertainty is significant. Therefore, robust MPC approaches that effectively handle uncertainty have garnered attention in recent decades.

An increasingly popular robust MPC approach is the multi-stage MPC, which is based on the principles of multi-stage stochastic programming. Multi-stage MPC is a scenario-based approach that aims at finding an optimal control input while satisfying constraints corresponding to all possible scenarios in a scenario tree. The earliest ideas of scenario-based MPC on linear systems appear in a paper by Scokaert and Mayne [2], where it was referred to as “feedback min-max MPC”. The approach was extended to nonlinear systems by Lucia [3], and was termed multi-stage NMPC.

In stochastic programming, the decision variables may take one of two forms: *here-and-now* variables, representing decisions that must be made before the uncertainty is realized, and *wait-and-see* variables, representing decisions that must be made after the uncertain data becomes known, and that can be used to hedge against future realizations of the uncertainty [4]. The latter variables allow for recourse actions that can reduce the conservativeness of the *here-and-now* decision variables. This is the fundamental concept behind the multi-stage MPC approach, where future control decisions serve as recourse action to the uncertainty evolution in response to the current control decision. A key feature of this approach is its robustness in terms of constraint feasibility without being overly conservative or cautious with control decisions.

Multi-stage MPC has been successfully applied to various applications, such as to semi-batch polymerization [5–8], a batch bioreactor [9], hydrodesulfurization [10], gas lifted wells in oil and gas production [11], multi-product distillation [12], and thermal energy storage operation [13] providing robust constraint satisfaction.

The multi-stage MPC framework involves modeling parametric uncertainty using a scenario tree that tracks uncertainty over time, with each scenario representing a unique parameter realization across the prediction horizon. The finite realizations of uncertain parameters are combined to form each scenario, and the controller performance is influenced by the selection of these realizations, particularly in terms of how conservative the controller is. The scenario tree formulation results in an exponential growth in the number of scenarios with both the number of uncertain parameters and the number of discrete realizations of each parameter, which increases problem size and computational complexity. This causes a time delay in obtaining solutions, which may result in suboptimality or closed-loop instability.

Therefore, the primary focus of this doctoral thesis centers around two important aspects of multi-stage MPC:

1. Improving the computational efficiency by limiting the large problem size that results from the exponentially growing scenario tree.
2. Selection of scenarios and disturbance realizations in the scenario tree that best capture the uncertainty hence avoiding unnecessary conservativeness.

1.1.1 Improving computational efficiency

Improving the computational efficiency in multi-stage MPC can be addressed by (i) decomposition and (ii) approximation methods. Decomposition methods decouple all the scenarios and solve smaller subproblems separately, while approximation methods replace the larger problem with a smaller problem without losing its main features.

Currently, primal decomposition [14] and dual decomposition [15, 16] algorithms have been proposed for multi-stage MPC. Approximation methods that have been proposed include the use of neural networks to approximate the cost-to-go functions in each scenario [17], advanced-step MPC [18], and an online scenario tree generation [19]. The online approximation method by Holtorf et al. [19] approximates the multi-stage MPC with a smaller scenario tree by first identifying the worst-case uncertainty realizations with respect to constraint feasibilities. Further, Thombre et al. [20] proposes a nonlinear programming (NLP) sensitivity-based approximation strategy for multi-stage NMPC, which prunes the scenario tree by identifying scenarios most likely to cause constraint violations.

In Part I of this thesis, an online prediction horizon update algorithm for the multi-stage NMPC is proposed. The strategy reduces the multi-stage NMPC problem size and maintains its closed-loop stability properties [21, 22]. The horizon update is efficiently determined online with the aid of NLP sensitivity analysis, and pre-determined terminal ingredients (i.e. terminal cost and terminal region).

1.1.2 Scenario selection

The best way to select scenarios in the scenario tree of the multi-stage MPC is to find uncertain parameter realizations that most accurately describe the system's uncertainty. In nonlinear systems, one can treat uncertainty by using first-order approximations of the process model with respect to the uncertain parameters [23–25]. Moreover, the formulation may include probabilistic chance constraints generated by polynomial chaos expansion to propagate the uncertainty through the system model [26–28].

A heuristic by Lucia et al. [6] suggests building a scenario tree by taking combinations of {max, nominal, min} values of the uncertain parameter ranges as discrete realizations in each stage. However, this heuristic may lead to highly conservative performance, and computational delay due to having many unnecessary constraints and variables in the optimization problem. To obtain less conservative but robust solutions, a range reduction using a dynamic design of experiments has been suggested by Lucia and Paulen [29]. Also use of recursive Bayesian weighting approaches has been proposed by Krishnamoorthy et al. [30]. These approaches focus on tightening the uncertainty set but do not necessarily ensure computational efficiency. To achieve computationally efficient scenario selection, data-driven scenario selection approaches based on sampled uncertainty data are proposed in Part II of this thesis.

1.2 Scope of the thesis

In the author's view, this thesis aims to improve the multi-stage MPC framework in two ways.

1. Increasing the computational efficiency while maintaining its robust performance, and
2. More accurate uncertainty modeling, for a less conservative, efficient,

and robust performance.

They are achieved by incorporating multivariate data analysis, nonlinear systems theory, nonlinear optimization theory, and sensitivity analysis. These method extensions enable efficient real-time implementation of robust MPC on large-scale problems, which can be formulated easily. The performance of these approaches is evaluated through various case studies and shown to outperform standard MPC in terms of constraint satisfaction, and conventional multi-stage MPC in the computational speed metric.

1.3 Thesis structure and main contributions

Apart from this chapter, the introductory part includes three more chapters with preliminary information before the main part of the thesis. Chapter 2 briefly describes model predictive control, MPC stability theory, and nonlinear programming theory. Chapter 3 presents the relevant background for multi-stage NMPC and its formulations. Chapter 4 presents a computationally efficient multi-stage MPC formulation known as the sensitivity-assisted multi-stage NMPC (SAMNMPC).

As explained in the previous section, this thesis consists of two major parts that are organized as follows: Part I focuses on achieving multi-stage MPC with fast computation times. The goal is to improve the computational efficiency of multi-stage MPC by reducing the problem size using an approximation method. In Chapter 5, an adaptive horizon multi-stage MPC algorithm is presented that gives a prediction horizon update at each multi-stage MPC iteration. This chapter shows a significant reduction of computational delay as the controlled system approaches its optimal equilibrium point. The approach is based on the approximation of terminal cost and region for closed-loop stability and an approximate prediction of the solution in the subsequent iteration using nonlinear optimization theory and sensitivity analysis. The results in this chapter are published in [21] and [22].

Chapter 6 presents a recursive feasibility and robust stability analysis of the proposed adaptive horizon multi-stage MPC. The results show that the recursive feasibility of the framework is guaranteed with a fully branched scenario tree. A relaxed formulation can be implemented in practice to avoid infeasibility but the robust constraint satisfaction of multi-stage MPC is given up. Provided that the recursive feasibility property and terminal conditions such as a non-empty common terminal region are satisfied, the closed-loop system is proven to be input-to-state practical stable (ISpS). This result also applies to the multi-stage MPC formulation with a robust horizon. These results are published in [22].

Part II is concerned with improving robustness and reducing conservativeness of the multi-stage MPC by improved scenario selection. The general idea is to combine data-driven methods and NLP sensitivities for better scenario selection. Chapter 7 examines with the aid of a simple optimization with uncertainty problem, the benefit of using both PCA and NLP sensitivities to select scenarios for the approximate optimization problem. This new approach was compared with other scenario selection approaches such as the conventional box over-approximation, PCA-based (PCA only, without sensitivities) approach, and sensitivity-assisted (without PCA) approach. The proposed approach demonstrated the best cost at the solution and the smallest problem size. The content of this chapter is based on an unpublished article [31].

In Chapter 8 the proposed scenario selection approach is applied to the multi-stage MPC formulation. The aim is to improve the performance of the overly conservative SAMNMPC using realizations from principal components. This PCA-SAMNMPC framework is a SAMNMPC with linearly transformed model parameters such that the sensitivities of the MPC problem are evaluated along the principal components instead. The results show that the PCA-SAMNMPC when compared to SAMNMPC has a significant reduction in conservativeness while maintaining its robustness and computational efficiency. Moreover, the results demonstrated on a simplified energy network with a thermal storage tank that the solution increases savings in peak heating compared to the PCA-based strategy. This chapter is mainly adopted from [32] and [31].

The thesis concludes with a summary of the contributions in each chapter, highlights its limitations, and provides possible avenues for future research.

1.4 List of publications

During the PhD study, a total of six articles have been produced. This thesis includes results from four of the articles. Some of the results have been presented in the form of oral presentations on several occasions. Detailed lists of the research papers and presentations are presented below.

Papers included in the main part of this thesis

1. Zawadi Mdoe, Dinesh Krishnamoorthy, and Johannes Jäschke. Stability properties of the adaptive horizon multi-stage MPC. *Journal of Process Control.* 128: 103002, 2023. **- Chapters 5 and 6**
2. Zawadi Mdoe, Dinesh Krishnamoorthy, and Johannes Jäschke. Adaptive Horizon Multi-stage Nonlinear Model Predictive Control. *In 2021 American Control Conference (ACC).* pages 2088-2093. IEEE, 2021. − **Chapter 5**
3. Zawadi Mdoe and Johannes Jäschke “Scenario selection for multi-Stage MPC using NLP sensitivities along principal components: application to robust optimal operation of thermal energy storage.” *Computers & Chemical Engineering.* (Under review), 2024. − **Chapters 7 and 8**
4. Zawadi Mdoe, Mandar Thombre, and Johannes Jäschke. Data-driven online scenario selection for multi-stage NMPC. *Computer Aided Chemical Engineering.* Vol. 49. pages 1627-1632. Elsevier, 2022. − **Chapter 8**

Papers not included in this thesis

1. Evren Turan, Zawadi Mdoe, and Johannes Jäschke. Learning convex objectives to reduce the complexity of model predictive control. *Systems & Control Letters.* IEEE, 2023. (Under review) [33]
2. Mandar Thombre, Zawadi Mdoe, and Johannes Jäschke. Data-driven robust optimal operation of thermal energy storage in industrial clusters.

Processes 8.2: 194, 2020. [13]

Accepted abstracts and invited presentations

1. Zawadi Mdoe, Dinesh Krishnamoorthy, and Johannes Jäschke. Improving the computational efficiency of multi-stage NMPC using an adaptive horizon. *AIChE Annual Meeting.* Phoenix, AZ, USA, November 2022. - Accepted abstract.
2. Zawadi Mdoe and Johannes Jäschke. Improving the computational efficiency of robust multi-stage MPC. *Internal Ph.D./Post.Doc Seminar HighEFF.* Trondeim, Norway, May 2022. - Invited.
3. Zawadi Mdoe, Dinesh Krishnamoorthy, and Johannes Jäschke. Adaptive horizon multi-stage nonlinear MPC: Stability and Recursive feasibility. *23rd Nordic Process Control Workshop.* Luleå, Sweden, March 2022. - Accepted abstract.
4. Zawadi Mdoe, Mandar Thombre, and Johannes Jäschke. Data-driven online scenario selection for multi-stage NMPC. *AIChE Annual Meeting.* Boston, MA, USA, November 2021. - Accepted abstract.
5. Zawadi Mdoe, Mandar Thombre, and Johannes Jäschke. Data-driven online scenario selection for multi-stage NMPC. *2021 HighEFF Cross-sector Workshop.* Hell, Norway, October 2021. - Invited.

Preliminaries & Theoretical Foundations

2 | Model predictive control

"Kupanga ni kuchagua"

Mwl. Julius K. Nyerere (1922-1999)

This chapter presents a short historical background on optimal control using model predictive control (MPC) and introduces the MPC algorithm. After introducing the general notation used in the thesis, the standard MPC formulation is presented. Further, some definitions of terms used in nonlinear stability theory are presented. Since MPC formulations generally involve nonlinear optimization problems, this chapter also gives a brief introduction to nonlinear optimization theory and properties that are used in the main parts of the thesis.

2.1 Background

Model predictive control (MPC) is a model-based control strategy that originates from optimal control theory. Optimal control theory involves solving for the extremum of a function that is subject to system dynamics usually modeled as continuous-time ordinary differential equations (ODEs) or differential and algebraic equations (DAEs). Over the years, the optimal control field has focused on solving these infinite-dimensional problem formulations which are generally not easy to solve.

In the special case of unconstrained linear systems, the analytic solution that minimizes an infinite time quadratic cost is the Riccati differential equation. This is called the linear quadratic regulator (LQR). On the other hand, a linear quadratic estimator (LQE) problem finds the optimal state estimates when a model is affected by process noise and/or noisy measurements. The Kalman filter (KF) proposed by Kalman in 1960 converges to the optimal solution (the "truth") using a recursive formulation. Combining the two problems (the LQR and the KF) results in the linear quadratic Gaussian (LQG) problem.

The robustness of LQG was famously investigated by Doyle in 1978 and the controller was not found to be robust even for very small perturbations. Therefore, the robustness of linear controllers was brought to light and gained strong attention resulting in robust linear control theory. The most common approaches to the robust synthesis and analysis of linear controllers include set-theoretic methods such as using bounding ellipsoids in [36], Kharitonov's methods [37], and $\mathcal{H}_\infty$ control [38].

Until the 1970s, the main focus was to find analytical solutions to optimal control problems. However, finding the analytical solution for nonlinear systems, including the robust optimal control problem, is impossible. Therefore, to solve general optimal control problems, numerical methods are often required. These methods are typically divided into two categories: *indirect* and *direct* methods.

Indirect methods involve formulating the optimality conditions, such as those derived from Pontryagin's minimum principle [39], and then discretizing them to obtain a numerical solution. As a result, these methods are sometimes referred to as *first optimize then discretize* methods, as explained in the article by [40].

Direct methods to solve optimal control problems were introduced by several researchers including Bosarge and Johnson [41]. These methods include the control vector parametrization approach [42], the multiple shooting approach [43], and the collocation approach [44, 45]. Direct methods are also known as *first discretize then optimize* methods since the system dynamics and control inputs are initially discretized to form a nonlinear programming problem (NLP) to be solved numerically. These methods are commonly used in dynamic optimization applications including in model predictive control. Moreover, for robust optimal control problems, direct methods transform the problem into an optimization problem making it possible to apply general techniques used for optimization under uncertainty such as stochastic programming principles.

2.2 MPC algorithm

MPC framework was initially proposed by Richalet et al. [46] and Cutler and Ramaker [47] as a control heuristic in industrial processes. MPC has been used in industrial processes even before most of its theory was developed. The MPC control strategy is described by Mayne et al. [48] as:

> A form of control in which the current control action is obtained by solving online, at each sampling instant, a finite horizon open-loop optimal control problem, using the current state of the plant as the initial state; the optimization yields an optimal control sequence and the first control in this sequence is applied to the plant.

In MPC, a process system is optimized with respect to a control objective subject to constraints at each sampling time. It computes at each sampling time a sequence of optimal control inputs by solving an optimization problem that minimizes some pre-defined cost function over a finite prediction horizon, subject to the state and input constraints at each time step in the future. Only the first control input is applied to the plant and the process is repeated at the next sampling time using the new state measurements or estimates as the initial condition.

MPC includes constraints for online decision-making and has a good control performance even when the system is disturbed away from the desired reference trajectory. The equality constraints include the system dynamics which are usually a set of DAEs for most chemical processes, and the inequality constraints are usually bounds for the control inputs and states. The control inputs are usually considered piece-wise constants in between time intervals.

Most models are inaccurate causing plant-model mismatch, usually as a result of uncertainty caused by (i) unknown changes in process behavior and (ii) measurement noise. It is necessary to have feedback to control such an uncertain system. In MPC, steady-state accuracy can be achieved by adding a bias correction term to the predicted output that is obtained from the last measurements. Feedback information enters the MPC control loop only by the re-initialization of the optimization problem and the bias term, hence the controller performance and closed-loop stability are strongly reliant on the model accuracy.

2.2.1 System notation

As previously stated, chemical processes are generally represented by a set of dynamic and algebraic equations (DAEs). There exist several techniques for solving DAEs [49], and collocation methods, in particular, have demonstrated high efficacy for the simultaneous optimization of DAEs [50, 51]. Hence, for the objectives of this study, the direct focus is on a general uncertain discrete-time system that is written as follows:

$$x_{k+1} = f(x_k, u_k, d_k) \tag{2.1}$$

where $x_k \in \mathbb{X} \subset \mathbb{R}^{n_x}$ is a state vector that fully defines the system model at time t_k where $k \geq 0$, $u_k \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$ is a control input vector that is implemented at time t_k, and $d_k \in \mathbb{D} \subset \mathbb{R}^{n_d}$ is the uncertain parameter (disturbance) vector realized at time t_k. The set of real numbers is denoted by $\mathbb{R}$, and the set of integers by $\mathbb{Z}$, with the subscript $_+$ referring to their non-negative counterparts. The Euclidean vector norm is $|\cdot|$ and the corresponding induced matrix norm is $\|\cdot\|$.

The nominal parameter value (vector) is denoted as d^0 with the nominal model written as:

$$x_{k+1} = f(x_k, u_k) := f(x_k, u_k, d^0) \tag{2.2}$$

where the function $f\colon \mathbb{R}^{n_x+n_u+n_d} \mapsto \mathbb{R}^{n_x}$ is assumed to be twice differentiable in all its arguments, has Lipschitz continuous second derivatives, and can be transformed to obtain the steady state nominal model $f(0,\, 0,\, d^0) = 0$ where d^0 is the nominal disturbance.

2.3 Standard MPC

The standard MPC for the system (2.1) is defined as follows. The MPC receives the current state x_k and assumes a nominal parameter d^0. The optimal control problem for standard MPC is an NLP, parametric in x_k and d^0 and can be written as follows:

$$\begin{aligned}
V_N^{\text{nom}}(x_k, d^0) = \min_{z_i, \nu_i} \quad & \psi(z_N, d^0) + \sum_{i=0}^{N-1} \ell(z_i, \nu_i, d^0) && \text{(2.3a)}\\
\text{s.t.} \quad & z_{i+1} = f(z_i, \nu_i, d^0), \quad i = 0, \dots, N-1 && \text{(2.3b)}\\
& z_0 = x_k, && \text{(2.3c)}\\
& z_i \in \mathbb{X},\ \nu_i \in \mathbb{U}, && \text{(2.3d)}\\
& z_N \in \mathbb{X}_f && \text{(2.3e)}
\end{aligned}$$

where N is the prediction horizon, V_N^{nom} is the optimal cost function, z_i and ν_i are the predicted state and input variable vectors, respectively at time t_{k+i}. The objective (2.3a) consists of a stage cost $\ell\colon \mathbb{R}^{n_x+n_u+n_d} \mapsto \mathbb{R}$, and a terminal cost $\psi\colon \mathbb{R}^{n_x+n_d} \mapsto \mathbb{R}$. The initial condition (2.3c) is the current state x_k. The equations (2.3b) are constraints predicting system dynamics, and (2.3d) are the bounds on state and input variables. The terminal constraints are in (2.3e), where $\mathbb{X}_f \subset \mathbb{X}$ is the terminal region set. This thesis consistently assumes that all the vector sets are subsets of the real-vector space. Further, $\mathbb{X}$ and $\mathbb{X}_f$ are closed, while $\mathbb{X}_f$ and $\mathbb{U}$ are compact. Additionally, the constraint sets defined by (2.3d) are treated as box constraints unless explicitly stated otherwise.

The standard MPC receives x_k and solves (2.3) for an optimal input sequence $\boldsymbol{v}^{\star}_{[0,N-1]}$ and their corresponding state predictions $\mathbf{z}^{\star}_{(0,N]}$ in the time interval $(t_k,\, t_{k+N}]$ at every iteration k. The first stage input from the sequence is applied to the plant as u_k. Then the prediction horizon is shifted one step, and the problem is reinitialized from the new state x_{k+1}.

Since the current optimal input is obtained with respect to x_k, the standard MPC policy yields an implicit feedback control law $u_k = \kappa(x_k)$, where $\kappa\colon \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_u}$. This gives standard MPC limited robustness to model uncertainty [52–54]. However, standard MPC exhibits a substantial decline in performance for significant plant-model mismatch due to uncertainty.

2.4 Stability theory

This section presents basic definitions of the Lyapunov stability theory that is used to analyze the stability properties of nonlinear systems. Depending on whether model uncertainty is present, either nominal or robust stability analysis must be performed.

2.4.1 Nominal stability

The following definitions are relevant to the nominal stability analysis of discrete-time nonlinear systems.

Definition 1. (Comparison functions [55]) A function $\alpha\colon \mathbb{R}_+ \mapsto \mathbb{R}_+$ is said to be of class $\mathcal{K}$ if $\alpha(0) = 0$, $\alpha(n) > 0$ for all $n > 0$ and is strictly increasing. It becomes of class $\mathcal{K}_\infty$ if in addition, α is unbounded. A function $\beta\colon \mathbb{R}_+ \times \mathbb{Z}_+ \mapsto \mathbb{R}_+$ is of class $\mathcal{KL}$ if $\beta(\,\cdot\,, k)$ is of class $\mathcal{K}$ for each fixed $k \geq 0$, and

$\beta(n,\,\cdot\,)$ is decreasing for each fixed $n \geq 0$ and $\beta(n,k) \to 0$ as $k \to \infty$.

Definition 2. (Attractivity) The system (2.2) is attractive on $\mathbb{X}$ if $\lim_{k\to\infty} x_k = 0$ for all $x_0 \in \mathbb{X}$.

Definition 3. (Stable equilibrium point) The system (2.2) is asymptotically stable on $\mathbb{X}$ if $\lim_{k\to\infty} x_k = 0$ for all $x_0 \in \mathbb{X}$ and $x = 0$ is a stable equilibrium point.

Definition 4. (Asymptotic stability) The system (2.2) is attractive on $\mathbb{X}$ if $\lim_{k\to\infty} x_k = 0$ for all $x_0 \in \mathbb{X}$.

Definition 5. (Positive invariant set) A closed set $\Gamma \subseteq \mathbb{R}^n$ is a positive invariant set for an autonomous system $x^+ = F(x)$ if $x \in \Gamma$ implies $F(x) \in \Gamma$.

Definition 6. (Control positive invariant set) The set $\Gamma \subseteq \mathbb{R}^n$ is a control positive invariant set for a system $x^+ = f(x,u)$ and for each $x \in \Gamma$ there exists a control input $u \in \mathcal{U}$ such that $f(x,u) \in \Gamma$.

Definition 7. (Local Lyapunov function) A function $V\colon \mathbb{R}^n \mapsto \mathbb{R}_+$ is a local Lyapunov function for an autonomous system $x^+ = F(x)$ if there exists a positive invariant set Γ, a feedback control law κ_N, $\mathcal{K}_\infty$ functions α_1, α_2 and $\mathcal{K}$ function α_3 for all $x \in \Gamma$ such that

$$\alpha_1(|x|) \leq V(x) \leq \alpha_2(|x|) \tag{2.4a}$$

$$V(x^+) - V(x) \leq -\alpha_3(|x|) \tag{2.4b}$$

Nominal stability analysis is only suitable for deterministic systems. For a system affected by uncertainty, a robust stability analysis must be performed.

2.4.2 Robust stability

The following are relevant definitions in robust stability analysis of discrete-time nonlinear systems.

Definition 8. (Robust positively invariant (RPI) set, [56]) A set $\Gamma \subseteq \mathbb{R}^n$ is a robust positively invariant for an autonomous system $x^+ = F(x,d)$ if

$F(x,d) \in \Gamma$ holds for all $x \in \Gamma$ and for all $d \in \Omega$, where Ω is the uncertain parameter (disturbance) set.

Definition 9. (Robust control invariant set, [57]) A set $\Gamma \subseteq \mathbb{R}^n$ is a robust control invariant set for a system $x^+ = f(x,u,d)$ if for all $x \in \Gamma$, there exists an admissible control input $u \in \mathcal{U}$ such that $f(x,u,d) \in \Gamma$ for all $d \in \Omega$, where Ω is the uncertain parameter (disturbance) set.

Definition 10. (Regional input-to-state practical stability (ISpS), [58]) An autonomous system $x^+ = F(x,d)$ is ISpS in $\Gamma \subseteq \mathbb{R}^n$ and $0 \in \Gamma$ if there exists a $\mathcal{KL}$ function β, a $\mathcal{K}$ function γ, and a constant $c \geq 0$ such that for all $k \geq 0$:

$$|x_k| \leq \beta(|x_0|, k) + \gamma(\|\mathbf{d}_{[0,k-1]} - \mathbf{d}^0\|) + c, \quad \forall\, x_0 \in \Gamma \tag{2.5}$$

where $\mathbf{d}_{[0,k-1]} := \begin{bmatrix} d_0, & d_1, & \dots, & d_{k-1} \end{bmatrix}$ is the sequence of true parameter realizations and $\mathbf{d}^0$ is a sequence of nominal parameters of the same length.

Definition 11. (ISpS Lyapunov function in Γ, [57]) Suppose Γ is an RPI set and there exists a compact set $\Theta \subseteq \Gamma$ with the origin as an interior point. A function $V\colon \mathbb{R}^n \mapsto \mathbb{R}_+$ is an ISpS Lyapunov function in Γ for the system $x^+ = f(x,d)$, if there exist $\mathcal{K}_\infty$ functions α_1, α_2, α_3, $\mathcal{K}$ function σ, and constants $c_1, c_2 \geq 0$ such that:

$$V(x) \geq \alpha_1(|x|), \quad \forall\, x \in \Gamma \tag{2.6a}$$

$$V(x) \leq \alpha_2(|x|) + c_1, \quad \forall\, x \in \Theta \tag{2.6b}$$

$$\begin{gathered} V(x^+) - V(x) \leq -\alpha_3(|x|) + \sigma(|d - d^0|) + c_2, \\ \forall\, x \in \Gamma, \forall\, d, d^0 \in \Omega. \end{gathered} \tag{2.6c}$$

2.5 Nonlinear programming properties

Parametric nonlinear programming (NLP) properties are presented because the variants of the MPC problem are nonlinear parametric problems, being parametric in the current state x_k and the model uncertain parameters, d^0 for nominal MPC or d^c for multi-stage MPC. Let us consider a general parametric NLP form:

$$\begin{aligned} \min_{w} \quad & J(w,p) && \text{(2.7a)} \\ \text{s.t.} \quad & h(w,p) = 0, && \text{(2.7b)} \\ & g(w,p) \leq 0 && \text{(2.7c)} \end{aligned}$$

where $w \in \mathbb{R}^{n_w}$ is the vector of all optimization variables and $p \in \mathbb{R}^{n_p}$ is the vector of all the parameters in the NLP. The problem (2.7) has a scalar objective function $J\colon \mathbb{R}^{n_w+n_p} \mapsto \mathbb{R}$, equality constraints $h\colon \mathbb{R}^{n_w+n_p} \mapsto \mathbb{R}^{n_e}$, and inequality constraints $g\colon \mathbb{R}^{n_w+n_p} \mapsto \mathbb{R}^{n_i}$.

Definition 12. (Lagrange function and KKT conditions) The Lagrange function of (2.7) is:

$$\mathcal{L}(w,\lambda,\mu,p) = J(w,p) + \lambda^\top h(w,p) + \mu^\top g(w,p) \tag{2.8}$$

where, λ and μ are the Lagrange multipliers of appropriate dimension. A point that satisfies the *Karush-Kuhn-Tucker* (KKT) conditions is known as a KKT-point [59]. Given a parameter p, a point $w^\star$ is called a KKT-point if there exists some multipliers (λ,μ) that satisfy:

$$\begin{aligned} \nabla_w \mathcal{L}(w^\star,\lambda^\star,\mu^\star,p) &= 0 \\ h(w^\star,p) &= 0 \\ g(w^\star,p) &\leq 0 \\ 0 \leq \mu^\star \perp g(w^\star,p) &\leq 0 \end{aligned} \tag{2.9}$$

The set of all multipliers λ and μ that satisfy the KKT conditions for a certain parameter p is denoted as $\mathcal{M}(p)$. The active constraint set is given by $\mathcal{A}(w^\star) = \{j \mid g_j(w^\star,p) = 0\}$. $\perp$ is the complementarity operator, i.e. either μ or g (or both) must be zero.

Then we define strict complementarity as follows.

Definition 13. (SC, [59]) At the KKT point $w^\star$ of (2.7) with multipliers ($\lambda^\star$, $\mu^\star$), strict complementarity (SC) condition holds if $\mu_j + g_j(w^\star,p) > 0$ for all $j \in \mathcal{A}(w^\star)$.

For the KKT conditions to be first-order necessary optimality conditions, the KKT point $w^\star$ requires a constraint qualification to be satisfied. Further, it requires a constraint qualification to be a local minimizer of (2.7). The following are the definitions of three well-known constraint qualifications.

Definition 14. (LICQ, [59]) The linear independence constraint qualification (LICQ) holds at $w^\star$ when the gradient vectors

$$\nabla h(w^\star, p) \quad \text{and} \quad \nabla g_j(w^\star, p),\ j \in \mathcal{A}(w^\star) \tag{2.10}$$

are linearly independent. LICQ implies that the multipliers $\lambda^\star$, $\mu^\star$ are unique i.e. $\mathcal{M}(p)$ is a singleton.

Definition 15. (MFCQ, [59]) The Mangasarian-Fromovitz constraint qualification (MFCQ) holds at $w^\star$ if and only if, $\nabla h(w^\star, p)$ has full column rank (linearly independent), and there exists a non-zero vector q, such that:

$$\begin{aligned} \nabla h(w^\star, p)^\top q &= 0, \\ \nabla g_j(w^\star, p)^\top q &< 0,\ \forall j \in \mathcal{A}(w^\star) \end{aligned} \tag{2.11}$$

MFCQ implies that $\mathcal{M}(p)$ is a compact convex polytope [60].

Definition 16. (CRCQ, [59]) For problem (2.7), the constant rank constraint qualification (CRCQ) holds at $w^\star(p)$, when for any subset $\bar{\mathcal{A}} \subseteq \mathcal{A}(w^\star)$, the gradients:

$$\begin{aligned} \nabla h(w^\star, p)^\top q &= 0, \\ \nabla g_j(w^\star, p)^\top q &< 0,\ \forall j \in \bar{\mathcal{A}} \end{aligned} \tag{2.12}$$

retain constant rank near the point $w^\star(p)$.

The KKT conditions are necessary but are not sufficient conditions for optimality. A second-order condition is needed to guarantee a minimizer. Here are the definitions of the two second-order sufficient conditions.

Definition 17. (SSOSC, [61]) The strong second-order sufficient condition (SSOSC) holds at $w^\star$ with multipliers $\lambda^\star$ and $\mu^\star$ if:

$$q^\top \nabla^2_{ww}\mathcal{L}(w^\star, \lambda^\star, \mu^\star, p)q > 0 \quad \forall q \neq 0 \tag{2.13}$$

such that

$$\begin{aligned} \nabla h_i(w^\star, p)^\top q = 0, &\quad \forall i = 1, \ldots, n_e \\ \nabla g_j(w^\star, p)^\top q = 0, &\quad \forall j \in \mathcal{A}(w^\star) \cap \{j \mid \mu_j > 0\} \end{aligned} \tag{2.14}$$

Definition 18. (GSSOSC, [62]) The general strong second-order sufficient condition (GSSOSC) holds at $w^\star(p)$ if:

$$q^\top \nabla^2_{ww}\mathcal{L}(w^\star, \lambda^\star, \mu^\star, p)q > 0 \quad \forall q \neq 0 \tag{2.15}$$

such that

$$\nabla h(w^\star, p)^\top q = 0, \quad \text{and} \quad \nabla g(w^\star, p)^\top q = 0 \tag{2.16}$$

holds for all $\lambda^\star, \mu^\star \in \mathcal{M}(p)$.

2.5.1 Parametric nonlinear programming sensitivity

The sensitivity of the primal-dual solution $s^\star = [w^{\star\top}, \lambda^{\star\top}, \mu^{\star\top}]^\top$ shows how it changes with respect to a perturbation in the parameter p, from $p = p_0$ to $p = p$. Assuming SC, LICQ, and SSOSC and applying the Implicit Function Theorem (IFT) to the KKT conditions in (2.9) we obtain Theorem 1.

Theorem 1. *(IFT applied to KKT conditions) Assume a KKT point $w^\star(p_0)$ that satisfies (2.9), and that SC, LICQ and SSOSC hold at $w^\star(p_0)$. Furthermore, the functions F, h, and g are at least $k+1$ times differentiable in w and k times differentiable in p. Then the primal-dual solution $s^\star(p)^\top = [w^\star(p)^\top, \lambda^\star(p)^\top, \mu^\star(p)^\top]$ has the following properties:*

- *$s^\star(p_0)$ is an isolated local minimizer of (2.7) at p_0 and contains unique multipliers $(\lambda^\star(p_0),\ \mu^\star(p_0))$.*
- *For p in a neighborhood of p_0 the active constraint set $\mathcal{A}(w^\star)$ remains unchanged.*

- *For p in a neighborhood of p_0 there exists a unique, continuous and differentiable function $s^\star(p)$ which is a local minimizer satisfying SSOSC and LICQ for* (2.7) *at p.*

- *There exists positive Lipschitz constants L_s, L_v such that $|s^\star(p) - s^\star(p_0)| \leq L_s|p - p_0|$ and $|J(p) - J(p_0)| \leq L_v|p - p_0|$, where $|\cdot|$ is the Euclidean norm.*

Proof. see Fiacco [61]. □

Theorem 1 above establishes Lipschitz continuity of the optimal solution and optimal objective function of (2.7) with respect to the NLP parameters. This result is important for the sensitivity-based approximation schemes of the multi-stage NMPC that have been proposed in this thesis, and their stability analysis. The theorem ensures that we can always obtain from (2.9) the following linear system for NLP sensitivity if SC, LICQ, and SSOSC hold at a KKT point $w^\star(p)$,

$$\begin{bmatrix} \nabla^2_{ww}\mathcal{L} & \nabla h & \nabla g_{\mathcal{A}} \\ \nabla h^\top & 0 & 0 \\ \nabla g_{\mathcal{A}}^\top & 0 & 0 \end{bmatrix} \begin{bmatrix} \nabla_p w^\star \\ \nabla_p \lambda^\star \\ \nabla_p \mu^\star_{\mathcal{A}} \end{bmatrix} = \begin{bmatrix} \nabla^2_{wp}\mathcal{L} \\ \nabla h^\top \\ \nabla g_{\mathcal{A}}^\top \end{bmatrix} \tag{2.17}$$

where all the derivatives of are evaluated at $w^\star$, p_0, $(\lambda^\star, \mu^\star) \in \mathcal{M}(p_0)$, and $g_{\mathcal{A}}$ and $\mu^\star_{\mathcal{A}}$ are vectors with elements g_j and $\mu^\star_j$ for all $j \in \mathcal{A}(w^\star)$, respectively.

2.5.2 Nonlinear program solution approximation

The Taylor expansion of the primal solution at p_0 is written as:

$$w^\star(p) = w^\star(p_0) + \nabla_p {w^\star}^\top (p - p_0) + \mathcal{O}(|p - p_0|^2) \tag{2.18}$$

As a result, a first-order approximation is obtained in the neighborhood of p_0 using:

$$w^\star(p) \approx w^\star(p_0) + \nabla_p {w^\star}^\top (p - p_0) \tag{2.19}$$

The KKT matrix in (2.17) is evaluated at p_0 and is obtained for free from the previous solution. The benefit of the sensitivity step (2.19) is to perform one-step-ahead predictions of the solution at the subsequent MPC iteration with a minimal computational effort compared to solving the NLP from scratch. The following subsection presents the horizon update algorithm for adaptive horizon multi-stage MPC using the sensitivity update.

The weakest conditions that ensure a perturbed solution of (2.7) is locally unique are MFCQ and GSSOSC [63]. Further, Lipschitz continuity of $w^\star(p)$ with respect to p is guaranteed.

3 | Robust Multi-stage MPC

"Some see the glass half full, some see it half empty, and some see it crawling with toxic alien parasites who want to devour your pancreas."

James Alan Gardner (1955-present)

This chapter provides a brief history leading up to the development of robust MPC approaches. It then describes the multi-stage MPC and the robust horizon assumption which is the conventional approximation strategy to reduce the multi-stage MPC problem size. Finally, it introduces several multi-stage MPC formulations, which are essential for comprehending the novel techniques presented in later chapters.

3.1 Model predictive control under uncertainty

When the system model and measurements are inaccurate the performance of the standard MPC degrades. Although standard MPC has some inherent robustness against uncertainty, this property may break when there are significant disturbances that cause problem infeasibility. As a result, MPC approaches have been developed to account for the influence of uncertainty and can be broadly categorized as follows:

1. *MPC reformulation* with softened constraints and a violation penalty,
2. *Stochastic MPC* with chance constraints that allow violations only at a prespecified probability, and
3. *Robust MPC* where constraint satisfaction is guaranteed for a set of expected uncertain parameter realizations.

3.1.1 MPC reformulation to avoid infeasibility

The state constraints in standard MPC may result in problem infeasibility when there is uncertainty. To ensure that the MPC problem is always feasible, a reformulation of the MPC problem is done by adding softened state constraints and a penalty for the slack variables in the objective function. The reformulated standard MPC problem is written as:

$$\begin{aligned} V_N^{\mathrm{rf}}(x_k, \mathbf{d}^0) := \min_{z_i, \nu_i, \varepsilon_i} \ & \psi(z_N, d^0) + \sum_{i=0}^{N-1} \ell(z_i, \nu_i, d^0) \\ & + \rho_\psi e^\top \varepsilon_N + \sum_{i=0}^{N-1} \rho_\ell e^\top \varepsilon_i \end{aligned} \tag{3.1a}$$

$$\text{s.t.} \quad z_{i+1} = f(z_i, \nu_i, d^0), \quad i = 0, \ldots, N-1 \tag{3.1b}$$

$$z_0 = x_k, \tag{3.1c}$$

$$x^L - \varepsilon_i \leq z_i \leq x^U + \varepsilon_i, \tag{3.1d}$$

$$x_f^L - \varepsilon_N \leq z_N \leq x_f^U + \varepsilon_N, \tag{3.1e}$$

$$\nu_i \in \mathbb{U}, \ \varepsilon_i \geq 0, \ \varepsilon_N \geq 0$$

where V_N^{rf} is the optimal value function of the relaxed formulation, $\varepsilon_i^c, \varepsilon_N^c \in \mathbb{R}^{n_x}$, are the slack variables to the corresponding state variables, ρ_ψ^c and ρ_ℓ^c are large weights on penalty terms, and $e^\top = [1, \ldots, 1]$. The state constraints in (2.3) are replaced by bound inequalities to easily depict the constraint relaxations. Further, this type of constraint is the most common in process applications, to keep a specific variable within its operational limits.

The presence of slack variables in the inequalities (3.1d) and (3.1e) guarantees the existence of strictly feasible directions into the interior of the feasible region. In that case, the relaxed problem (3.1) guarantees that MFCQ always holds at the optimal solution. The optimal value function V_N^{rf} is uniformly continuous in its argument x_k.

3.1.2 Stochastic MPC

Stochastic MPC is an optimal control approach that systematically handles system uncertainties by transcribing them into a stochastic program. Although stochastic MPC is not the focus of this thesis, it is briefly presented because multi-stage MPC also applies stochastic programming principles.

Stochastic MPC considers that real-world systems have probabilistic uncertainties. When it is possible to characterize stochastic system uncertainties, they should be accounted for in the controller framework. Stochastic MPC uses probabilistic uncertainty definitions to describe chance constraints, which specify in advance that state and output constraints are to be satisfied to a minimum probability level. Therefore, it provides a systematic way of establishing the tradeoffs between fulfilling control objectives and guaranteeing probabilistic constraint satisfaction [64].

The following is a general stochastic MPC problem formulation with joint chance constraints on states.

$$V_N^{\mathrm{smpc}}(x_k) := \min_{z_i, \nu_i} \mathbb{E}_{x_k}\Big[\psi(z_N, d_i) + \sum_{i=0}^{N-1} \ell(z_i, \nu_i, d_i)\Big] \tag{3.2a}$$

$$\text{s.t. } z_{i+1} = f(z_i, \nu_i, d_i), \quad i = 0, \ldots, N-1 \tag{3.2b}$$

$$z_0 = x_k, \tag{3.2c}$$

$$\nu_i \in \mathbb{U}, \quad i = 0, \dots, N-1 \tag{3.2d}$$

$$\mathbf{Pr}_{x_k}[g_j(z_i, \nu_i, d_i) \leq 0, \, \forall \, j = 1, \dots, n_g] \geq \beta, \, i = 1, \dots, N \tag{3.2e}$$

$$d_i \sim \mathbf{P}_d, \quad i = 0, \dots, N-1 \tag{3.2f}$$

where $\mathbf{P}_d$ is the (multivariate) probability distribution of the unknown disturbances and measurement noise d and $\beta \in (0, 1)$ is the lower bound for the probability that the inequality constraint $g_j \leq 0$ must be satisfied. The conditional probability $\mathbf{Pr}_{x_k}$ in (3.2e) shows that the probability that $g_j \leq 0$ for all $j = 1, \dots, n_g$ through the prediction horizon holds and it is dependent on the initial state x_k. $V_N^{\text{smpc}}(x_k)$ is the optimal cost function. The optimal cost function for stochastic MPC is denoted by $V_N^{\text{smpc}}(x_k)$, commonly defined as the conditional expectation of the sum of stage costs and terminal cost.

Most of the stochastic MPC literature focuses on stochastic linear systems. The development of stochastic nonlinear MPC has been limited due to the computational complexities of uncertainty propagation in nonlinear systems [65]. This thesis focuses on multi-stage MPC for nonlinear systems because it has a more computationally feasible uncertainty propagation strategy. Multi-stage MPC also derives its objective function from stochastic programming principles as in stochastic MPC.

3.1.3 Robust MPC strategies

Two major ways have been proposed to achieve robust constraint satisfaction for MPC. The first is the direct way where a contraction in the reachable set is included in the optimization problem, to guarantee constraint satisfaction. This direct strategy is adopted by the tube-based MPC first proposed by Mayne et al. [66] for linear systems. In this approach, a nominal controller predicts a nominal trajectory and an ancillary controller is used to keep the uncertain system within a tube centered on the nominal trajectory. While this tube-based control method can ensure robustness, it is often excessively conservative. Additionally, calculating the ancillary control law for nonlinear systems can be challenging, restricting the practical usefulness of tube-based robust NMPC.

The second approach is the indirect way which does not explicitly include a reachable set contraction in the problem formulation but is implied in its solution. The min-max MPC approach is a robust MPC strategy that uses an indirect way to compute the optimal control trajectory, aiming to minimize the cost of the worst-case scenario [67]. However, this framework overlooks future recourse actions that could counteract the uncertainty, resulting in overly conservative or infeasible outcomes. A feedback min-max MPC was introduced by Scokaert and Mayne [2], addressing this issue by seeking closed-loop optimization over different control policies for various uncertainty realizations, leading to a lower degree of conservativeness and avoiding infeasibility. This approach uses a scenario tree to describe the evolution of uncertainty, incorporating aspects of stochastic programming. The scenario tree framework was extended to nonlinear systems by Lucia et al. [68], leading to the development of the robust multi-stage NMPC, which is the primary focus of this thesis. The next section presents the robust multi-stage MPC and its formulation in further detail.

3.2 Multi-stage MPC

Multi-stage MPC is a robust MPC approach that models the evolution of the uncertainty of the system along the prediction horizon as a tree of discrete scenarios, known as a *scenario tree* [2, 6, 67]. This approach is based on stochastic programming principles and is known as multi-stage stochastic programming in decision theory and finance [69]. Fig. 3.1 is an illustration of a scenario tree that is fully branched to the end of the prediction horizon N, with a selection of three uncertain parameter realizations. The total number of scenarios in a fully branched scenario tree is given by $(N_D)^N$, where N_D is the number of discrete parameter realizations selected. This thesis generalizes that each branch in the scenario tree represents a selected uncertain parameter realization regardless of the selection strategy. However, it is a common heuristic in multi-stage MPC literature to select the realizations such that three levels, usually $\{\text{min, nominal, max}\}$, are considered for each uncertainty. Then, the number of selected discrete parameter realizations becomes $N_D = (n_d)^{n_{\text{levels}}}$ where n_{levels} is the number of levels picked from

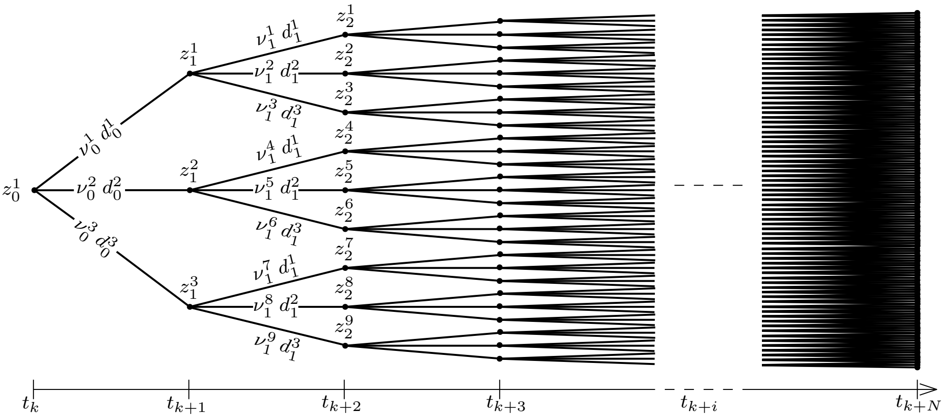

Figure 3.1: A fully branched scenario tree with three selected uncertain parameter realizations, $N_D = 3$, and prediction horizon, N. The dashed lines represent all the nodes in between t_{k+3} and t_{k+N}.

each uncertainty. For this specific case, the total number of scenarios in a fully branched scenario tree is given by $({n_d}^{n_{\text{levels}}})^N$.

Each branch from a node represents the effect of the unknown disturbances or modeling errors, and the selected control inputs. Multi-stage MPC uses the tree structure to explicitly account for uncertainties in future control inputs. This structure considers that new information will be available in the future that can inform new decisions. The approach allows for adjusting the control inputs to counteract the effects of uncertainties, resulting in a closed-loop robust MPC with lower conservative performance than other robust MPC approaches with no recourse, such as the open-loop min-max MPC, multi-model or multi-scenario approaches. Scokaert and Mayne [2] presented the feedback min-max MPC where the scenario tree was first applied for MPC. Then de la Peña et al. [70] and Bernardini and Bemporad [71] followed with more results for its application in linear MPC. In his doctoral thesis, Lucia [3] presents the application of the scenario tree to handle uncertainties for nonlinear systems.

It is worth noting that the tree structure does not necessarily indicate time-varying uncertainties or disturbances, but rather signifies that if the uncertainty is not identified at any given sampling time, it will remain unknown in the subsequent sampling time when a new tree (shifted forward in time) will need to be taken into account.

3.3 Robust horizon assumption

When the prediction horizon is long, or when there are many branches (many parameter realizations selected), the number of scenarios in the scenario tree increases exponentially resulting in an intractable problem. This is a major challenge in the implementation of multi-stage MPC. A simple strategy to reduce the multi-stage MPC problem size has been proposed, both by de la Peña et al. [72] for linear systems, and Lucia et al. [6] for the nonlinear case. They suggest limiting the branching of the tree only up to a certain stage to stop the rapid growth of the scenario tree. The stage at which the branching of the scenario tree stops is called the *robust horizon*. This simplification stems from the fact that MPC proceeds in a receding horizon fashion, thus modeling the uncertainties in the far future is not very critical because the control inputs will be recomputed at the next MPC iteration anyway. In addition, a recent study by Shin et al. [73] supports this claim by showing that the effect of a perturbation of the uncertain parameter on the current optimal input decays exponentially to its distance away from the root node.

Again, assume that the number of discrete parameter realizations selected is $N_D = 3$. Fig. 3.2 shows a scenario tree with a robust horizon $N_R = 2$ and a total of nine scenarios, where branching stops at time t_{k+N_R} and the uncertain parameter realizations are kept constant until t_{k+N}. Generally, the total number of scenarios is given by $\mathrm{card}(\mathbb{C}) = (N_D)^{N_R}$, where $\mathbb{C}$ is the scenario index set and N_D is the number of discrete parameter realizations selected. The problem size is significantly reduced when $N_R << N$. For example, if $N = 6$, and there are three uncertainties $n_d = 3$, each selected at three levels $n_{\text{levels}} = 3$, such that $N_D = 3^3 = 27$, then choosing a robust horizon $N_R = 2$ reduces the number of scenarios from $387,420,489$ to 729.

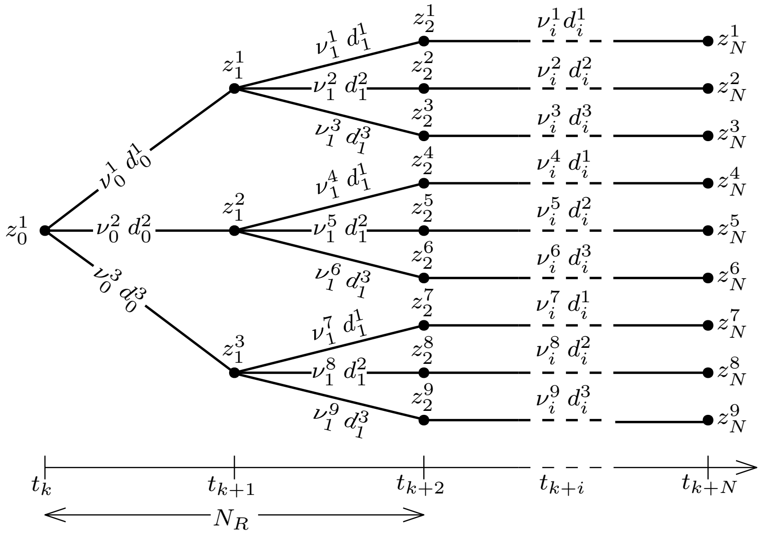

Figure 3.2: A scenario tree with nine scenarios, prediction horizon N, and robust horizon $N_R = 2$, provided that number of selected uncertain parameter realizations is $N_D = 3$. Generally, the number of scenarios is $(N_D)^{N_R}$. The dashed lines represent all the nodes in between t_{k+2} and t_{k+N}.

3.4 Multi-stage MPC formulation

Given the current state x_k at time t_k, multi-stage MPC with a robust horizon N_R requires solving the following problem:

$$V_N^{\mathrm{ms}}(x_k, \mathbf{d}^c) = \min_{z_i^c, \nu_i^c} \sum_{c \in \mathbb{C}} \omega_c \Big(\psi(z_N^c, d_{N-1}^c) + \sum_{i=0}^{N-1} \ell(z_i^c, \nu_i^c, d_i^c) \Big) \tag{3.3a}$$

$$\text{s.t. } z_{i+1}^c = f(z_i^c, \nu_i^c, d_i^c),\ i = 0, \ldots, N-1 \tag{3.3b}$$

$$z_0^c = x_k, \tag{3.3c}$$

$$\nu_i^c = \nu_i^{c'},\ \{(c, c') \mid z_i^c = z_i^{c'}\} \tag{3.3d}$$

$$d_{i-1}^c = d_i^c,\ i \geq N_R \tag{3.3e}$$

$$z_i^c \in \mathbb{X},\ \nu_i^c \in \mathbb{U},\ d_i^c \in \mathbb{D}, \tag{3.3f}$$

$$z_N^c \in \mathbb{X}_f^c, \tag{3.3g}$$

$$\forall\, c, c' \in \mathbb{C}$$

where z_i^c, ν_i^c, and d_i^c represent the state, input, and disturbance vectors at stage i and scenario c. The set $\mathbb{C}$ is the scenario index set, and the nominal scenario has a zero index. The objective function (3.3a) is the weighted sum of the stage and terminal costs across all the scenarios with ω_c being the probability for each scenario.

Non-anticipativity constraints (NACs) ensuring that inputs from a common parent node in the scenario tree (Fig. 3.2) must be equal are given by (3.3d). This is because it is only possible to make a single control action $u_k = \nu_0^\star$ on the plant. (3.3e) is included to show that parameter realizations are constant after the robust horizon. The problem formulation for a fully branched scenario tree is recovered by setting $N_R = N$. The state and control inputs inequality constraints across all scenarios are given in (3.3f) and $\mathbb{D}$ is the set of the discrete parameter realizations.

Problem (3.3) is a parametric optimization problem, with the parameters being x_k and $\mathbf{d}^c$. The parameter $\mathbf{d}^c$ denotes $\mathbf{d}_{i=0,\dots,N-1}^{c=1,\dots,\text{card}(\mathbb{C})}$ which is the concatenated vector of all the parameter realizations in the scenario tree (across all scenarios and across all time steps in the prediction horizon).

The robust horizon assumption makes solving the multi-stage MPC problem practically feasible. However, this can still be expensive, especially for nonlinear problems, leading to a significant computational delay. To reduce the computational costs and computational delay of the multi-stage NMPC even further, Thombre et al. [20] proposed the sensitivity-assisted multi-stage NMPC (SAMNMPC). This approximation framework has an algorithm to prune irrelevant scenarios from the scenario tree using NLP sensitivities and speed up computations. The sensitivity-assisted multi-stage NMPC is discussed in detail in Chapter 4.

4 | Sensitivity-Assisted Multi-stage NMPC

"Scientists investigate that which already is;
Engineers create that which has never been."

Albert Einstein (1879-1955)

In this chapter, the sensitivity-assisted multi-stage NMPC [20] is briefly presented. It is a sensitivity-based approach to improve the computational efficiency of the multi-stage NMPC. In Part II of this thesis, this computationally efficient scenario selection approach is extended by combining it with multivariate statistical analysis of process data to reduce its degree of conservativeness.

4.1 Background

It is now evident that multi-stage MPC faces a computational challenge due to its problem size growing rapidly with the scenario tree. The number of variables in the multi-stage MPC formulation increases exponentially with respect to the following:

- the number of uncertain parameters,
- the number of discrete realizations in each realization parameter, and
- the prediction horizon length.

The large-sized NLP is too computationally expensive to solve, resulting in a computational delay that is problematic for its real-time implementation. To improve the computational efficiency, it is necessary to consider approximate solutions for the multi-stage NMPC.

The heuristic for multi-stage NMPC is to pick the {max, nominal, min} parameter values as the realizations represented in the scenario tree branches. Out of these realizations, the worst-case realization is the one that is most likely to cause constraint violations [19]. Critical scenarios for each inequality constraint are assembled from a combination of these worst-case realizations at each time step. The critical scenarios are those with predicted trajectories that are most likely to violate constraints. They are determined by using sensitivities of the constraints with respect to the uncertain parameters. Apart from the nominal scenario, the remaining scenarios are the non-critical scenarios.

The algorithm for the sensitivity-assisted multi-stage NMPC (SAMNMPC) is based on efficiently categorizing scenarios into two:

- a small number of critical scenarios, and
- a larger number of noncritical scenarios.

Then, a smaller optimization problem is assembled using variables and constraints that correspond only to the nominal scenario and critical scenarios. An approximate contribution of the noncritical scenarios based on their sensitivities is added to the cost function to account for the effect of the noncritical scenarios.

In real-time implementation, these parameter values may result from estimation and measurements. The nominal is the mean estimate, and the maximum and minimum values are identified based on the uncertainty in the estimator or measurement. Measurement and estimation delays must be carefully considered in deciding a suitable sampling rate. Therefore, at a certain sampling time, these identified parameter realization values are used to formulate the subsequent multi-stage MPC problem. A standard MPC is solved with the nominal parameters and then sensitivities with respect to the model parameters are obtained. These are then used to determine the critical scenarios based on whether constraint violations will occur in their predicted trajectories. Then finally the multi-stage MPC problem is formulated with only the critical scenario constraints and variables which will be fewer, achieving reduced computational effort.

4.2 Problem formulation

The SAMNMPC problem at time t_k from the current state x_k, is written as follows:

$$
\begin{aligned}
V_N^{\text{sam}}(x_k, \mathbf{d}^c) = \min_{\substack{z_i^c, \nu_i^c \\ c \in \widehat{\mathbb{C}} \cup \{0\}}} \; & \sum_{c \in \widehat{\mathbb{C}} \cup \{0\}} \omega_c \Big(\psi(z_N^c, d_{N-1}^c) + \sum_{i=0}^{N-1} \ell(z_i^c, \nu_i^c, d_i^c) \Big) + \\
& \sum_{c \in \bar{\mathbb{C}}} \omega_c \Big(\psi(z_N^0 + \Delta z_N^c, d_{N-1}^c) + \sum_{i=0}^{N-1} \ell(z_i^0 + \Delta z_i^c, \nu_i^0 + \Delta \nu_i^c, d_i^c) \Big)
\end{aligned} \tag{4.1a}
$$

$$
\begin{aligned}
\text{s.t. } & z_{i+1}^c = f(z_i^c, \nu_i^c, d_i^c), \quad i = 0, \dots, N-1 && (4.1\text{b}) \\
& z_0^c = x_k, && (4.1\text{c}) \\
& \nu_i^c = \nu_i^{c'}, \quad \{(c, c') \mid z_i^c = z_i^{c'}\} && (4.1\text{d}) \\
& d_{i-1}^c = d_i^c, \quad i \geq N_R && (4.1\text{e}) \\
& z_i^c \in \mathbb{X},\ \nu_i^c \in \mathbb{U},\ z_N^c \in \mathbb{X}_f,\ d_i^c \in \mathbb{D}, && (4.1\text{f}) \\
& \forall\, c, c' \in \widehat{\mathbb{C}} \cup \{0\}
\end{aligned}
$$

where the sets $\widehat{\mathbb{C}}$ and $\bar{\mathbb{C}}$ are the critical and noncritical scenario index sets, respectively, and $\{0\}$ represents the nominal scenario. The set $\mathbb{D} \in \mathbb{R}^{n_d}$ is the uncertain parameter set containing a finite number of realizations, $\mathbb{X} \in \mathbb{R}^{n_x}$, $\mathbb{U} \in \mathbb{R}^{n_u}$ are the feasible sets for states and inputs, respectively, and $\mathbb{X}_f$ represents the terminal region. N is the prediction horizon length and N_R is the robust horizon. z_i^c and ν_i^c are the predicted state and control variable vectors for scenario c at time t_{k+i}, respectively. The stage cost function is given by ℓ, the terminal cost is denoted by ψ, and ω_c represents the weights on scenario c to the objective function. As shown in (4.1a), the noncritical scenarios are approximated with their NLP sensitivity steps Δz_i^c and $\Delta \nu_i^c$. The variables and constraints in problem (4.1) are only those associated with critical scenarios, thus making the problem smaller than that of the multi-stage NMPC with a robust horizon.

4.3 Critical scenario selection

This section presents how critical scenarios are determined in a multi-stage nonlinear MPC problem. The critical scenario set includes the worst-case parameter realizations that are most likely to violate the inequalities (usually state bounds).

Let $\mathbf{g}(z_i, \nu_i, d_i) \leq 0$ represent the inequality constraints in the problem (4.1) at time t_k such that $\mathbf{g}\colon \mathbb{R}^{n_x \times n_u \times n_d} \mapsto \mathbb{R}^{n_g}$. Considering each inequality is indexed as $g_j(\,\cdot\,,\,\cdot\,,\,\cdot\,) \leq 0$, the critical scenarios are found by solving the following optimization problem at time t_k with a fixed control sequence ν_i for all $i = 1, \ldots, N$ and $i' = 0, \ldots, i-1$:

$$\max_{d_{i'}} \quad g_j(z_i, \nu_i, d_i) \tag{4.2a}$$

$$\text{s.t.} \quad z_{\bar{i}+1} = f(z_{\bar{i}}, \nu_{\bar{i}}, d_{\bar{i}}), \quad \bar{i} = i', \ldots, i-1 \tag{4.2b}$$

$$z_0 = x_k \tag{4.2c}$$

Problem (4.2) is solved around a reference trajectory $(z_i, v_i)|_{\text{ref}}$, $i = 1, \ldots, N$

and has the following implicit relationship:

$$\begin{aligned} g_j(z_i, \nu_i, d_i) &= g_j(z_i(d_{i'}), \nu_i, d_i) \\ &= g_j(d_{i'}, d_i) \leq 0, \quad i' + 1 \leq i = 1, \dots, N \end{aligned} \tag{4.3}$$

where the reference trajectory can be any feasible sequence of states and control inputs. In SAMNMPC, the reference trajectory is the standard (nominal) MPC solution.

Assuming that g_j is strictly monotonic with respect to d_i, the solution of (4.2) can be found by linearization of the system model f, and the inequalities g, around the reference (nominal) trajectory. Now concatenate the uncertain parameters and states, and define $\mathbf{d}^\top = [d_0^\top,\ d_1^\top, \dots,\ d_{N_R}^\top]$ and $\mathbf{z}^\top = [z_0^\top,\ z_1^\top, \dots,\ z_N^\top]$. To obtain the sensitivities $\frac{d\mathbf{g}}{d\mathbf{d}} \in \mathbb{R}^{n_g \times n_d \cdot N_R}$ around the reference trajectory we can then write:

$$\frac{d\mathbf{g}}{d\mathbf{d}}^\top = \nabla_\mathbf{z}\mathbf{g}^\top \left(\frac{d\mathbf{z}}{d\mathbf{d}}\right)^\top + \nabla_\mathbf{d}\mathbf{g}^\top \tag{4.4}$$

Note that if the inequality constraints represent the state bounds $z_i \in \mathbb{X}$, then $\nabla_\mathbf{d}\mathbf{g} = 0$, and $\nabla_\mathbf{z}\mathbf{g} = \pm 1$, depending on whether it is an upper or lower bound.

To compute the sensitivity $\frac{d\mathbf{z}}{d\mathbf{d}}$, let $\mathbf{h}(z_i, d_i) = 0$ represent the equality constraints in (4.2), with a fixed ν_i. Differentiating the equality constraints using IFT gives:

$$\begin{aligned} \nabla_\mathbf{z}\mathbf{h}^\top \cdot d\mathbf{z} + \nabla_\mathbf{d}\mathbf{h}^\top \cdot d\mathbf{d} = 0 \\ \frac{d\mathbf{z}}{d\mathbf{d}}^\top = -(\nabla_\mathbf{z}\mathbf{h}^{-\top})\nabla_\mathbf{d}\mathbf{h}^\top \end{aligned} \tag{4.5}$$

Substituting (4.5) in (4.4) leads to:

$$\frac{d\mathbf{g}}{d\mathbf{d}}^\top = -\nabla_\mathbf{z}\mathbf{g}^\top(\nabla_\mathbf{z}\mathbf{h}^{-\top})\nabla_\mathbf{d}\mathbf{h}^\top + \nabla_\mathbf{d}\mathbf{g}^\top \tag{4.6}$$

In this approach, the solution of the standard MPC problem at time t_k is selected as the reference trajectory. By doing this, the terms $\nabla_z\mathbf{h}$ and $\nabla_d\mathbf{h}$

are obtained from the Jacobian matrix at the optimal solution derived from solving the standard MPC problem. This enables efficient computation of the critical scenarios, even for longer robust horizons, as $\nabla_d \mathbf{h}$ can be easily obtained by parameterizing the standard MPC problem based on the uncertain parameters $(\mathbf{d}_i)_{i=1,\ldots,N_R}$.

Additionally, the assumption that the worst-case realization for the uncertain parameter $(d_m)_{m=1,\ldots,n_d}$ lies at either its maximum value $d_m^{\max}$ or its minimum value $d_m^{\min}$ is made. By combining this assumption with strict monotonicity of g, the analytical solution for (4.2) for $i \in N_R$ and $m = 1, \ldots, n_d$ can be explicitly stated as:

$$\begin{aligned} d_{i,m}^{\mathrm{wc}} &= \arg\max_{d_i \in \mathbb{D}} \quad \frac{d\mathbf{g}}{d\mathbf{d}}^{\top} \mathbf{d} \\ &= \begin{cases} d_{i,m}^{\min}, & \text{if } \frac{d(g_j)}{d(d_{i,m})}|_{(z_i,\nu_i)|_{\mathrm{ref}}} \leq 0 \\ d_{i,m}^{\max}, & \text{otherwise} \end{cases} \end{aligned} \tag{4.7}$$

Only in the particular case of strictly monotonic systems guarantees that the worst-case realizations lie on the extremes of the parameters. Therefore the critical scenarios are easily obtained from (4.7). Moreover, a critical scenario can be ignored if an element $|\frac{d(g_j)}{d(d_{i,m})}| \leq \epsilon$ implying that the constraint is insensitive to a change in the uncertain parameter $d_{l,m}$. This means that the number of active inequality constraints is always an upper bound for the number of critical scenarios, which is usually much smaller than the full scenario tree.

4.4 SAMNMPC algorithm

The overall strategy of SAMNMPC proposed by Thombre et al. [20] is outlined in Algorithm 1. This method incorporates the same type of feedback information and its influence on the controller as the multi-stage NMPC. This is achieved by including the predicted state and control trajectories for the critical scenarios, which represent the worst-case uncertainty realiz-

[1] Notation from [20]. $\mathbf{K}_c$ represents the KKT matrix for a scenario $c \in \mathbb{C}$ where $\mathbf{K}_0$ is for the nominal scenario.

Algorithm 1 Sensitivity-assisted multi-stage NMPC [20]

```
Given: {max, nominal, min} of all uncertain parameters;
for $k = 1 \to \infty$ do
    Get the current state of the plant $x_k$;
    Solve the standard NMPC problem (2.3) for the nominal uncertainty.; $d_k^0$, and get the KKT matrix $\mathbf{K}_0$[1] at the optimal solution;
    For the critical scenarios: Extract $\nabla_z \mathbf{h}$ and $\nabla_d \mathbf{h}$ from $\mathbf{K}_0$, and solve (4.6) to form the critical scenario set $\hat{\mathbb{C}}$;
    For noncritical scenarios: Solve the linear system (29) in [20] (derived from (2.17)), with the approximation $\mathbf{K}_c = \mathbf{K}_0 \; \forall c \in \mathbb{C}$, and get the sensitivity steps for the noncritical scenarios $c \in \bar{\mathbb{C}}$;
    Solve the SAMNMPC formulation (4.1), where the constraints are imposed for critical scenarios and the noncritical scenarios are approximated with their sensitivity steps in the objective function;
    Set $u_k = \nu_0^c$, $c \in \hat{\mathbb{C}} \cup \{0\}$ and inject into the plant.
end for
```

ations. Additionally, sensitivity approximations for the predicted state and control trajectories are included for all other scenarios. Consequently, all scenarios considered in multi-stage NMPC are also considered in SAMNMPC. The solution obtained from SAMNMPC differs from the multi-stage MPC solution by a term of $\mathcal{O}(|\Delta \mathbf{d}|)$, and the closed-loop stability and recursive feasibility properties have been established in [20].

Part I

Adaptive Horizon Multi-stage MPC

5 | Adaptive Horizon Multi-stage MPC

"Intelligence is the ability to adapt to change."

STEPHEN HAWKING (1942-2018)

This chapter presents how a fast online horizon update algorithm can be implemented for multi-stage MPC to improve its computational efficiency. The proposed control algorithm that includes horizon adaptation is implemented on two numerical examples to demonstrate its control performance.

The results in this chapter are based on and adapted from the following research papers.

- A peer-reviewed conference article: "Adaptive horizon multi-stage nonlinear model predictive control" [21], and
- A part of the subsequent journal article: "Stability properties of the adaptive horizon multi-stage MPC" [22].

5.1 Motivation

In the introductory Section 1.4, it is established that there is a growing interest in nonlinear MPC but its implementation is hindered by the numerical complexity of solving a nonlinear program (NLP), and the associated large computational delay [74]. The delay is further increased when uncertainty is explicitly considered, in robust MPC approaches, such as in multi-stage MPC formulations. For multi-stage MPC, where multiple scenarios are considered, the problem of computational delay becomes more severe because the problem size increases exponentially with an increase in the prediction horizon. This implies that the computational time, and hence the computational delay, both increase exponentially with increasing size of the scenario tree. We have presented the robust horizon [6, 72] as a means to decrease the number of variables. However, an additional reduction may be required when there is a substantial number of parameter realizations.

One approach proposed to speed up MPC from the literature is the adaptive horizon MPC. The method aims to provide an online horizon update for subsequent MPC iterations. It shortens the prediction horizon as the controlled system approaches a stabilizing region. The MPC problem size is reduced, leading to shorter solution times while preserving the MPC stability properties. Krener [75, 76] proposed using Lyapunov function properties to detect stabilization before shortening the horizon. This method requires the determination of a good Lyapunov function, which can be difficult for nonlinear systems. Griffith et al. [77] employs linearization around the operating point to establish a stabilizing region where the horizon can be truncated.

Despite their successful implementation in standard MPC, the adaptive horizon techniques have not yet been employed in multi-stage MPC. Multistage MPC has significantly larger solution times than standard MPC because of the explicit consideration of multiple scenarios in its optimization problem. Hence, the adaptive horizon strategy is even more beneficial for multi-stage MPC, particularly for nonlinear systems.

This chapter presents the work done by Mdoe et al. [21], extending the adaptive horizon scheme by Griffith et al. [77] for multi-stage MPC. The approach requires performing one-step-ahead predictions, terminal region calculations, and selection of the sufficient (stabilizing) prediction horizon length. Although this was studied for standard MPC by [77], the extension of these ideas to multi-stage MPC where we have multiple scenarios is not trivial, because:

- The determination of terminal ingredients depends on the nature of the scenario tree, and
- The horizon update using parametric NLP sensitivity needs to ensure that the one-step-ahead approximate solution for each scenario reaches its stabilizing terminal region.

This chapter also provides a proof of concept of the proposed algorithm using two numerical case studies with nonlinear complexity. Later in Chapter 6, a study that is based on the subsequent article [22] is presented to extend these results by providing a robust stability analysis of the resulting controller based on Input-to-State practical Stability (ISpS).

5.2 Adaptive horizon multi-stage MPC

Adaptive horizon algorithms provide an update for the prediction horizon length of the MPC problem. The prediction horizon is not an optimization variable in these approaches, resulting in faster horizon updates compared to solving a mixed-integer NLP (MINLP) in variable horizon MPC [78]. The adaptive horizon MPC has been implemented by [77] for reference tracking, and then extended for economic MPC by [79]. This chapter aims to extend the algorithm by [77] to multi-stage MPC with a reference tracking objective. The algorithm is expected to have the capability to achieve a high level of robustness by considering multiple scenarios in the optimization problem while maintaining fast computation times.

5.2.1 Transformation of variables

Setpoint tracking under uncertainty implies tracking different implicit references for each scenario. Therefore, the solution to the multi-stage MPC problem (3.3) has states and control inputs tracking different implicit references for each parameter realization. It is possible to enforce a common zero by reformulating (3.3) with transformed variables. But before that, let us define the following properties:

Definition 19. (Mapping scenarios to parameter realizations) There exists a function $r\colon \mathbb{C} \mapsto \mathbb{Q}$ mapping any scenario c to its parameter realization index at its leaf node, where $\mathbb{Q}$ is the set of parameter realization indices.

Definition 20. (Stable and optimal equilibrium pairs for each parameter realization) For any given parameter realization $d^r \in \mathbb{D}$, a state-input pair $(x^r, u^r) \in \mathbb{X} \times \mathbb{U}$ is a *stable equilibrium pair* for system (2.1) if $x^r = f(x^r, u^r, d^r)$ holds. Further, if it yields the lowest stage cost among all equilibrium points, then it is the *optimal equilibrium pair* (x_f^r, u_f^r).

Then let us make the following assumption to ensure that there exists a solution to the steady state optimization problem of (2.1) for each parameter realization $d^r \in \mathbb{D}$.

Assumption 1. There exists an optimal equilibrium pair (x_f^r, u_f^r) for each parameter realization $d^r \in \mathbb{D}$. The optimal equilibrium pairs for each parameter realization are the implicit references for the corresponding scenarios.

To enforce a common zero to all the scenarios, reformulate (3.3) using the following:

$$\overline{z}_i^c = z_i^c - x_f^r, \quad \overline{\nu}_i^c = \nu_i^c - u_f^r \tag{5.1}$$

for all $c \in \mathbb{C}$, and for all $r \in \mathbb{Q}$. The transformed system model becomes:

$$\overline{z}_{i+1}^c = \overline{f}(\overline{z}_i^c, \overline{\nu}_i^c, d_i^c) = f(\overline{z}_i^c + x_f^r, \overline{\nu}_i^c + u_f^r, d_i^c) - x_f^r \tag{5.2}$$

such that $\overline{z}_i^c \in \overline{\mathbb{X}}$, $\overline{\nu}_i^c \in \overline{\mathbb{U}}$, where $\overline{\mathbb{X}}$ and $\overline{\mathbb{U}}$ are the corresponding new feasible sets of the transformed system. Equations (5.1) and (5.2) imply when $(\overline{z}_i^c, \overline{\nu}_i^c) = (0,0)$ we have $x_f^r = f(x_f^r, u_f^r, d_i^c)$ for all $c \in \mathbb{C}$, and for all $r \in \mathbb{Q}$. We define transformed stage costs as:

$$\overline{\ell}(\overline{z}_i^c, \overline{\nu}_i^c, d_i^c) = \ell(\overline{z}_i^c + x_f^r, \overline{\nu}_i^c + u_f^r, d_i^c) - \ell(x_f^r, u_f^r, d_i^c) \tag{5.3}$$

and transformed terminal costs as:

$$\overline{\psi}(\overline{z}_N^c, d_{N-1}^c) = \psi(\overline{z}_N^c + x_f^r, d_{N-1}^c) - \psi(x_f^r, d_{N-1}^c) \tag{5.4}$$

such that $\overline{\psi}(0, d_{N-1}^c) = \overline{\ell}(0,0,d_i^c) = 0$ for all $c \in \mathbb{C}$.

The problem (3.3) is transformed using Eqs. (5.1) to (5.4) to obtain a common terminal region containing zero. This is a necessary property for the closed-loop stability analysis which is presented in Chapter 6.

To keep the notation simple, we will continue using the original notation. However, from here on, all variables and functions in equation (3.3) are assumed to be transformed.

Now that the desired system properties have been defined, the next subsection discusses the methodology to compute the terminal ingredients for each parameter realization in the multi-stage MPC.

5.2.2 Terminal ingredients for the adaptive horizon multi-stage MPC

The adaptive horizon multi-stage MPC requires the following:

- a terminal cost function ψ, and
- a positive invariant terminal region $\mathbb{X}_f$.

The terminal cost is the weighted average of the terminal costs across all parameter realizations. For each parameter realization, a terminal cost function is computed beforehand, based on a linearized system about its optimal equilibrium pair [80].

The terminal region is assumed to be the region where a terminal control law on the nonlinear system has a negligible error. However, the determination of the terminal control law depends on whether the scenario tree is fully branched or has a robust horizon.

Terminal ingredients for multi-stage MPC with a robust horizon

First, consider when the scenario tree has a robust horizon i.e. $N_R < N$. The scenario tree has no NACs at the leaf nodes leading to decoupled terminal control input variables across all the scenarios, ν^c_{N-1}. Different stabilizing control laws may exist inside the terminal region for each parameter realization. Then the terminal regions for each parameter realization are approximated independently. We use an infinite horizon LQR for the control law in the terminal region and the terminal cost.

Assumption 2. Given a parameter realization $d^r \in \mathbb{D}$ for all $r \in \mathbb{Q}$, there exists a stabilizing LQR with a local control law $h^r_f(x) := u^r_f - K_r x$ such that $f(x, h^r_f(x), d^r) \in \mathbb{X}^r_f$ for all $x \in \mathbb{X}^r_f$.

Assumption 2 considers that different LQR controllers may exist in the terminal region across the different scenarios.

For a given parameter realization d^r, linearize (2.1) about (x^r_f, u^r_f) and write the resulting system as a sum of the linear and nonlinear terms as follows:

$$x_{k+1} = A_{K_r} x_k + \phi(x_k, h^r_f(x_k), d^r) \tag{5.5}$$

where $A_{K_r} = A_r - B_r K_r$, $A_r = \nabla^\top_x f(0, 0, d^r)$ and $B_r = \nabla^\top_u f(0, 0, d^r)$ and (A_r, B_r) is stabilizable, $\phi\colon \mathbb{X} \times \mathbb{U} \times \mathbb{D} \mapsto \mathbb{X}$ is the nonlinear part of the dynamics. The infinite horizon LQR for the linearized system with the parameter realization d^r is:

$$\begin{aligned} \psi(x_k, d^r) = x_k^\top P_r x_k = \min_{\nu_i} \sum_{l=0}^{\infty} (z_i^\top Q z_i + \nu_i^\top R \nu_i) \\ \text{s.t. } z_0 = x_k,\ z_{i+1} = A_r z_i + B_r \nu_i, \quad \forall\, i = 0,\ \ldots,\ \infty \end{aligned} \tag{5.6}$$

where $Q \succ 0$, $R \succ 0$ are tuning matrices. $P_r \succ 0$ is the Ricatti matrix for d^r.

To determine the terminal region for the LQR on the nonlinear system, there must exist an upper bound on the nonlinear effects for any parameter realization given by Lemma 2.

Lemma 2. *For any parameter realization $d^r \in \mathbb{D}$, there exists M_r, $q_r \in \mathbb{R}_+$ such that $|\phi(x, d^r)| \leq M_r |x|^{q_r}$, $\forall\, x \in \mathbb{X}$ where $\phi(x, d^r) := \phi(x, h_f^r(x), d^r)$.*

Proof. Consider a single d^r and then follow the proof of Theorem 14 in [77]. □

Analytical determination of the linearization error and its bound is tedious or impossible. The bounds are determined offline by one-step simulations instead. One-step simulations from random initial states are used to obtain the linearization error numerically by evaluating for each d^r, the difference $\phi(x, d^r) = f(x, h_f^r(x), d^r) - A_{K_r} x$, $\forall\, x \in \mathbb{X}$.

Assumption 3. For a given d^r, there exists a terminal region of attraction $\mathbb{X}_f^r$ around (x_f^r, u_f^r) such that $\mathbb{X}_f^r := \{z \mid |z - x_f^r| \leq c_f^r\}$ where c_f^r is the terminal radius corresponding to d^r.

Assumption 3 means that the terminal region set for each parameter realization is approximated by the interior of an n-sphere with radius c_f^r. After fitting M_r and q_r for all parameter realizations according to Lemma 2, and Assumption 3 holds, the terminal region radii are computed using Lemma 3.

Lemma 3. *Suppose Assumption 3 holds, the terminal radii c_f^r for all $d^r \in \mathbb{D}$ depend on their corresponding linearization error bounds which are given by:*

$$c_f^r := \left(\frac{-\bar{\sigma}_r \Lambda_r + \sqrt{(\bar{\sigma}_r \Lambda_r)^2 + (\underline{\lambda}_{W_r} - \epsilon_{LQ})\Lambda_r)}}{\Lambda_r M_r} \right)^{\frac{1}{q_r - 1}} \tag{5.7}$$

where $\bar{\sigma}_r$ is the maximum singular value of A_{K_r}, $\bar{\lambda}_{W_r}$ and $\underline{\lambda}_{W_r}$ are the maximum and minimum eigenvalues of $W_r := Q + {K_r}^\top R K_r$, $\Lambda_r := \frac{\bar{\lambda}_{W_r}}{(1-\bar{\sigma}_r)^2}$, and $\epsilon_{LQ} > 0$ is a small constant: an allowable tolerance for the terminal cost $\psi(x, d^r)$.

Proof. As in Lemma 2, consider a single d^r and use the proof from [77]. □

Terminal ingredients for multi-stage MPC with a fully branched scenario tree

Let us consider the case when the scenario tree is fully branched i.e. $N_R = N$. The stabilizing control law for each parameter realization inside the terminal region cannot be determined independently due to the NACs at the leaf node. Therefore, a common terminal control law must exist (i.e. $h_f^r(x) = \kappa_f(x)$, for all r).

After linearizing (2.1) as outlined in the previous subsection, the following steps are used to determine the terminal ingredients for the adaptive horizon multi-stage MPC with a fully branched scenario tree:

1. Select matrices Q, and R, for the LQR and choose a K such that the common terminal control law $\kappa_f(x) = -Kx$ is stabilizing for all parameter realizations r, and $-Kx \in \mathbb{U}$, $\forall x \in \mathbb{X}$,
2. Solve for P_r in the discrete Lyapunov equation $A_r^\top P_r A_r + Q = P_r^\top$, for each parameter realization r,
3. Use Q, R, A_{K_r} to obtain the terminal radii for each parameter realization r by evaluating (5.7) in Lemma 3 where $A_{K_r} = A_r - B_r K$. Then pick the smallest as the common terminal region's radius,
4. Obtain the terminal cost function as the weighted sum $\psi(x) = \sum_r x^{r\top} P_r x^r$.

Steps 3 and 4 above are similar procedures to those in the previous subsection, only that K is common for all realizations r.

5.2.3 Problem formulation

Now that the terminal ingredients for each parameter realization have been defined, we present the formulation for the adaptive horizon multi-stage MPC problem. The formulation is obtained by replacing the fixed horizon N in (3.3) with a variable horizon N_k as follows:

$$V_{N_k}^{\text{ahm}}(x_k, \mathbf{d}^c) = \min_{z_i^c, \nu_i^c} \sum_{c \in \mathbb{C}} \omega_c \Big(\psi(z_{N_k}^c, d_{N_k-1}^c) + \sum_{i=0}^{N_k-1} \ell(z_i^c, \nu_i^c, d_i^c) \Big) \tag{5.8a}$$

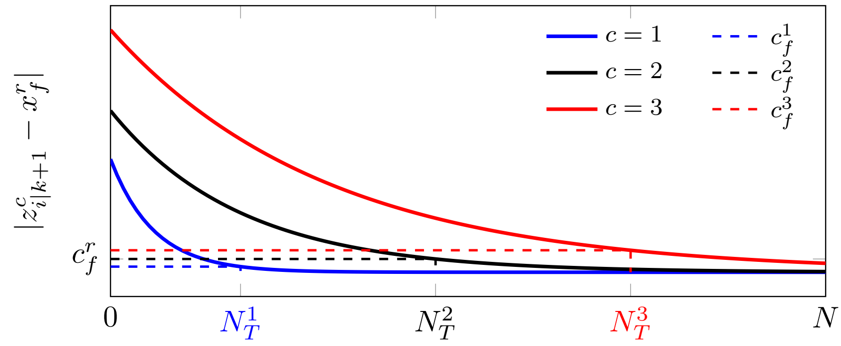

Figure 5.1: Sensitivity prediction (solid lines) of (5.8) for three scenarios at t_{k+1}, and the terminal regions (below dashed lines). N_{k+1} will be reduced from N to at least N_T^3.

$$\text{s.t. } z_{i+1}^c = f(z_i^c, \nu_i^c, d_i^c), \quad i = 0, \ldots, N_k - 1 \tag{5.8b}$$

$$z_0^c = x_k, \tag{5.8c}$$

$$\nu_i^c = \nu_i^{c'}, \quad \{(c, c') \mid z_i^c = z_i^{c'}\} \tag{5.8d}$$

$$d_{i-1}^c = d_i^c, \quad i \geq N_R \tag{5.8e}$$

$$z_i^c \in \mathbb{X},\ \nu_i^c \in \mathbb{U},\ d_i^c \in \mathbb{D}, \tag{5.8f}$$

$$z_N^c \in \mathbb{X}_f^c, \tag{5.8g}$$

$$\forall\, c, c' \in \mathbb{C}$$

where $V_{N_k}^{\text{ahm}}$ is the optimal cost function, and N_k is the prediction horizon of the current MPC iteration given by a horizon update algorithm. The horizon update algorithm requires one-step-ahead predictions obtained from parametric NLP sensitivities that are discussed in the next subsection.

5.2.4 Horizon update algorithm

The algorithm begins by selecting $N_{\max}$, a safety factor $N_{\min} > N_R$, and computing the terminal ingredients offline. The value $N_{\max}$ is chosen as a sufficiently long horizon that guarantees the feasibility of (5.8) and is used as an initialization for N. In each MPC iteration, the algorithm determines a horizon update such that all scenarios reach their terminal regions irrespective of the initial state in the subsequent MPC iteration. Fig. 5.1

is a simple illustration of how a new horizon length is determined for three scenarios. The one-step-ahead prediction of the solution to (5.8) from a sensitivity update is in solid lines and the terminal regions for each scenario are below the dashed lines. Each scenario reaches its respective terminal region at stage N_T^c. To ensure that each scenario reaches its terminal region at t_{k+1}, the new prediction horizon must be at least equal to the largest N_T^c. Therefore in Fig. 5.1 N_{k+1} must be greater than N_T^3.

Algorithm 2 presents the horizon update algorithm for multi-stage MPC.

Algorithm 2 Horizon update for multi-stage MPC

```
 1: Define $N_{\max}$, $N_{\min} > N_R$
 2: Determine $\mathbb{X}_f^r$, $P_r$ for all $d^r \in \mathbb{D}$
 3: Initialize: $k \leftarrow 0$, $N_0 \leftarrow N_{\max}$
 4: $w^\star(x_k) \leftarrow$ Solve (5.8) with $N_k$ and initial state $x_k$
 5: $\mathcal{X}_{k+1} \leftarrow \{x_{k+1}^r \,|\, x_{k+1}^r = f(x_k, u_k, d^r)\, \forall d^r \in \mathbb{D}\}$
 6: $\nabla_{x_k} w^\star \leftarrow$ Evaluate (2.17) to obtain NLP sensitivities at $x_k$
 7: $\mathcal{W}_{k+1}^\star \leftarrow \{\text{get } w^\star(x_{k+1}^r) \text{ using (2.19) } \forall x_{k+1}^r \in \mathcal{X}_{k+1}\}$
 8: for all $w^\star(x_{k+1}^r) \in W_{k+1}^\star$ do
 9:     if $z_{N_k|k+1}^c \in \mathbb{X}_f^r \; \forall c \in \mathbb{C}$ then[1]
10:         $N_T^c \leftarrow$ step at which $\mathbb{X}_f^r$ is reached
11:         $N_T(x_{k+1}^r) \leftarrow \max_{c \in \mathbb{C}} N_T^c$
12:     else
13:         $N_T(x_{k+1}^r) \leftarrow N_{\max}$
14:     end if
15: end for
16: $N_T \leftarrow \max_{x_{k+1}^r \in \mathcal{X}_{k+1}} N_T(x_{k+1}^r)$
17: $N_{k+1} \leftarrow \min(N_{\max}, N_T + N_{\min})$
18: $k \leftarrow k + 1$
19: Go to Step 4
```

The flowchart in Fig. 5.2 summarizes Algorithm 2 which is a mapping of the current state and prediction horizon to the subsequent prediction horizon. The mapping is defined as a function in Definition 21 as follows:

Definition 21. (Horizon update function) Let $\mathcal{N}$ be the universal set of admissible horizon lengths such that $\mathcal{N} = \{N \,|\, N_{\min} \leq N \leq N_{\max},\, N \in \mathbb{Z}_+\}$.

[1] $z_{N_k|k+1}^c$ is extracted from the perturbed solution vector $w^\star(x_{k+1}^r)$

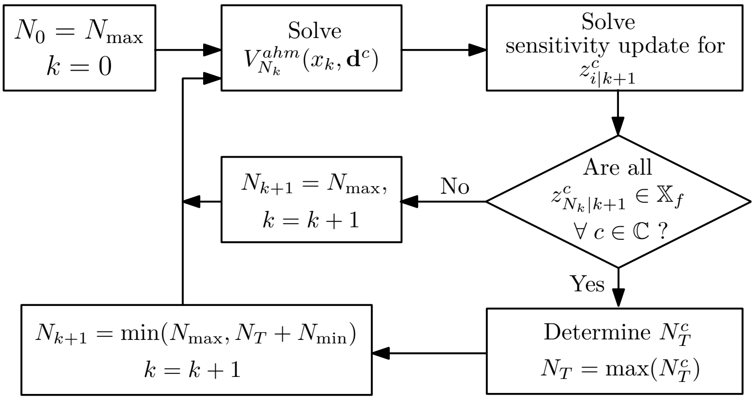

Figure 5.2: Horizon update algorithm for multi-stage MPC that gives admissible prediction horizons for subsequent MPC iterations.

Further, there exists a subset of horizon lengths $\mathcal{N}_k \subset \mathcal{N}$ that define the feasible problems (5.8) at time t_k. We define a function $H\colon \mathbb{R}^{n_x} \times \mathcal{N} \times \mathbb{R}^{n_x} \mapsto \mathcal{N}$ that determines the horizon lengths for the adaptive horizon multi-stage algorithm such that $N_{k+1} = H(x_k, N_k, z_{k+1|k}^c) \in \mathcal{N}_{k+1}$.

5.3 Numerical experiments

Two numerical examples presented here consider a robust horizon (i.e. no full branching). The aim is to compare the performances of the adaptive horizon multi-stage MPC (varying horizon lengths) with fixed horizon multi-stage MPC and to demonstrate the computational cost savings as a result of the adaptive horizon algorithm.

5.3.1 Example 1 — Cooled CSTR

Consider the control of a CSTR with a cooling jacket example from [81]. The dynamics of the cooled CSTR are given by,

$$\dot{c_A} = F(c_{A,0} - c_A) - k_1 c_A - k_3 c_A^2$$
$$\dot{c_B} = -F c_B + k_1 c_A - k_2 c_B$$

$$
\begin{aligned}
\dot{T}_R &= F(T_{\text{in}} - T_R) + \frac{k_W A_R}{\rho c_p V_R}(T_J - T_R) \\
&\quad - \frac{k_1 c_A \Delta H_{AB} + k_2 c_B \Delta H_{BC} + k_3 c_A^2 \Delta H_{AD}}{\rho c_p} \\
\dot{T}_J &= \frac{1}{m_J c_{p,J}}(\dot{Q}_J + k_W A_R (T_R - T_J))
\end{aligned}
$$

where reaction rates k_i follow the Arrhenius law, $k_i = A_i \exp\left(\frac{-E_i}{RT_R}\right)$.

The state variable vector $x = [c_A,\, c_B,\, T_R,\, T_J]^\top$ consists of the concentrations of A and B, reactor and coolant temperatures, respectively. The control inputs u, are inlet flow per reactor volume $F = V_{\text{in}}/V_R$, and cooling rate $\dot{Q}_J$.

The control objective is to regulate c_B at a desired setpoint. The system operates at two setpoints: $c_B^{\text{set}} = 0.5\,\text{mol}/\ell$, and $c_B^{\text{set}} = 0.7\,\text{mol}/\ell$. The activation energy $E_3 = 8560R \pm 2.5\%$ K is the only uncertain parameter in the system. Table A.2 summarizes the system bounds and its initial conditions. The stage cost is a setpoint tracking squared error of c_B plus control movement penalization terms $\Delta F_i = F_i - F_{i-1}$ and $\Delta \dot{Q}_{Jk} = \dot{Q}_{Ji} - \dot{Q}_{Ji-1}$ given by $\ell_k = (c_{Bi} - c_B^{\text{set}})^2 + r_{\Delta 1}\Delta F_i^2 + r_{\Delta 2}\Delta \dot{Q}_{Ji}^{2}$. where the control penalties are $r_{\Delta 1} = 10^{-5}$ and $r_{\Delta 2} = 10^{-7}$. Regularization terms are added to impose implicit references on the remaining states and strong convexity [82].

The initial prediction horizon is $N_0 = 40$. Terminal constraints are included for all the scenarios. The simulations are performed for $N_R = 1,\, 2$. Three parameter realizations are sampled (selected), such that when $N_R = 2$ there are 9 scenarios as in Fig. 3.2.

Offline approximation of terminal conditions

To design a stabilizing LQR for each parametric realization, we use $Q = I_4$, and $R = \text{diag}([10^{-3}\, 10^{-4}])$. As outlined before, 10^5 one-step simulations are performed offline from random initial states to compute linearization error for each parametric realization and each operating point. Fig. 5.3 shows the plots of the linearization error against $|\Delta x|$ for each parametric realization and each setpoint. All the plots show an upper bound on the linearization error (red solid lines in Fig. 5.3) from which M_r and q_r values are fitted. The terminal region radii for each parametric realization are obtained by evaluating (5.7) as shown in Table 5.1.

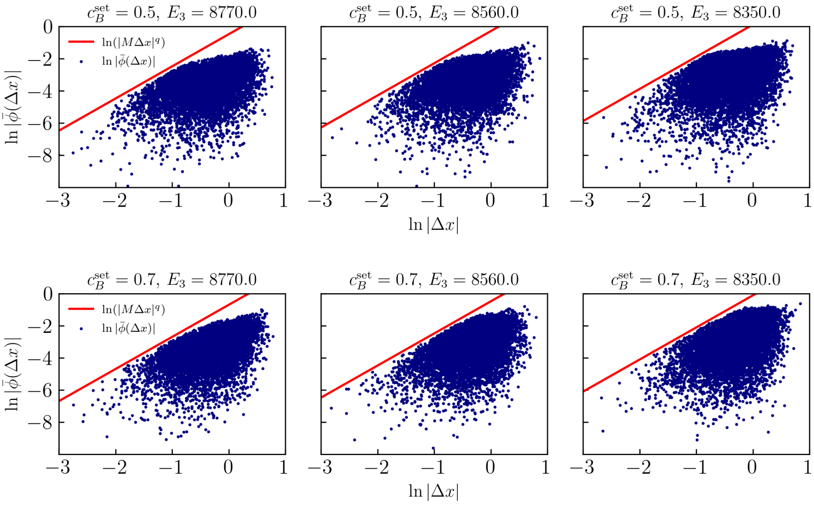

Figure 5.3: CSTR — plots of linearization error against $|\Delta x|$ for 10000 simulations for each parametric realization of E_3, at each setpoint $c_B^{\text{set}} = 0.5$ mol/ℓ and 0.7 mol/ℓ showing their bounds (red lines).

Table 5.1: CSTR — linearization error bounds and terminal radii

	$c_B^{\text{set}} = 0.5$ mol/l			$c_B^{\text{set}} = 0.7$ mol/l		
E_3 (K)	M_r	q_r	c_f^r	M_r	q_r	c_f^r
8774	0.62	2.0	0.1429	0.50	2.0	0.1718
8560	0.75	2.0	0.1159	0.62	2.0	0.1337
8346	1.12	2.0	0.0747	0.90	2.0	0.0846

Simulation results

The simulations are done using `JuMP v.0.21.10` [83] as the NLP modeler in a `Julia` [84] environment. The NLP solver used is `IPOPT 3.13.4` [85], and the linear solver is `HSL-MA57` by the STFC Rutherford Appleton Laboratory, on a 2.6 GHz Intel Core-i7 with 16 GB memory.

A sample time of 18 s is used, and the process is simulated for 0.6 h. At the beginning $c_B^{\text{set}} = 0.5\,\text{mol}/\ell$ and it changes to $c_B^{\text{set}} = 0.7\,\text{mol}/\ell$ at $t_k = 0.3\,\text{h}$. The uncertain parameter E_3 is a random sequence of the sampled realizations, where the value of $E3$ changes at every time step. The control performance of the adaptive horizon multi-stage MPC is compared with the fixed horizon multi-stage MPC in this system. The value of $N_{\text{min}} = 5$ in the adaptive horizon multi-stage algorithm. Figs. 5.4 and 5.5 show closed-loop simulation results when $N_R = 1, 2$, respectively. The optimal input sequences and state trajectories for both controllers are nearly identical. The two controllers show similar performance in tracking setpoint changes c_B^{set}. This implies that the adaptive horizon algorithm does not affect the tracking performance of the multi-stage MPC.

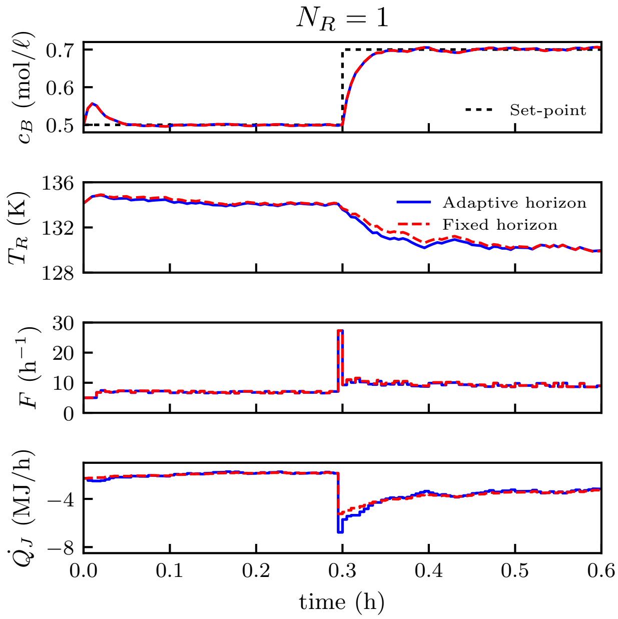

Figure 5.4: CSTR — Simulation results comparing the adaptive horizon multi-stage MPC with the fixed horizon multi-stage MPC when robust horizon $N_R = 1$.

The prediction horizon and total computation times per iteration for the two controllers when $N_R = 1, 2$ are plotted in Figs. 5.6 and 5.7, respectively. It is evident in Figs. 5.6 and 5.7 that the prediction horizon increases slightly

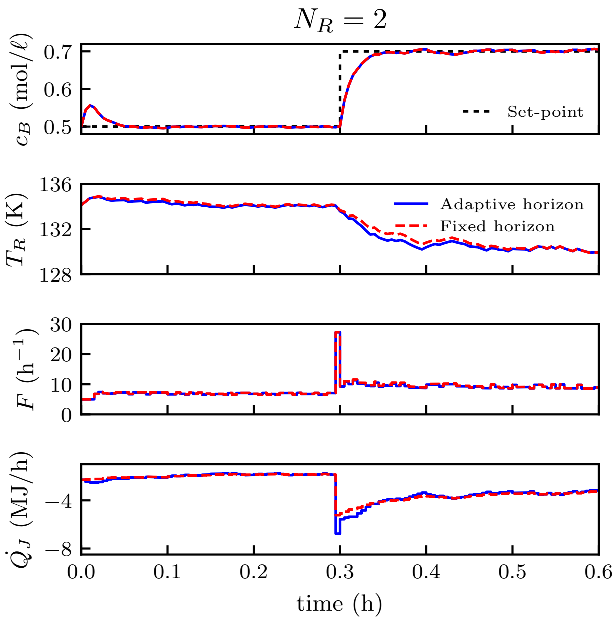

Figure 5.5: CSTR — Simulation results comparing the adaptive horizon multi-stage MPC with the fixed horizon multi-stage MPC when robust horizon $N_R = 2$.

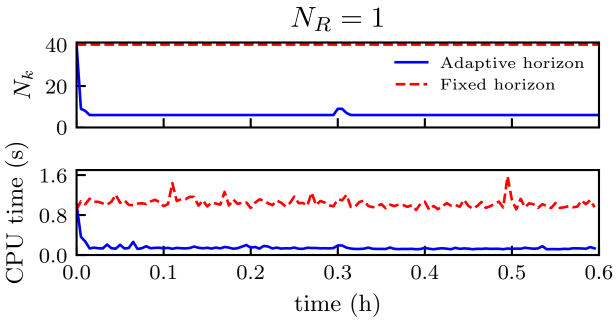

Figure 5.6: CSTR — Simulation results comparing prediction horizons and computation times per iteration when robust horizon $N_R = 1$.

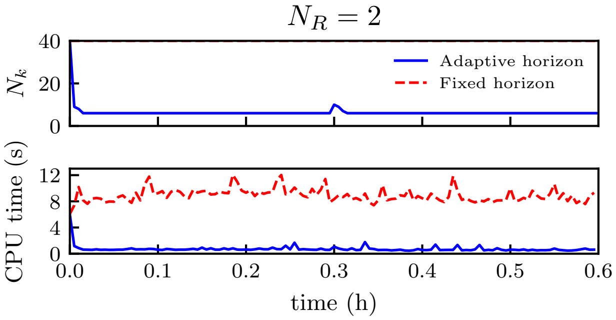

Figure 5.7: CSTR — Simulation results comparing prediction horizons and computation times per iteration when robust horizon $N_R = 2$.

to adapt to the setpoint change at 0.3 hours. This shows that the adaptive horizon algorithm can flexibly handle setpoint changes in operation and automatically schedule prediction horizon lengths.

The adaptive horizon multi-stage MPC reduces the prediction horizon significantly, thus saving the computation time needed in each multi-stage MPC iteration. There is an 86% savings in CPU time on average from fixed horizon multi-stage MPC with $N_R = 1$ by using the adaptive horizon multi-stage MPC. When $N_R = 2$, there is also a 92% savings in CPU time on average. The regulatory performances of the two controllers are nearly identical but the prediction horizon and computation time are reduced significantly for the adaptive horizon multi-stage MPC.

5.3.2 Example 2 — Quad-tank system

The second example is from [87], on the control of a quad-tank system with four tanks whose configuration is illustrated in Fig. 5.8. The liquid levels in the four tanks are described by the following set of differential equations:

$$
\begin{aligned}
\dot{x_1} &= -\frac{a_1}{A_1}\sqrt{2gx_1} + \frac{a_3}{A_1}\sqrt{2gx_3} + \frac{\gamma_1}{A_1}u_1 \\
\dot{x_2} &= -\frac{a_2}{A_2}\sqrt{2gx_2} + \frac{a_4}{A_2}\sqrt{2gx_4} + \frac{\gamma_2}{A_2}u_2 \\
\dot{x_3} &= -\frac{a_3}{A_3}\sqrt{2gx_3} + \frac{1-\gamma_2}{A_3}u_2
\end{aligned}
$$

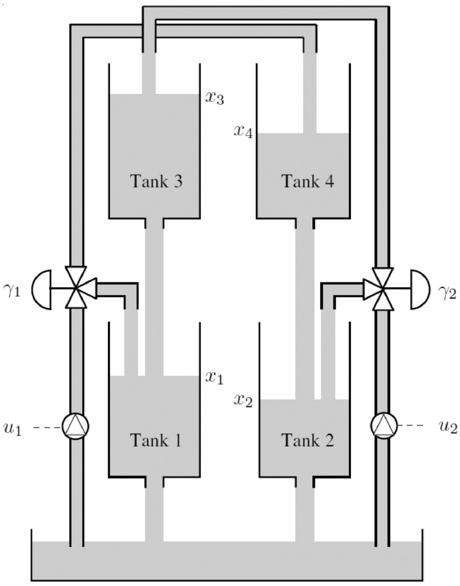

Figure 5.8: Quad-tank system diagram

$$\dot{x_4} = -\frac{a_4}{A_4}\sqrt{2gx_4} + \frac{1-\gamma_1}{A_4}u_1$$

where x_i represents the liquid level in tank i, u_i is the flow rate of pump i. The system has four states: $x = [x_1, x_2, x_3, x_4]^\top$ and two control inputs: $u = [u_1, u_2]^\top$. A_i and a_i are the cross sectional areas of the tank i and its outlet, respectively. The valve coefficients γ_1 and γ_2 are the uncertain parameters of this system. The values of $\gamma_1, \gamma_2 = 0.4 \pm 0.05$ and the system bounds are shown in Table A.4.

The controller has a reference tracking objective, regulating the two levels in the lower tanks (tanks 1 and 2). The setpoints are $x_1^{\text{set}} = x_2^{\text{set}} = 14\,\text{cm}$. We introduce predefined pulse changes in the state values at specific iterations to reset reference tracking as shown in Table A.5.

The closed-loop multi-stage MPC simulations are run for 150 iterations with a sample time of 10 s. The stage cost function is given by: $\ell_k = (x_{1i} - x_1^{\text{set}})^2 + (x_{2i} - x_2^{\text{set}})^2 + r_\Delta(\Delta u_{1i}^2 + \Delta u_{2i}^2)$, where $\Delta u_{1i} = u_{1i} - u_{1i-1}$ and $\Delta u_{2i} = u_{2i} - u_{2i-1}$ are the control movement terms for the pump flow-rates that are penalized in the objective function with the penalty parameter $r_\Delta = 0.01$.

Offline approximation of terminal conditions

LQR controller tuning $Q = 1.5I_4$ and $R = I_2$ are selected for the stabilizing controller for each parameter realization of γ_1 and γ_2. As done previously, 10^5 one-step offline simulations from randomly sampled initial state values are performed for each parameter realization to compute the linearization error. The set $\mathbb{D}$ is obtained via grid-based sampling including the nominal and extreme parameter values only. A total of 9 parameter realizations are considered for the two uncertain parameters. Fig. 5.9 is a matrix of linearization error plots from the one-step simulations with estimated upper bounds (red lines) for each uncertain parameter realization. The terminal radii for each parameter realization are computed using (5.7). The estimated linearization error bounds and terminal radii for each parameter realization are shown in Table 5.2.

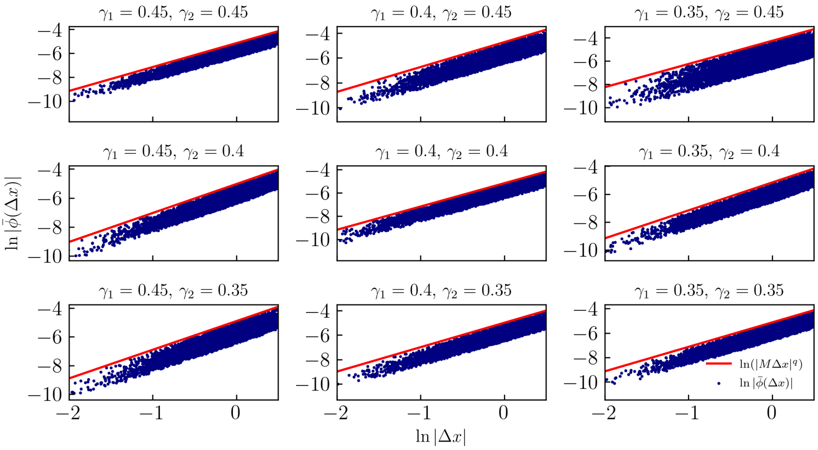

Figure 5.9: Quad-tank — matrix of plots of linearization error against $|\Delta x|$ of $10,000$ simulations for every uncertain parameter realization.

Table 5.2: Quad-tank — linearization error bounds and terminal radii

γ_1	γ_2	M_r	q_r	c_f^r
0.45	0.45	0.0058	2.0	30.43
0.40	0.45	0.0088	2.0	19.99
0.35	0.45	0.0142	2.0	12.12
0.45	0.40	0.0065	2.0	27.04
0.40	0.40	0.0056	2.0	31.77
0.35	0.40	0.0057	2.0	31.57
0.45	0.35	0.0074	2.0	23.69
0.40	0.35	0.0068	2.0	26.63
0.35	0.35	0.0059	2.0	30.97

Simulation results

The simulations are done using `JuMP v.0.21.10` [83] as the NLP modeler in a `Julia` [84] environment. The NLP solver used is `IPOPT 3.13.4` [85], and the linear solver is `HSL-MA57` by the STFC Rutherford Appleton Laboratory, on a 2.6 GHz Intel Core-i7 with 16 GB memory.

The parameters γ_1 and γ_2 have an uncertainty range of ± 0.05 about their nominal value. Similar to Example 2, the simulations are performed with a parameter sequence randomly sampled from {max, nom, min} values. Again, two simulation sets compare the control performances of the adaptive horizon and fixed horizon multi-stage MPC when $N_R = 1, 2$. Simulation results of the quad-tank system for the two controllers when $N_R = 1, 2$ are plotted in Figs. 5.10 and 5.11, respectively. It is seen from the levels x_1, x_2 that the setpoint tracking performance of the two controllers for both cases of $N_R = 1, 2$ are again nearly identical. There are also no significant constraint violations in the bounds of x_3 and x_4 for both controllers. Therefore, there are no performance losses in both reference tracking and robustness of the multi-stage MPC by including the horizon update algorithm. This result, however, must not mislead the reader to think that the choice of robust horizons does not affect performance. In this example, increasing the robust horizon from 1 to 2 led to increased conservativeness for each controller. This is evident with increased accumulated cost as summarized

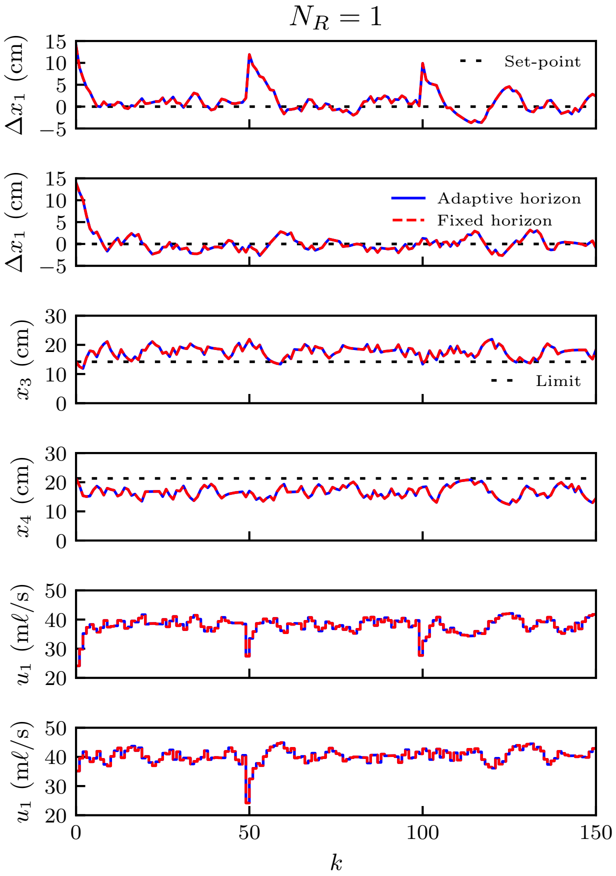

Figure 5.10: Quad-tank — Simulation results comparing fixed horizon multi-stage and adaptive horizon multi-stage MPC with $N_R = 1$.

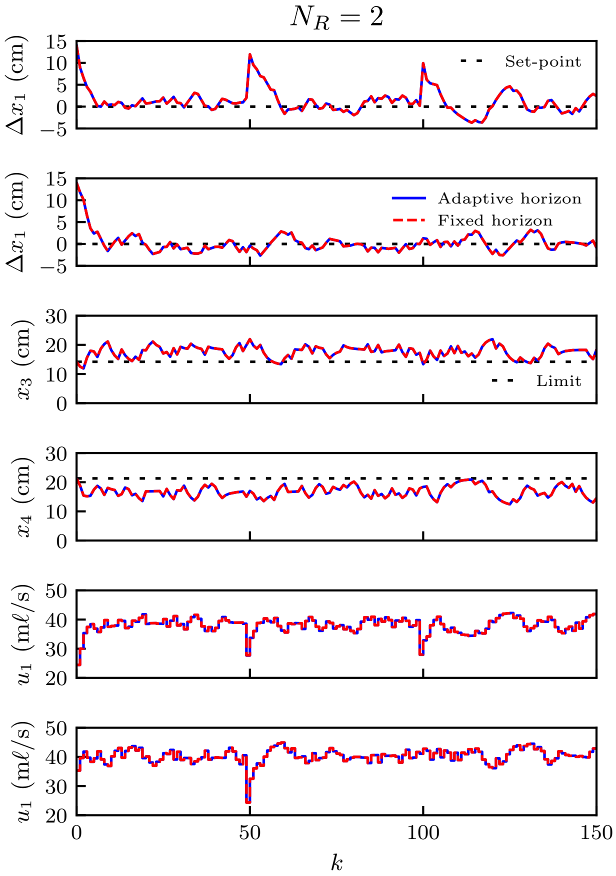

Figure 5.11: Quad-tank — Simulation results comparing fixed horizon multi-stage and adaptive horizon multi-stage MPC with $N_R = 2$.

in Table 5.3.

Table 5.3: Quad-tank — Comparing accumulated costs for each robust horizon length and controller

MSMPC scheme	Robust horizon	
	$N_R = 1$	$N_R = 2$
Adaptive horizon	2161.34	2175.86
Fixed horizon	2161.82	2180.60

The prediction horizon and computation time per iteration for both controllers when $N_R = 1$, 2 are plotted in Figs. 5.12 and 5.13, respectively.

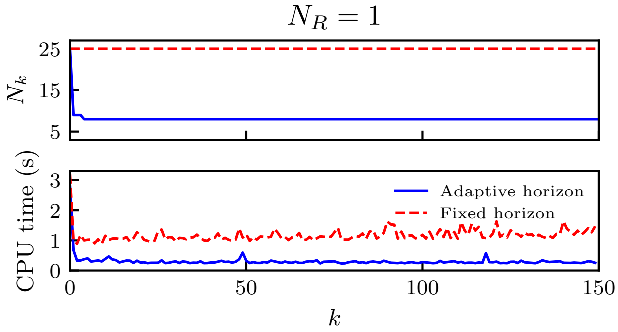

Figure 5.12: Quad-tank — plot comparing prediction horizons and computation times at each iteration for ideal multi-stage and adaptive horizon multi-stage MPC with $N_R = 1$.

There is a reduction in the prediction horizon as the process approaches its setpoint. The horizon update algorithm is unaffected by the pulse changes in the tank levels. This is probably because the terminal radii values (see Table 5.2) are of the same order as the tank levels. Therefore the pulse changes do not displace the system far enough from the terminal region to cause a significant horizon increase. The average CPU time of the fixed horizon multi-stage MPC was significantly reduced by 74% with an adaptive horizon when $N_R = 1$. Similarly, the CPU time reduction is 33% when $N_R = 2$.

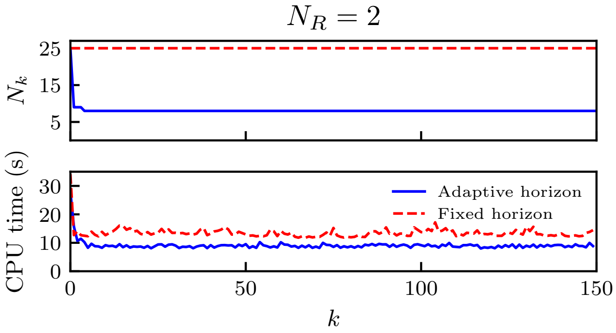

Figure 5.13: Quad-tank — plot comparing prediction horizons and computation times at each iteration for ideal multi-stage and adaptive horizon multi-stage MPC with $N_R = 2$.

5.4 Conclusion

This chapter adapts the existing adaptive horizon algorithm in [77] and extends it for multi-stage MPC to improve much-needed computational efficiency. First, the terminal costs and regions are computed independently for each uncertain parameter realization about their optimal steady states. The existence of a common terminal region is assumed for all uncertain parameter realizations, and the prediction horizon is updated such that it is always reached.

The adaptive horizon multi-stage MPC formulation is found to be both computationally efficient and robust. Simulation experiments show that the control performance of the adaptive horizon multi-stage MPC framework is similar to that of the fixed horizon multi-stage NMPC. However, the adaptive horizon framework has faster solve times than the fixed horizon multi-stage MPC. The adaptive horizon update is beneficial for multi-stage MPC to reduce NLP size and computational delay. This method can be applied to control a nonlinear system provided that it is attracted to a terminal region.

6 | Stability Properties of the Adaptive Horizon Multi-stage MPC

"Design is not just what it looks like and feels like. Design is how it works."

Steve Jobs (1955-2011)

This chapter presents and discusses the recursive feasibility and stability properties of multi-stage MPC and standard MPC with variable horizon lengths. A theoretical analysis is presented to show the effects of the robust horizon assumption on the recursive feasibility and robust constraint satisfaction for adaptive horizon multi-stage MPC.

The results in this chapter are adopted from the article "Stability properties of the adaptive horizon multi-stage MPC" [22].

6.1 Motivation

In the previous chapter, an adaptive horizon approach has been proposed to address the issue of computational burden in multi-stage MPC. The horizon update algorithm, however, has to guarantee that the resulting optimization problem is always feasible. The algorithm allows expansion of the prediction horizon to ensure closed-loop stability when look-ahead predictions do not meet terminal conditions. Therefore, it is vital to perform a theoretical analysis to investigate the conditions and margins for recursive feasibility and stability properties of the adaptive horizon multi-stage MPC. The recursive feasibility analysis is done for both the fully branched scenario tree and robust horizon formulations to understand their limitations. A stability analysis under reasonable assumptions shows that the adaptive horizon multi-stage MPC is Input-to-State practically Stable (ISpS) for both shrinking and expanding horizon lengths. Finally, a numerical example demonstrates the control performance of the proposed controller when all conditions of recursive feasibility and ISpS are satisfied.

6.2 Preliminaries

Applying the optimal control input computed by the MPC controller to system (2.1) results in a closed-loop uncertain system whose dynamics are denoted by $x_{k+1} = F(x_k, d_k)$. Stability analysis is performed for the closed-loop uncertain system. Input-to-state practical stability (ISpS) is the suitable framework to analyze the stability properties of such a system.

A standard MPC equivalent with an adaptive horizon can be obtained from (5.8) by setting $d_i^c = d_i^0$ for all $c \in \mathbb{C}$ as follows:

$$V_{N_k}^{\text{ahm}}(x_k, \mathbf{d}^0) = \min_{z_i^c, \nu_i^c} \sum_{c \in \mathbb{C}} \omega_c \Big(\psi(z_{N_k}^c, d_{N_k-1}^0) + \sum_{i=0}^{N_k-1} \ell(z_i^c, \nu_i^c, d_i^0) \Big) \tag{6.1a}$$

$$\text{s.t.} \quad z_{i+1}^c = f(z_i^c, \nu_i^c, d_i^0), \quad i = 0, \dots, N_k - 1 \tag{6.1b}$$

$$z_0^c = x_k, \tag{6.1c}$$

$$\nu_i^c = \nu_i^{c'}, \quad \{(c, c') \mid z_i^c = z_i^{c'}\} \tag{6.1d}$$

$$d^0_{i-1} = d^0_i, \quad i = N_R, \ldots, N_k - 1 \tag{6.1e}$$

$$z^c_i \in \mathbb{X},\ \nu^c_i \in \mathbb{U},\ d^0_i \in \mathbb{D}, \tag{6.1f}$$

$$z^c_{N_k} \in \mathbb{X}^c_f, \tag{6.1g}$$

$$\forall\, c, c' \in \mathbb{C},$$

where (6.1) effectively consists of c identical copies of the standard MPC with an adaptive horizon.

It follows from Theorem 1 that the difference between the primal solutions (z_i, ν_i) of (5.8) and (6.1) is $|w^\star(\mathbf{d}^c) - w^\star(\mathbf{d}^0)| \le L_s|\Delta\mathbf{d}|$, where $w^\star = [z^{\star\top}, \nu^{\star\top}]^\top$, $L_s > 0$ is the Lipschitz constant, and $|\Delta\mathbf{d}|$ is defined as:

$$|\Delta\mathbf{d}| := \max_{d^r, d^{r'} \in \mathbb{D}} |d^r - d^{r'}| \tag{6.2}$$

that is the maximum difference in the disturbance vector between any two parameter realizations. Since the evolution of (2.1) in closed-loop depends on the true disturbance d_k, then $x_{k+1} = f(x_k, \kappa(x_k), d_k)$, and $x_{k+1} \le f(x_k, \kappa(x_k), d^0_k) + \mathcal{O}(|\Delta\mathbf{d}|)$. This result facilitates the recursive feasibility and ISpS stability analysis for the adaptive horizon multi-stage MPC that follows. But first, let us define the following basic assumptions:

Assumption 4. (Basic assumptions for adaptive horizon multi-stage MPC) The adaptive horizon multi-stage MPC has the following properties:

A. Lipschitz continuity: The functions f, ℓ, and ψ are Lipschitz continuous with respect to x, u and d in the compact set $\mathbb{X} \times \mathbb{U} \times \mathrm{convhull}(\mathbb{D})$.
B. Constraint set: The constraint sets $\mathbb{X}$ and $\mathbb{U}$ are compact, and contain the origin in their interiors.
C. The solution to (5.8) satisfies LICQ and SSOSC such that Theorem 1 applies.

The equality constraints in (5.8) are always feasible and their gradients have full rank because, for any fixed admissible input ν^c_i and initial condition x_k, there exists a unique solution. Therefore, the equality constraints in (5.8) are linearly independent. In addition, LICQ is ensured at the solution of (5.8) by including the inequalities of only one of the control input variables subjected to NACs. SSOSC holds at the solution of (5.8) when regularization terms for all variables are added to the objective.With these properties, the Lipschitz continuity holds w.r.t. x_k and $\mathbf{d}^c$ for the optimal solution and the optimal cost $V^{\text{ahm}}_{N_k}$.[1]

Assumption 5. (Property of the stage cost) Given a common control law $\kappa\colon \mathbb{X} \mapsto \mathbb{U}$ then for any parameter realization $d^r \in \mathbb{D}$, the stage cost ℓ is bounded as follows: $\alpha_p(|x|) \leq \ell(x, \kappa(x), d^r) \leq \alpha_q(|x|) + \sigma_q(|d^r - d^0|)$ where $\alpha_p(\,\cdot\,),\ \alpha_q(\,\cdot\,) \in \mathcal{K}_\infty$, and $\sigma_q(\,\cdot\,) \in \mathcal{K}$.

Assumption 5 requires that the stage cost is positive, with lower and upper bounds proportional to the state at the nominal disturbance. When the disturbance realization is not nominal, the upper bound increases by a term proportional to the distance of the disturbance realization from the nominal.

Assumption 6. (Assumptions on terminal conditions of the adaptive horizon multi-stage MPC). Let Assumption 4 hold for all parameter realizations $d^r \in \mathbb{D}$ then:

A. There exists a common terminal region $\mathbb{X}_f = \bigcap\limits_{r \in \mathbb{Q}} \mathbb{X}^r_f$, that is robust control invariant.
B. $\mathbb{X}_f \subseteq \mathbb{X}$ is compact and contains the origin in its interior.
C. The terminal cost is bounded as follows: $\alpha_{p,\psi}(|x|) \leq \psi(x, d^r) \leq \alpha_{q,\psi}(|x|) + \sigma_{q,\psi}(|d^r - d^0|)$ where $\alpha_{p,\psi}, \alpha_{q,\psi} \in \mathcal{K}_\infty$, and $\sigma_{q,\psi} \in \mathcal{K}$.
D. The terminal cost is a local control Lyapunov function $\psi(f(x, \kappa_f(x), d^r), d^r) - \psi(x, d^r) \leq -\ell(x, \kappa_f(x), d^r)$ for all $x \in \mathbb{X}_f$.

Assumption 6 is needed because a common terminal region must exist for a possible reduction of the prediction horizon. The selection of a prediction horizon where all scenarios reach their respective terminal regions in Step 11 of Algorithm 2 implies that the common terminal region is reached. This region may be a null set when $|\Delta \mathbf{d}|$ is sufficiently large. Further, this can be determined beforehand when computing the terminal ingredients.

[1] The LICQ condition may be relaxed to MFCQ or CRCQ, and SSOSC relaxed to GSSOSC. In that case, a similar Lipschitz argument can be made [82].

6.3 Recursive feasibility of the adaptive horizon multi-stage MPC with a fully branched scenario tree

Before stability analysis, recursive feasibility must be guaranteed by ensuring robust constraint satisfaction. Given the assumptions above, we first present the following result for a fully branched scenario tree i.e. $N_R = N_k$:

Theorem 4. *(Recursive feasibility of the adaptive horizon multi-stage MPC). Suppose Assumption 4, 5 and 6 hold, then problem* (5.8) *from a fully branched scenario tree $N_R = N_k$ is recursively feasible.*

Proof. Let Assumption 4 and 6 hold and consider the fully branched multi-stage MPC with $N_R = N_k$. Proposition 4 in [88] and Theorem 6 in [89] show that if a robust control invariant terminal region exists then the fully branched multi-stage MPC with a fixed horizon $N_R = N_k$ is recursively feasible. Since the adaptive horizon multi-stage MPC with $N_R = N_k$ ensures that the horizon is updated such that the common terminal region is always reached, it is also recursively feasible for any horizon update N_{k+1} given by Algorithm 2. □

6.4 Recursive feasibility of the adaptive horizon multi-stage MPC with a robust horizon

Recursive feasibility cannot be guaranteed by using the same arguments in Theorem 4 when the scenario tree is not fully branched. But the infeasibility of problem (5.8) when $N_R < N_k$ may be avoided by relaxing the state inequalities using soft constraints [20]. However, robust constraint satisfaction is not guaranteed because the state trajectories may cross the feasible set $\mathbb{X}$. Therefore, before proceeding to the stability analysis the following assumption must be made:

Assumption 7. Problem (5.8) is feasible at time t_k when $N_k = N_{\max}$. Furthermore, if problem (5.8) at time t_k with x_k and N_k is feasible, then so is problem (5.8) solved at time t_{k+1} with x_{k+1} and $N_{k+1} = H(x_k, N_k, x_{k+1}) \in \mathcal{N}_{k+1}$ for all $x_k, x_{k+1} \in \mathbb{X}$, $N_k \in \mathcal{N}_{k+1}$.

Assumption 7 ensures that H (see Definition 21) produces feasible horizon lengths for any robust horizon $N_R \leq N_k$, essentially implying recursive feasibility. This allows us to present results on the ISpS property of the adaptive horizon multi-stage MPC.

6.5 ISpS for adaptive horizon multi-stage MPC

First, the relationships between the linear and nonlinear systems are needed because the nonlinear system (2.1) is controlled by a stabilizing linear controller inside the common terminal region $\mathbb{X}_f$. Consider the nominal problem (6.1) inside the terminal region evaluated using a stabilizing terminal control law κ_f^0:

$$V_N^{\text{lqr}}(x_k, \mathbf{d}^0) := \sum_{c\in\mathbb{C}} \omega_c \Big(\psi(z_N^0, d_{N-1}^0) + \sum_{i=0}^{N-1} \ell(z_i^0, \kappa_f^0(z_i^0), d_i^0) \Big) \tag{6.3}$$

where $z_{i+1}^0 = A_0 z_i^0 + B_0 \nu_i^0 + \psi(z_i^0, d^0)$, $\forall i = 0, \dots, N-1$, and $V_N^{\text{lqr}}(x_k, \mathbf{d}^0)$ is the nominal LQR cost. The relationships between the optimal costs of the nominal equivalents of the LQR and adaptive horizon multi-stage MPC in the terminal region are given in Lemma 5.

Lemma 5. *(Effect of linearization on costs inside the terminal region) There exists $\alpha_n \in \mathcal{K}_\infty$ such that $|V_N^{\text{lqr}}(x, \mathbf{d}^0) - \psi(x, d^0)| \leq \alpha_n(|x|)$ and $\alpha_v \in \mathcal{K}_\infty$ such that $|\psi(x, d^0) - V_N^{\text{ahm}}(x, \mathbf{d}^0)| \leq \alpha_v(|x|)$ for all $x \in \mathbb{X}_f$ and for all $N \in \mathcal{N}$.*

Proof. Because this is a nominal problem, the result holds because of Lemmas 18 and 19 in [77]. □

Assumption 8. The solution to (6.1) with a prediction horizon $N_k \geq N_{\min}$ satisfies:

$$\alpha_n(|z_{N_k|k}^0|) - \alpha_p(|x_k|) \leq -\alpha_3(|x_k|) \text{ if } N_{k+1} \geq N_k \tag{6.4a}$$

$$\alpha_v(|z_{N_{k+1}+1|k}^0|) - \alpha_p(|x_k|) \leq -\alpha_3(|x_k|) \text{ if } N_{k+1} < N_k \tag{6.4b}$$

for some $\alpha_3 \in \mathcal{K}_\infty$, where $N_{k+1} = H(x_k, N_k, z^0_{k+1|k}) \in \mathcal{N}_{k+1}$, α_p satisfy Assumption 5, and α_n, α_v satisfy Lemma 5.

This assumption implies that the approximation error from using the stabilizing linear control law inside the terminal region must be negligible. Assumption 8 can be satisfied by finding a suitably large $N_{\min}$ that is determined through simulations as explained in [77].

Theorem 6. *(ISpS stability of adaptive horizon multi-stage MPC) Suppose Assumption 5 and 6 hold for (5.8), the optimal value function V^{ahm} is an ISpS Lyapunov function and the resulting closed-loop system is ISpS stable.*

Proof. To show the ISpS property of the adaptive horizon multi-stage MPC, a Lyapunov function must exist that satisfies the three conditions in (2.6).

Lower bound: Suppose Assumption 5 holds then there exists a lower bound (2.6a) on the optimal value function.

$$V_N^{\text{ahm}}(x_k, \mathbf{d}^c) \geq \ell(x_k, \kappa_N(x_k), d_0^0) \geq \alpha_p(|x_k|) \quad \forall x_k \in \mathbb{X} \tag{6.5}$$

Upper bound: The existence of an upper bound (2.6b) is shown as follows:

$$\begin{aligned} V_N^{\text{ahm}}(x_k, \mathbf{d}^c) &\leq V_N^{\text{ahm}}(x_k, \mathbf{d}^0) + L_v|\Delta\mathbf{d}| && \text{(6.6a)} \\ &\leq \psi(x_k, d^0) + L_v|\Delta\mathbf{d}| && \text{(6.6b)} \\ &\leq \alpha_{q,\psi}(|x_k|) + c_1 && \text{(6.6c)} \end{aligned}$$

for all $x_k \in \mathbb{X}_f$, where $c_1 = L_v|\Delta\mathbf{d}|$ is a constant. It also applies to all $x_k \in \mathbb{X}$ by induction because $\mathbb{X}_f \subseteq \mathbb{X}$, and $V_N^{\text{ahm}}(x_k, \mathbf{d}^0)$ is continuous at the origin. Equation (6.6a) follows from Lipschitz continuity property in Assumption 4. Since $V_N^{\text{ahm}}(x_k, \mathbf{d}^0)$ is the equivalent of the standard MPC cost function then (6.6b) follows from the monotonicity property [90]. Finally, (6.6c) follows from the boundedness of the terminal cost in Assumption 6.

Function descent: The descent property (2.6c), which is the main part of stability analysis, is shown for two possible cases of horizon update:

1. *A decreasing prediction horizon $N_R \le N_{k+1} < N_k$:* An approximate solution for (6.1) at time t_{k+1} is found by shifting the optimal solution at time t_k as follows:

$$\hat{\nu}^0_{i|k+1} = \nu^0_{i+1|k}, \qquad i = 0, \dots, N_{k+1} - 1, \tag{6.7}$$

and the corresponding initialization for state variables:

$$\hat{z}^0_{0|k+1} = z^0_{1|k}, \tag{6.8a}$$

$$\hat{z}^0_{i+1|k+1} = f(\hat{z}^0_{i|k}, \hat{\nu}^0_{i|k}, d^0_i) \tag{6.8b}$$

leads to a suboptimal cost such that the descent inequality becomes:

$$\begin{aligned}
&V^{\text{ahm}}_{N_{k+1}}(f(x_k, \kappa(x_k), d^0_0), \mathbf{d}^0) - V^{\text{ahm}}_{N_k}(x_k, \mathbf{d}^0)\\
&\le \sum_{c\in\mathbb{C}} \omega_c \Big(\psi(\hat{z}^0_{N_{k+1}|k+1}, d^0_{N_{k+1}-1}) + \sum_{i=0}^{N_{k+1}-1} \ell(\hat{z}^0_{i|k+1}, \hat{\nu}^0_{i|k+1}, d^0_i)\Big) \\
&- \sum_{c\in\mathbb{C}} \omega_c \Big(\psi(\nu^0_{N_k|k}, d^0_{N_k-1}) - \sum_{i=0}^{N_k-1} \ell(z^0_{i|k}, \nu^0_{i|k}, d^0_i)\Big) \qquad (6.9a)\\
&= -\ell(x_k, \kappa(x_k), d^0_0) + \psi(\hat{z}^0_{N_{k+1}|k+1}, d^0_{N_{k+1}-1}) \\
&+ \sum_{i=0}^{N_{k+1}-1} \big(\ell(\hat{z}^0_{i|k+1}, \hat{\nu}^0_{i|k+1}, d^0_i) - \ell(z^0_{i+1|k}, \nu^0_{i+1|k}, d^0_i)\big) \qquad (6.9b)\\
&- \psi(z^0_{N_k|k}, d^0_{N_k-1}) - \sum_{i=N_{k+1}+1}^{N_k-1} \ell(\hat{z}^0_{i|k}, \hat{\nu}^0_{i|k}, d^0_i) \\
&= -\ell(x_k, \kappa(x_k), d^0_0) + \psi(\hat{z}^0_{N_{k+1}|k+1}, d^0_{N_{k+1}-1}) \\
&- \psi(z^0_{N_k|k}, d^0_{N_k-1}) - \sum_{i=N_{k+1}+1}^{N_k-1} \ell(\hat{z}^0_{i|k}, \hat{\nu}^0_{i|k}, d^0_i) \qquad (6.9c)\\
&= -\ell(x_k, \kappa(x_k), d^0_0) \\
&+ \psi(\hat{z}^0_{N_{k+1}|k+1}, d^0_{N_{k+1}-1}) - V^{\text{ahm}}_{N_k - N_{k+1} - 1}(z^0_{N_{k+1}+1|k}, \mathbf{d}^0) \qquad (6.9d)
\end{aligned}$$

In the presence of true uncertainty, the successor state is $x_{k+1} = f(x_k, \kappa(x_k), d_k)$

where d_k is the true parameter realization, such that:

$$\begin{aligned}&V^{\mathrm{ahm}}_{N_{k+1}}(x_{k+1},\mathbf{d}^0) - V^{\mathrm{ahm}}_{N_k}(x_k,\mathbf{d}^0) \leq -\ell(x_k,\kappa(x_k),d^0_0)\\ &+\psi(\hat{z}^0_{N_{k+1}|k+1},d^0_{N_{k+1}-1}) - V^{\mathrm{ahm}}_{N_k-N_{k+1}-1}(z^0_{N_{k+1}+1|k},\mathbf{d}^0)\\ &+L_k|d_k-d^0_i|\end{aligned} \tag{6.9e}$$

Then from Eqs. (5.8) and (6.1) we have:

$$V^{\mathrm{ahm}}_{N_k}(x_k,\mathbf{d}^c) = V^{\mathrm{ahm}}_{N_k}(x_k,\mathbf{d}^0) + \mathcal{O}(|\Delta\mathbf{d}|),\ \text{and} \tag{6.10a}$$

$$V^{\mathrm{ahm}}_{N_{k+1}}(x_{k+1},\mathbf{d}^c) = V^{\mathrm{ahm}}_{N_{k+1}}(x_{k+1},\mathbf{d}^0) + \mathcal{O}(|\Delta\mathbf{d}|) \tag{6.10b}$$

that can be combined with (6.9e) to form:

$$\begin{aligned}&V^{\mathrm{ahm}}_{N_{k+1}}(x_{k+1},\mathbf{d}^c) - V^{\mathrm{ahm}}_{N_k}(x_k,\mathbf{d}^c) \leq -\ell(x_k,\kappa(x_k),d^0_0)\\ &+\psi(\hat{z}^0_{N_{k+1}|k+1},d^0_{N_{k+1}-1}) - V^{\mathrm{ahm}}_{N_k-N_{k+1}-1}(z^0_{N_{k+1}+1|k},\mathbf{d}^0)\\ &+L_k|d_k-d^0_i| + \left(V^{\mathrm{ahm}}_{N_{k+1}}(x_{k+1},\mathbf{d}^c) - V^{\mathrm{ahm}}_{N_{k+1}}(x_{k+1},\mathbf{d}^0)\right)\\ &+\left(V^{\mathrm{ahm}}_{N_k}(x_k,\mathbf{d}^c) - V^{\mathrm{ahm}}_{N_k}(x_k,\mathbf{d}^0)\right)\end{aligned} \tag{6.11a}$$

$$\begin{aligned}&\leq -\ell(x_k,\kappa(x_k),d^0_0) + \psi(\hat{z}^0_{N_{k+1}|k+1},d^0_{N_{k+1}-1})\\ &- V^{\mathrm{ahm}}_{N_k-N_{k+1}-1}(z^0_{N_{k+1}+1|k},\mathbf{d}^0) + L_k|d_k-d^0_i| + 2L_v|\Delta\mathbf{d}|\end{aligned} \tag{6.11b}$$

$$\leq -\alpha_p(|x_k|) + \alpha_v(|z^0_{N_{k+1}+1|k}|) + L_k|d_k-d^0_i| + 2L_v|\Delta\mathbf{d}| \tag{6.11c}$$

$$\leq -\alpha_3(|x_k|) + \sigma(|d_k-d^0_i|) + c_2 \tag{6.11d}$$

where $c_2 = 2L_v|\Delta\mathbf{d}| \geq 0$ and (6.11c) follows from Lemma 5 and (6.11d) follows from (6.4b).

2. *An increasing prediction horizon $N_{k+1} \geq N_k$:* Similar to the first case approximate the solution of (6.1) at t_{k+1} by shifting:

$$\hat{\nu}^0_{i|k+1} = \begin{cases}\nu^0_{i+1|k} & \forall\ i = 0,\dots,N_k-2\\ -K_0 z^0_i & \forall\ i = N_k-1,\dots,N_{k+1}-1\end{cases} \tag{6.12}$$

The initialization of the states is the same as in (6.8). Then the descent inequality becomes:

$$
\begin{aligned}
&V^{\text{ahm}}_{N_{k+1}}(f(x_k,\kappa(x_k),d^0_0),\mathbf{d}^0) - V^{\text{ahm}}_{N_k}(x_k,\mathbf{d}^0) && (6.13\text{a})\\
&\leq \sum_{c\in\mathbb{C}} \omega_c\Big(\psi(\hat{z}^0_{N_{k+1}|k+1}, d^0_{N_{k+1}-1}) + \sum_{i=0}^{N_{k+1}-1} \ell(\hat{z}^0_{i|k+1},\hat{\nu}^0_{i|k+1},d^0_l)\Big) && \\
&-\sum_{c\in\mathbb{C}} \omega_c\Big(\psi(z^0_{N_k|k}, d^0_{N_k-1}) - \sum_{i=0}^{N_k-1} \ell(z^0_{i|k},\nu^0_{i|k},d^0_i)\Big) && (6.13\text{b})\\
&= -\ell(x_k,\kappa(x_k),d^0_0) && \\
&+\sum_{i=0}^{N_k-2} \big(\ell(\hat{z}^0_{i|k+1},\hat{\nu}^0_{i|k+1},d^0_i) - \ell(z^0_{i+1|k},\nu^0_{i+1|k},d^0_i)\big) && (6.13\text{c})\\
&+\psi(\hat{z}^0_{N_{k+1}|k+1},d^0_{N_{k+1}-1}) + \sum_{i=N_k-1}^{N_{k+1}-1} \ell(\hat{z}^0_{i|k},\hat{\nu}^0_{i|k},d^0_i) && \\
&-\psi(z^0_{N_k|k},d^0_{N_k-1}) && \\
&= -\ell(x_k,\kappa(x_k),d^0_0) + \psi(\hat{z}^0_{N_{k+1}|k+1},d^0_{N_{k+1}-1}) && \\
&+\sum_{i=N_k-1}^{N_{k+1}-1} \ell(\hat{z}^0_{i|k+1},\hat{\nu}^0_{i|k+1},d^0_i) - \psi(z^0_{N_k|k},d^0_{N_k-1}) && (6.13\text{d})\\
&= -\ell(x_k,\kappa(x_k),d^0_0) + V_{N_k-N_{k+1}+1}(z^0_{N_k|k},\mathbf{d}^0) && (6.13\text{e})\\
&-\psi(z^0_{N_k|k},d^0_{N_k-1}) &&
\end{aligned}
$$

In the presence of true parameter realization d_k then:

$$
\begin{aligned}
&V^{\text{ahm}}_{N_{k+1}}(x_{k+1},\mathbf{d}^0) - V^{\text{ahm}}_{N_k}(x_k,\mathbf{d}^0) \leq -\ell(x_k,\kappa(x_k),d^0_0)\\
&+V^{lqr}_{N_{k+1}-N_k+1}(z^0_{N_k|k},\mathbf{d}^0) - \psi(z^0_{N_k|k},d^0_{N_k-1}) \qquad (6.13\text{f})\\
&+L_k|d_k - d^0_i|
\end{aligned}
$$

Substituting (6.10a) and (6.10b) on (6.13f) we have:

$$
\begin{aligned}
&V^{\text{ahm}}_{N_{k+1}}(x_{k+1},\mathbf{d}^c) - V^{\text{ahm}}_{N_k}(x_k,\mathbf{d}^c) \leq -\ell(x_k,\kappa(x_k),d^0_0)\\
&+V^{lqr}_{N_{k+1}-N_k+1}(z^0_{N_k|k},\mathbf{d}^0) - \psi(z^0_{N_k|k},d^0_{N_k-1}) \qquad (6.14\text{a})\\
&+L_k|d_k - d^0_i| + 2L_v|\Delta\mathbf{d}|
\end{aligned}
$$

$$\begin{aligned}
&\leq -\ell(x_k, \kappa(x_k), d_0^0) + V^{lqr}_{N_{k+1}-N_k+1}(z^0_{N_k|k}, \mathbf{d}^0) \\
&- \psi(z^0_{N_k|k}, d^0_{N_k-1}) + L_k|d_k - d_i^0| + 2L_v|\Delta\mathbf{d}| \qquad (6.14b) \\
&\leq -\alpha_p(|x_k|) + \alpha_n(|z^0_{N_{k|k}}|) + L_k|d_k - d_i^0| + 2L_v|\Delta\mathbf{d}| \qquad (6.14c) \\
&\leq -\alpha_3(|x_k|) + \sigma(|d_k - d_i^0|) + c_2 \qquad (6.14d)
\end{aligned}$$

where $c_2 = 2L_v|\Delta\mathbf{d}| \geq 0$, and (6.14c) follows from Lemma 5 and (6.14d) follows from (6.4a).

Thus $V^{\text{ahm}}_{N_k}$ satisfies the descent property (2.6c) for all the possible horizon length updates. Hence the adaptive horizon multi-stage MPC is ISpS stable.

□

6.6 Numerical example

This numerical example aims to show the performance of the proposed controller when all the conditions for recursive feasibility and ISpS stability presented in this chapter are satisfied.

This is a simple example adapted from [91] used to demonstrate the performance of an adaptive horizon multi-stage MPC with a fully branched scenario tree, and therefore suitable terminal conditions that guarantee recursive feasibility and ISpS stability (see Section 6.3). The system model is written as:

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -\frac{k_0}{m}\mathrm{e}^{-x_1}x_1 - \frac{h_d}{m}x_2 + \frac{u}{m}
\end{aligned}$$

where the state variable vector $x = [x_1,\ x_2]^\top$ includes the displacement and the velocity of the mass $m = 1$ kg, respectively. $k = k_0\mathrm{e}^{-x_1}$ is the elastic constant of the spring and $k_0 = 0.33$ N/m. The damping factor h_d is uncertain and can have three different possible values $h_d^r = \{1.0, 2.0, 4.0\}$ Ns/m with equal probabilities. The damping factor is assumed to vary unpredictably between sampling intervals of $\Delta t = 0.4$ s.

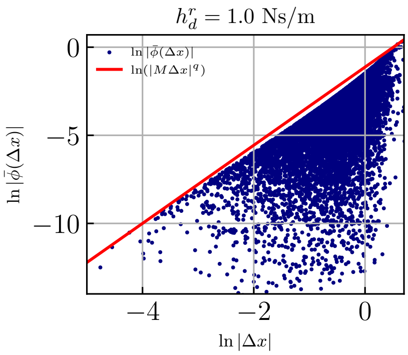

Figure 6.1: Spring-damper-mass — plots of linearization error against $|\Delta x|$ for 10000 simulations showing the bound (red line).

The objective is to control the mass to its equilibrium position $x_1 = 0$ using an external force u. The stage cost is given by $\ell = x^\top Qx + u^\top Ru$ and h_d varies randomly at each time step with the previously defined probabilities.

6.6.1 Approximation of terminal conditions

The LQR method explained in Section 5.2.2 is applied to find suitable ψ and $\mathbb{X}_f$. The tuning matrices used are $Q = \mathrm{diag}([30,\ 20])$ and $R = 1$. The terminal control law must be common for all scenarios due to full branching (see Section 5.2.2). Therefore, $K = [1.7409\ 2.0959]$ is chosen with $\kappa_f(x) = -Kx$ for all $x \in \mathbb{X}_f$. Different values of P_r are computed using the Lyapunov equation to obtain the respective terminal costs for each realization. The terminal radii are determined by performing 10^5 one-step simulations of the closed-loop system and estimating the linearization error bound. The linearization error of this system is independent of h_d, so the simulations were performed for only one of the disturbance realizations. After obtaining bound parameters $M_r = 0.3235$ and $q_r = 2.2176$, the terminal radii were found to be $c_f^r \approx \{0.8032,\ 0.8922,\ 0.6690\}$.

6.6.2 Simulation results

The simulations are done using `JuMP v.0.21.10` [83] as the NLP modeler in a `Julia` [84] environment. The NLP solver used is `IPOPT 3.13.4` [85], and

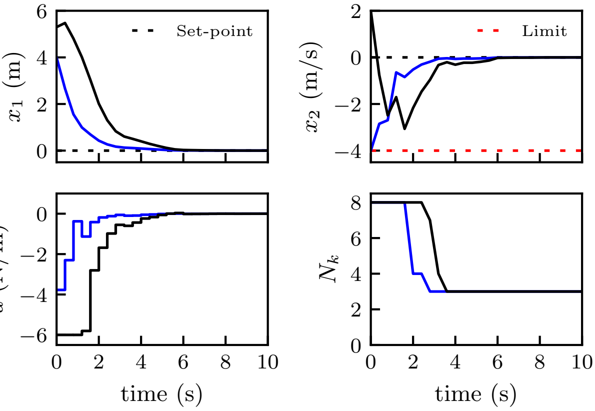

Figure 6.2: Spring-damper-mass — simulation results from two initial conditions $x_0 = [-4,\ 4]$ (blue lines) and $x_0 = [5.3,\ 2]^\top$ (black lines) showing the control performance of a fully branched adaptive horizon multi-stage MPC.

the linear solver is `HSL-MA57` by the STFC Rutherford Appleton Laboratory, on a 2.6 GHz Intel Core-i7 with 16 GB memory.

An initial prediction horizon of $N_0 = 8$ is chosen with $N_{\min} = 2$. The scenario tree is fully branched at each MPC iteration making a total of 6561 scenarios at the first iteration. Fig. 6.2 shows two sets of closed-loop simulations of the adaptive horizon multi-stage MPC that are done with initial conditions $x_0 = [-4,\ 4]^\top$ m (blue line) and $[5.3,\ 2]^\top$ m (black line). The state bounds are $x_1 \in [-10, 10]$ m and $x_2 \in [-4, 10]$ m/s and the control input bounds are $u \leq |6.0\,N|$. The system is controlled to its setpoint and the horizon length N_k is reduced as the setpoint is approached. Eventually, the N_k is reduced to 3 from 8 cutting the number of scenarios from 6561 to 27. This results in a speedup of computation times by a factor of 10 to 40 as illustrated in Fig. 6.3. Therefore, this example also demonstrates the advantage of the adaptive horizon method for computational cost savings in multi-stage MPC with a fully branched scenario tree.

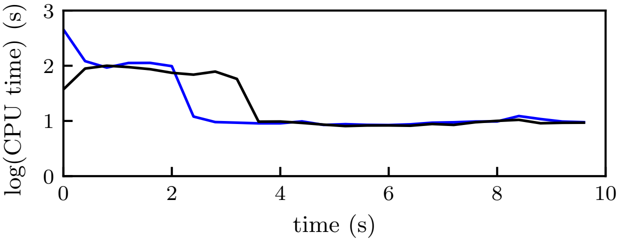

Figure 6.3: Spring-damper-mass — computation times (in logarithmic scale) from two initial conditions $x_0 = [-4,\ 4]$ (blue lines) and $x_0 = [5.3,\ 2]^\top$ (black lines) showing the computational efficiency of a fully branched adaptive horizon multi-stage MPC.

6.7 Conclusion

This chapter shows under reasonable assumptions that the adaptive horizon multi-stage MPC framework is recursively feasible when the scenario tree is fully branched. Implementation of soft constraints avoids problem infeasibility in practice when the scenario tree is not fully branched, but robust constraint satisfaction is not guaranteed. ISpS of the fixed horizon multi-stage MPC is retained by assuming a negligible linear control error inside the common terminal region.

Simulation results demonstrate an effective reduction in the prediction horizon and computational delay, provided the system progressively approaches its setpoint, and that it is not significantly disturbed away from it. Future avenues of this include handling economic costs, and the use of parametric sensitivities for both horizon and critical scenario updates to further reduce computation time.

Part II

Scenario Selection for Multi-stage MPC

7 Scenario Selection from Data and NLP Sensitivity Analysis

"The world is continuous, but the mind is discrete."

David Mumford (1937-present)

In this chapter, different scenario selection approaches for multi-stage MPC are presented. It is preferred that scenarios be selected such that the optimization problem shrinks in size, thus computationally efficient, and without an overly conservative solution. The effectiveness of the conventional box over-approximation, sensitivity-assisted, and data-driven scenario selection approaches are compared with the aid of a simple optimization problem. Results demonstrate that performing a combination of data and sensitivity analyses results in a smaller-sized optimization problem with a less conservative solution.

The results presented in this chapter are based on the article: "Sensitivity-based scenario selection for multi-stage MPC along principal components: Applied to robust thermal energy storage operation." [31]

7.1 Motivation

This chapter aims to examine the effect of scenario selection on the conservativeness of multi-stage MPC formulations and identify the best strategy. The conventional scenario selection approach implemented in multi-stage MPC is the box over-approximation that is likely to result in highly conservative solutions. To reduce conservativeness, the parameter realizations in the scenario tree must be closer to the actual process. This can be achieved through the extraction of scenarios from large process data sets. To improve scenario selection for multi-stage MPC, simple multivariate data analysis techniques have been applied. Data-driven approaches have been implemented in [92] to calibrate approximate uncertainty sets for a scenario-based stochastic MPC using support vector clustering, and in [93] for scenario selection in multi-stage MPC using principal component analysis (PCA).

The PCA-based scenario selection for multi-stage MPC performs an offline analysis of historical process data to select uncertainty representations "a-priori", by exploiting existing correlations in data. Further work in [94] presents how time-varying uncertainty can be handled by performing online PCA calculations to update scenarios whenever new data is acquired. Thus, the selected scenarios in multi-stage MPC can be dynamically adjusted based on data to achieve a less conservative performance. This method has been applied in [13] for robust operation of thermal energy storage in an industrial cluster with district heating using actual process data.

These methods select finite parameter realizations along the dominant principal component (PC), or along the axes in a reduced space of the PCs that capture most of the process variance. As a result, a less conservative solution (i.e. more economic benefits) is achieved while maintaining robustness. Also, the number of scenarios and size of the optimization problem scale exponentially with the number of PCs. Therefore, if a smaller number of PCs are used to approximate the uncertainty space then computational efficiency can also be achieved. However, problem infeasibility may occur because of the ignored non-dominant PCs that may have a significant effect on the constraints.

To ensure that the assembled scenarios avoid constraint violations, the parameter realizations must correspond the the worst-case performance [19]. Provided that a nonlinear system is monotone in its feasible domain, [20] shows that it is trivial to find the worst-case parameter realization using NLP sensitivity analysis. The sensitivities filter out parameters that do not affect inequality constraints and select the worst-case realization along the axes of each of the critical parameters to assemble the "critical" scenarios. As a result, the size of the sensitivity-assisted multi-stage NMPC (SAMNMPC) problem becomes significantly smaller because it scales linearly with the number of critical scenarios and not exponentially with the number of parameter realizations. In this case, the SAMNMPC is computationally efficient but highly conservative because the realizations are selected using the conventional box over-approximation.

To address conservativeness and computational burden in multi-stage MPC, a scenario selection approach that combines both PCA and NLP sensitivities is proposed. This chapter illustrates the motivation and demonstrates how the proposed scenario selection approach mitigates the issues in multi-stage MPC with the aid of a simple example.

7.2 An illustrative example

Consider a linear-quadratic optimization problem presented below, that aims to find an optimal decision $u = [u_1, u_2]^\top$ for all the possible realizations of an uncertain parameter $d = [d_1, d_2]^\top$:

$$J^\star(u^\star, d) = \min_{0 \leq u \leq 2} (u_1 - 2)^2 + (u_2 - 1)^2 \tag{7.1a}$$

$$\text{s.t. } g_1 : (1 - d_1)u_1 + d_2(u_2 + 1) \leq 3 \tag{7.1b}$$

$$g_2 : (d_2 - 1)u_1 + d_1 u_2 \leq 1 \tag{7.1c}$$

$$\forall d \in \mathbb{D}_c, \text{and } \mathbb{D}_c = \left\{ d \mid \begin{bmatrix} -1.0 \\ 0.0 \end{bmatrix} \leq d \leq \begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix} \right\} \tag{7.1d}$$

where g_1 and g_2 are the constraint sets defined for each element in the infinite parameter set $\mathbb{D}_c$. When performing optimization under uncertainty, the

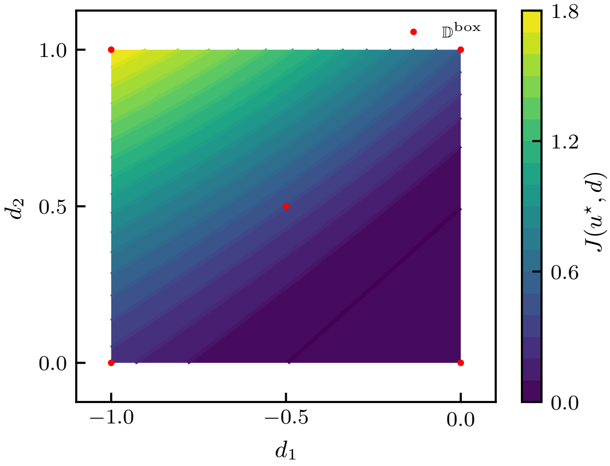

Figure 7.1: Contour plot showing the optimal cost of problem (7.1) for each $d \in \mathbb{D}$.

goal is to minimize the worst-case cost in the uncertainty region $\mathbb{D}_c$. The set $\mathbb{D}_c$ is infinite, and so we determine the worst-case realization by first creating a dense sample grid to discretize $\mathbb{D}_c$, and then solve problem (7.1) for each of the sampled realizations. The optimal cost for each sample is obtained and illustrated as a contour plot across the uncertainty set in Fig. 7.1. In this case, the worst-case corresponds to the realization $d^{\mathrm{wc}} = (-1, 1)^\top$ with an optimal cost $J^\star_{\mathrm{wc}} = 1.8$. Then the robust optimization problem (7.1) with $d = d^{\mathrm{wc}}$ becomes:

$$\begin{aligned} \min_{0 \le u \le 2} \quad & (u_1 - 2)^2 + (u_2 - 1)^2 \\ \text{s.t.} \quad & 2u_1 + u_2 \le 2 \end{aligned} \tag{7.2}$$

The feasible region corresponding to problem (7.1) with the worst-case parameter realization is illustrated by Fig. 7.2.

The dense sampling technique is computationally inefficient especially when the number of variables, number of constraints, and degree of nonlinearity increase, as is the case for multi-stage MPC problems. Therefore, a more efficient technique that selects the fewest possible samples that most probably correspond to the worst-case realization is needed. Notice that the constraints in (7.1) are monotonic with respect to the uncertain parameters

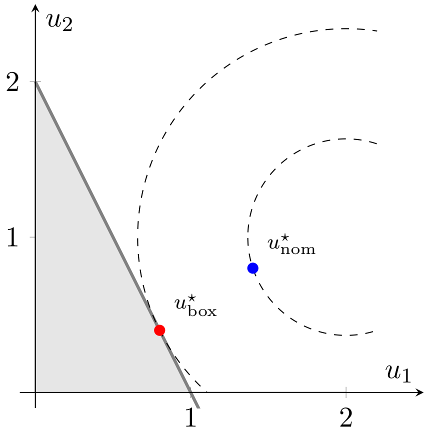

Figure 7.2: Robust feasible region for problem (7.1)

resulting in a monotonic optimal cost function as seen in Fig. 7.1. This implies that the worst case will always occur at one of the vertices of the uncertain region. Therefore, selecting the box's vertices to approximate problem (7.1) is a good approach because the worst-case will occur at one of these points. It is common to also include the nominal or expected parameter value, especially in nonlinear optimization. This technique is known as box over-approximation of the optimization with uncertainty problem.

7.3 Box over-approximation

The box over-approximation method has been commonly applied in several multi-stage MPC implementations such as to semi-batch polymerization [5–8], a batch bioreactor [9], hydrodesulfurization [10], gas lifted wells in oil and gas production [11], multi-product distillation [12], and penicillin fermentation [95]. These are nonlinear MPC applications where the combinations of the {max, nominal, min} values of each uncertain parameter were selected as the realizations assembled in the scenario tree.

Returning to the example (7.1) and applying the box over-approximation, a finite set of parameter realizations is obtained:

$$\mathbb{D}^{\text{box}} = \{(-0.5, 0.5)^\top, (-1, 0)^\top, (-1, 1)^\top, (0, 1)^\top, (0, 0)^\top\}.$$

The set of inequalities from $\mathbb{D}^{\text{box}}$ describes the same feasible region as the worst-case problem that is in Fig. 7.2. The solution is $u^{\star}_{\text{box}} = (0.8, 0.4)^{\top}$, with an optimal cost $J^{\star}_{\text{box}} = 1.8$ identical to the solution from the dense sampling.

Although this approach is simple, it can be further improved, especially when the dimensionality of the uncertain parameters and degree of nonlinearity increases. In this simple example, 5 realizations are selected but only one of them is "critical", leading to an extra 8 inequality constraints. The number of extra constraints and variables (if recourse is allowed) scales rapidly with dimensionality in the uncertain parameter, which is undesirably inefficient due to the increased computational complexity, especially for nonlinear optimization.

7.4 Sensitivity-assisted scenario selection

We want to improve computational efficiency by defining the robust feasible region for problem (7.1) using the least possible number of parameter realizations. Let us determine the realizations of d that maximize the constraints g_1 and g_2 for any feasible u. Minimizing problem (7.1) is equivalent to solving the following bilevel program:

$$\min_{0 \le u \le 2} \quad (u_1 - 2)^2 + (u_2 - 1)^2 \tag{7.3a}$$

$$\text{s.t.} \quad \max_{d \in \mathbb{D}} \, (1 - d_1)u_1 + d_2(u_2 + 1) \le 3 \tag{7.3b}$$

$$\max_{d \in \mathbb{D}} \, (d_2 - 1)u_1 + d_1 u_2 \le 1 \tag{7.3c}$$

where the finite realization set, $\mathbb{D}_c^{\text{sens}}$ is found by solving the inner optimization problems (7.3b) and (7.3c), such that:

$$\mathbb{D}_c^{\text{sens}} = \{d \mid \arg\max_{d \in \mathbb{D}} \, g_1(u, d) \cup \arg\max_{d \in \mathbb{D}} \, g_2(u, d)\} \tag{7.4}$$

To solve (7.4), determine the constraint gradients with respect to d:

$$\nabla_d g_1 = \begin{bmatrix} -u_1 & +u_2 \end{bmatrix} \quad \Rightarrow \quad d^{\text{wc}}_{[1]} = (-1, 1)^{\top} \tag{7.5a}$$

$$\nabla_d g_2 = \begin{bmatrix} +u_1 & +u_2 \end{bmatrix} \quad \Rightarrow \quad d^{\text{wc}}_{[2]} = (0,1)^\top \tag{7.5b}$$

Since $u \geq 0$ then $\nabla_d g_1$ and $\nabla_d g_2$ do not change sign, meaning g_1 and g_2 are monotones in d for any admissible u [96]. Consequently, the solution to (7.4) is trivial, with the finite realization set for (7.1) being:

$$\mathbb{D}^{\text{sens}}_c = \{(-1,1),\ (0,1)\}.$$

The set $\mathbb{D}^{\text{sens}}$ results in inequalities that describe the same feasible region as for the worst-case problem (see Fig. 7.2). Hence, selecting the parameters using sensitivities will lead to fewer constraints that define the same robust feasible region, when compared to box over-approximation. This is efficient for the optimizer due to a smaller problem size resulting in computational cost savings, especially for large-scale robust optimization problems (e.g. multi-stage MPC) provided that the sensitivities are cheap to obtain. Under the monotonicity assumption of the constraints in d, the worst-case realizations can be easily obtained even for nonlinear systems. This strategy is used by Thombre et al. [20] in the sensitivity-assisted multi-stage NMPC (SAMNMPC) algorithm to reduce the computational delay in multi-stage NMPC which is described in Chapter 4.

7.5 Nominal assumption and conservativeness

If the parameters are assumed to be nominal i.e. $d^0 = (-0.5, 0.5)^\top$, the solution to (7.1) is $u^\star_{\text{nom}} = (1.4, 0.8)^\top$, with an optimal cost $J^\star_{\text{nom}} = 0.4$. The optimal cost of the robust problem (7.1) is worse than that of the nominal problem ($J^\star_{\text{box}} > J^\star_{\text{nom}}$). This implies that optimization with uncertainty is conservative, and the loss in performance is a cost incurred for the robustness.

Nevertheless, as depicted in Fig. 7.2, it is evident that the nominal solution lies outside the region of robust feasibility. This implies that the nominal solution can be deemed feasible only under the assumption of perfect knowledge. Since achieving perfect knowledge is impractical in reality, it is advisable to account for an uncertain region surrounding the nominal (estimated) parameters. The degree of conservativeness can then be reduced by tightening the uncertainty region using historical process data. This implies that the selected parameter realizations must be determined from historical data, and whenever possible dynamically updated whenever new information is obtained. Considering that the worst-case realization value was chosen from one of the vertices, finding the worst-case parameter value closer to the actual process behavior will significantly reduce the degree of conservativeness in robust optimization.

7.6 Data-driven scenario selection

Data-driven techniques have been employed to reduce the conservativeness in optimization under uncertainty [97–99]. Moreover, multivariate analysis on process data has been applied specifically to robust MPC in prior works [93, 94, 100]. The aim is to improve uncertainty representation by revealing correlations among model parameters. Among the various techniques, principal component analysis (PCA) stands out as the most widely utilized method in the literature. It has been frequently employed for designing robust MPC controllers, and it is the selected approach in the present study. PCA is favored due to its simplicity and its potential to reduce the dimensionality of the uncertainty space.

Typically, uncertain parameters tend to exhibit correlations, leading to the formation of elliptical uncertainty sets that can be derived from the process data. For instance, the red dots in Fig. 7.3 represent data samples for a dataset corresponding to the parameters of problem (7.1). Particularly when considering the non-dominant axis of the ellipsoid, selecting the box vertices as the parameter realizations is overly cautious. In such situations, it is advisable to select the worst-case parameter realization along the ellipsoid axes. Moreover, if the method of sensitivities is applied, the uncertain parameters must be projected into a new coordinate that lies along the ellipsoid axes. This is done by a simple linear transformation which is determined from principal component analysis (PCA).

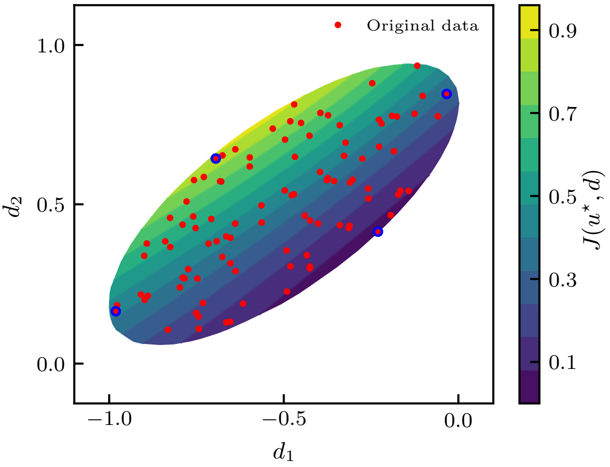

Figure 7.3: Ellipsoidal process data cloud with corresponding optimal cost values as a contour plot

7.6.1 Principal component analysis

Principal component analysis (PCA) is an orthogonal linear transformation that projects a dataset into a new coordinate system whose axes are called the *principal components* (PCs). The PCs are the unit directions that explain the total variation in the data [101]. PCA reveals hidden patterns in data by evaluating the variability in the dataset. As a result, PCA fits a hyperellipsoid to the dataset with the PCs corresponding to the ellipsoid axes. The first PC corresponds to the direction exhibiting the greatest variance within the dataset, whereas the subsequent components capture the remaining variance in descending order of explained variability. For example, in the example dataset shown in Fig. 7.3, the major axis of the ellipsoid is the first PC, and the minor axis is the second PC.

Assume a dataset with n_s samples for each uncertain parameter is represented by the matrix $\mathbf{X} \in \mathbb{R}^{n_s \times n_d}$. Before conducting the PCA procedure, it is necessary to center $\mathbf{X}$ around its mean as follows:

$$\mathbf{X_0} = \mathbf{X} - \mathbf{1} \cdot \boldsymbol{\mu}^\top \tag{7.6}$$

where $\mathbf{X_0} \in \mathbb{R}^{n_s \times n_d}$ denotes the pre-treated dataset for PCA, $\boldsymbol{\mu} \in \mathbb{R}^{n_d}$ is a vector of the means of each column in $\mathbf{X}$, and $\mathbf{1} \in \mathbb{R}^{n_s}$ is a vector of ones. Throughout this paper, we set the nominal parameters as the mean of $\mathbf{X}$, i.e. $\boldsymbol{\mu} = d^0$.

PCA is extremely sensitive to variations in scale within the dataset. Consequently, if the elements within each column exhibit dissimilar scales, it becomes essential to both standardize and mean-center the dataset $\mathbf{X}$. A technique for achieving this involves standardizing the samples to units of standard deviation, as depicted below:

$$\mathbf{X_0} = (\mathbf{X} - \mathbf{1} \cdot \boldsymbol{\mu}^\top)\boldsymbol{\Sigma}^{-1} \tag{7.7}$$

where the matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{n_d \times n_d}$ is a diagonal matrix of the standard deviations of each column in $\mathbf{X}$ i.e. $\boldsymbol{\Sigma} := \text{diag}([\sigma_1, \ldots, \sigma_{n_d}])$.

PCA applied to $\mathbf{X_0}$ yields a linearly transformed matrix denoted as $\mathbf{Y} = \mathbf{X_0}\mathbf{C}$, where $\mathbf{C} \in \mathbb{R}^{n_d \times n_p}$ represents the projection matrix with $n_p \leq n_d$. The matrix $\mathbf{Y} \in \mathbb{R}^{n_s \times n_p}$ comprises PC *scores* corresponding to each sample. These scores denote the projections of standardized samples onto the n_p principal components. Each column of the projection matrix $\mathbf{C}$ embodies weights assigned to the original samples, referred to as *loadings*. These loadings are essential for deriving the component scores. Moreover, the projection matrix $\mathbf{C}$ can be utilized to reconstruct the initial dataset $\mathbf{X}$ from the scores matrix $\mathbf{Y}$ through a linear transformation.

We represent the linear transformation aimed at reconstructing the uncertain parameter d from the PCA scores using a function denoted as $\Phi^{\text{PCA}} : \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_d}$ such that:

$$d = \Phi^{\text{PCA}}(\overline{d}) \tag{7.8}$$

where $\overline{d} \in \mathbb{R}^{n_p}$ represents a PC score. The function Φ^{PCA} is also influenced by how the original dataset $\mathbf{X}$ is treated before undergoing PCA.

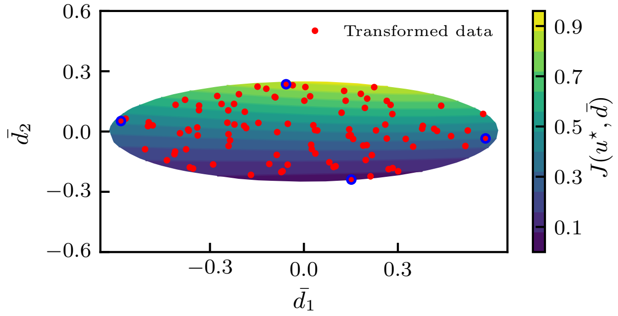

Figure 7.4: Transformed ellipsoid data cloud with the corresponding optimal cost values as a contour plot

Mean-centered samples. If the original dataset $\mathbf{X}$ is only mean-centered before the PCA procedure, the reconstruction function becomes:

$$\Phi^{\text{PCA}}(\bar{d}) = \mathbf{C}\bar{d} + d^0 \tag{7.9}$$

Standardized samples. Moreover, if the $\mathbf{X}$ is standardized by standard deviations before PCA is performed, then the projection matrix is scaled by the corresponding standard deviations, and the reconstruction function becomes:

$$\Phi^{\text{PCA}}(\bar{d}) = \mathbf{\Sigma}\mathbf{C}\bar{d} + d^0 \tag{7.10}$$

Now refer back to the dataset for the simple problem (7.1) in Fig. 7.3. PCA on the dataset gives the sample mean $d^0 = (-0.5209, 0.4914)^\top$ and the PCA projection matrix:

$$\mathbf{C} = \begin{bmatrix} 0.7813 & 0.6241 \\ -0.6241 & 0.7813 \end{bmatrix}$$

with explained variances for PC-1 and PC-2 being 84.1% and 15.9%, respectively. The data is transformed by scaling and rotation into the scores using (7.9). Fig. 7.4 also shows the sample scores (transformed data) and the contour plot of the optimal cost values at the transformed parameters.

7.7 PCA based scenario selection

PCA on data helps to select the discrete realizations of the uncertain parameters from a dataset based on explained variability. This method picks the data points corresponding to the maximum and minimum scores along the directions of the different PCs that explain the variability with sufficient component variance. The maximum and minimum scores of the example dataset are shown in Figure 7.4 as the red dots circled with blue margins. These are then transformed into the original parameter space using (7.9) and they correspond to the red dots with blue circles in Figure 7.3. The discrete realization set from PCA becomes:

$$\begin{aligned}\mathbb{D}^{\text{pca}} =& \{(-0.9811, 0.1644)^\top,\ (-0.0336, 0.8465)^\top,\ (-0.2303, 0.4141)^\top,\\ & (-0.6948, 0.6436)^\top,\ (-0.5209, 0.4914)^\top\}.\end{aligned}$$

To further reduce the number of scenarios Krishnamoorthy et al. [93] suggest selecting the discrete realizations from a subset of the PCs that sufficiently explain the total variance in the dataset. Therefore in the example dataset, the discrete realization set will consist of the points corresponding to the maximum and minimum scores along the dominant PC only.

$$\mathbb{D}^{\text{pca}} = \{(-0.9811, 0.1644),\ (-0.0336, 0.8465),\ (-0.5209, 0.4914)\}.$$

However, the worst case is not necessarily included in this case, which will lead to more constraint violations even though the performance will be less conservative. Deciding on the discrete realization set purely on the explained variance in the parameters is generally a bad idea as far as robustness is concerned. It is important to understand how these parameters affect the inequality constraints. Therefore, we suggest using sensitivities in conjunction with PCA to select the discrete realizations from the PCs.

7.8 Scenario selection using both PCA and sensitivities

This is where we motivate the main idea for this chapter. We propose to pick the discrete realizations along the PCs that are likely to cause constraint violations. Let us explain the method with the aid of the simple example above. To determine the effect of the PCs on the constraints check the constraint gradient sensitivities with respect to the transformed parameter $\overline{d}$. The inequality gradients are listed as follows:

$$\begin{aligned}\nabla_{\overline{d}} g_1^\top &= \mathbf{C}\begin{bmatrix}-u_1 & +u_2\end{bmatrix}^\top \\ \nabla_{\overline{d}} g_2^\top &= \mathbf{C}\begin{bmatrix}+u_1 & +u_2\end{bmatrix}^\top\end{aligned}$$

It is difficult to tell by inspection which realizations to pick, so evaluate the sensitivities above at the nominal as follows:

$$\begin{aligned}\nabla_{\overline{d}} g_1 &= \begin{bmatrix}-0.5945 & 1.4988\end{bmatrix} \Rightarrow \overline{d}^{\mathrm{wc}}_{[1]} = \{\overline{d}_1^{\min}, \overline{d}_2^{\max}\} \\ \nabla_{\overline{d}} g_2 &= \begin{bmatrix}1.5931 & -0.2480\end{bmatrix} \Rightarrow \overline{d}^{\mathrm{wc}}_{[2]} = \{\overline{d}_1^{\max}, \overline{d}_2^{\min}\}\end{aligned}$$

We end up with the same number of realizations as the box over-approximation for this example. Still, a reduction in scenarios is guaranteed when there are more variables and constraints considered as is the case in multi-stage MPC. This will be demonstrated in the numerical examples and case study in the following sections. Before describing the application of both PCA and sensitivities for multi-stage MPC, let us briefly describe the sensitivity-assisted multi-stage NMPC.

7.9 Conclusion

This chapter presents simple strategies that are used to select parameter realization values for multi-stage MPC scenario trees. With the aid of a simple example that can be analyzed easily by hand, the chapter compares the conservativeness of the different solutions and the number of realizations selected in each of the approaches. The greater number of realizations or scenarios selected by an approach indicates a greater computational burden when solving the final optimization problem. Therefore, these results demonstrate that the commonly used box over-approximation method is the simplest but computationally inefficient and results in highly conservative performances. A method combining both principal component analysis and NLP sensitivity analysis is proposed to reduce conservativeness and maintain a small computational delay. PCA extracts the correlations in the uncertain parameter space and the sensitivities of the inequality constraints with respect to new unit directions of maximum variance are computed. The analysis of the proposed method shows low conservativeness and a smaller-sized optimization problem that is efficient to solve.

8 | Combining Sensitivity-Assisted Multi-stage NMPC with PCA

"In God we trust; all others must bring data."

William Edwards Deming (1900-1993)

This chapter extends the ideas presented in the previous chapter. The previous chapter recommends analyzing historical process data to select parameter realizations for the scenarios and improve the performance of multi-stage MPC. This chapter aims to combine statistical learning of process data using principal component analysis (PCA) with sensitivity-assisted multi-stage NMPC (SAMNMPC) to achieve a less conservative controller performance.

The chapter is adapted from the following articles.

- A peer-reviewed conference article "Data-driven online scenario selection for multi-stage NMPC" [32], and
- A part of the subsequent journal article: "Sensitivity-based scenario selection for multi-stage MPC along principal components: Applied to robust thermal energy storage operation" [31]

8.1 Motivation

Multi-stage MPC is robust against constraint violations but it is rather conservative, and computationally inefficient resulting in performance loss. Chapter 7 shows that the conservativeness of the solution, and computational efficiency are dependent on how the uncertainty set is represented. So far, its implementation has mainly been done using a hyperbox over-approximation of the uncertainty set. The over-approximation is often very poor if the true uncertainty set is ellipsoidal (i.e. strong correlations exist).

A scenario selection approach aimed at improving computational efficiency in multi-stage MPC is the sensitivity-assisted multi-stage NMPC (SAMNMPC) [20]. Although SAMNMPC is computationally efficient, it overapproximates the uncertainty set leading to a conservative control performance. However, Chapter 7 demonstrated that by combining sensitivity analysis with multivariate data analysis for uncertainty identification, the degree of conservativeness can be significantly reduced. This chapter applies this to multi-stage MPC by performing sensitivity analysis along the PCs for SAMNMPC. An improved prediction of the uncertainty in the overall dynamics of the actual process is expected. Moreover, this chapter aims to demonstrate that the performance of the proposed approach is better than the PCA-based (without sensitivities) approach that assumes the worst-case realization lies on the dominant PC or the first few PCs.

The main contribution of this chapter is to demonstrate how principal component analysis (PCA) can be combined with the sensitivity-assisted multi-stage NMPC (SAMNMPC) framework to reduce conservativeness and retain its computational efficiency. Other specific contributions in this chapter include:

- Improving the existing SAMNMPC algorithm by selecting the parameter realizations for the critical scenarios from PCA.
- Demonstrating with the aid of two numerical examples that the proposed framework is computationally efficient and non-conservative.
- A 12% decrease in peak heating costs when the proposed framework is

implemented for the robust control of thermal energy storage set in a real industrial cluster with district heating operated by Mo Fjernvarme AS, Norway.

- Improving robust constraint satisfaction by reducing the frequency of infeasible problem occurrences.

8.2 SAMNMPC with PCA

This section presents the main idea of this chapter. An integration of SAMNMPC with PCA denoted as PCA-SAMNMPC, is proposed for improved scenario selection. This is to reduce the conservativeness of SAMNMPC and multi-stage MPC in general, by incorporating information from process data in the predetermination of critical scenarios. The aim is to achieve both low computational effort and a low degree of conservativeness in multi-stage MPC for nonlinear systems. The problem formulation is similar to that of the SAMNMPC problem (4.1), but with linearly transformed uncertain parameters. The transformation is based on the function $\Phi^{\mathrm{PCA}}(\,\cdot\,)$ that reconstructs the original variables from the PC scores. After obtaining $\Phi^{\mathrm{PCA}}(\,\cdot\,)$, the optimization problem at time t_k becomes:

$$\begin{aligned}
& V_N^{\mathrm{ps}}(x_k, \overline{\mathbf{d}}^c) = V_N^{\mathrm{sam}}(x_k, \Phi^{\mathrm{PCA}}(\overline{\mathbf{d}}^c)) && (8.1a)\\
& \text{s.t. } z_{i+1}^c = f(z_i^c, \nu_i^c, \Phi_i^{\mathrm{PCA}}(\overline{d}_i^c)), \quad i = 0, \dots, N-1 && (8.1b)\\
& \text{Equations (4.1c) and (4.1d)} && (8.1c)\\
& \Phi_i^{\mathrm{PCA}}(\overline{d}_{i-1}^c) = \Phi_i^{\mathrm{PCA}}(\overline{d}_i^c), \quad i \geq N_R && (8.1d)\\
& z_i^c \in \mathbb{X},\ \nu_i^c \in \mathbb{U},\ \Phi_i^{\mathrm{PCA}}(\overline{d}_i^c) \in \mathbb{D}_i^{\mathrm{pca}}, && (8.1e)\\
& \forall\, c, c' \in \widehat{\mathbb{C}} \cup \{0\}
\end{aligned}$$

where V_N^{ps} is the optimal cost for the SAMNMPC with PCA problem.

8.2.1 PCA-SAMNMPC algorithm

This algorithm requires a dataset $\mathbf{X}$ that contains sampled uncertain parameters. The samples in the dataset can be obtained by measurement, if the parameters are directly measurable, or can be calculated through estimation. This algorithm then applies PCA on the dataset to determine the maximum and minimum scores along the PCs that are to be selected by the SAMNMPC algorithm. To do this, first transform the uncertain parameters in the optimization problem using $\Phi^{\text{PCA}}(\,\cdot\,)$ (see (7.9) and (7.10)). The steps for the algorithm are outlined in Algorithm 3 as follows:

Algorithm 3 PCA-SAMNMPC

```
1: Given: Dataset $\mathbf{X}_k$[1], and $k = 1, \dots, N$;
2: Standardize the data set $\mathbf{X}_k$ to obtain $\mathbf{X}_{k,0}$;
3: Perform PCA on $\mathbf{X}_{k,0}$ to determine the PC scores $\boldsymbol{\Lambda}_k$ and matrix $\mathbf{C}_k$;
4: Transform the discrete realizations $d_i^c$ into the new orthogonal space
   using $\mathbf{C}$, such that, $d_i^c = \Phi_i^{\text{PCA}}(\overline{d}_i^c)$;
5: Substitute the transformation from Step 4 in (4.1) to obtain (8.1) in
   terms of the transformed parameters;
6: At $t_k$, determine critical $\hat{\mathbb{C}}$ and non-critical scenarios $\bar{\mathbb{C}}$ using NLP sensit-
   ivities with respect to the transformed parameters, as in the SAMNMPC
   algorithm;
7: Generate a pruned scenario tree with only the critical and nominal scen-
   arios and then solve (4.1).
```

Algorithm 3 is applied to two numerical examples and a case study for the robust operation of a thermal energy storage (TES) situated in an industrial cluster with district heating demand. The simulation examples are presented and results are discussed in the following sections.

8.3 Numerical examples

This section presents two benchmark numerical case studies to demonstrate the effectiveness of Algorithm 3 in controlling a nonlinear process robustly and with an improved cost.

8.3.1 Example 1 — Van der Pol oscillator

This is a 2-D nonlinear system chosen to show that SAMNMPC with PCA (PCA-SAMNMPC) has small computation times and is not over-conservative. The Van der Pol oscillator system is adapted from [102, 103], and the con-

[1] The dataset can be obtained through direct measurements or estimation. It may be time-dependent such that the dataset must correspond to the MPC time step. If the data is static then $\mathbf{X}_k = \mathbf{X}$ for all k.

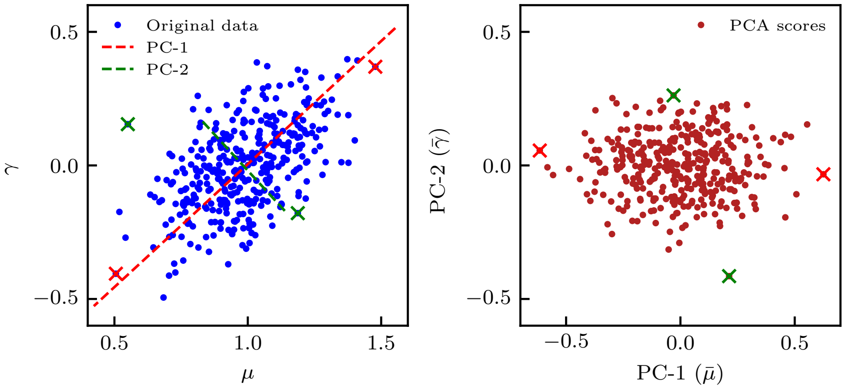

Figure 8.1: Van der Pol — PCA on process data. The left shows the original data, the right shows the PCA scores.

tinuous model is given by the following differential equations:

$$\dot{x}_1 = \mu(1 - x_2^2)x_1 - x_2 + u$$
$$\dot{x}_2 = x_1 + \gamma$$

where $x = [x_1, x_2]^\top$ is the vector of states and u is the control input. The uncertain parameters are μ and γ. A forward Euler method with a sample time $\Delta t = 1\text{s}$ is used to transcribe the model into discrete time. The controller has the following control objective: $\ell = x_1^2 + x_2^2 + u^2$.

Data analysis

Uncertain parameters are randomly generated from an arbitrary multivariate probability distribution of the process to obtain a dataset for controller design. This is done to mimic data acquisition through measurements and estimation from the real system. In this case, the data is historical and is usually pretreated, for example, by removing outliers and filling missing values.

The left plot of Figure 8.1 shows the synthetic process data cloud of the samples. PCA on the dataset gives the following projection matrix:

$$\mathbf{C} = \begin{bmatrix} -0.7356 & 0.6775 \\ -0.6775 & -0.7356 \end{bmatrix}$$

Simulation results.

The simulations are performed in `Julia` [84] environment and using `JuMP v.1.12.0` [83] as the NLP modeler. The NLP solver is `IPOPT 3.14.4` [85], and the linear solver is `HSL-MA97` [86] on a 2.6 GHz Intel Core-i7 with 16 GB memory. At each iteration, a random uncertain parameter is drawn from the probability distribution of the process. The same random uncertain parameter sequence is used in the simulation of each controller for performance comparison. Simulations were performed for both standard NMPC, multi-stage NMPC, SAMNMPC, and PCA-SAMNMPC. Fig. 8.2 is a simulation plot for a robust horizon, $N_R = 2$. The accumulated costs for each robust MPC scheme are summarized in Table 8.1. The accumulated cost of standard NMPC is not compared with the other controllers because it violates constraints. The table presented in Table 8.1 indicates that among all the robust MPC schemes, PCA-SAMNMPC exhibits the lowest accumulated cost. Fig. 8.2 shows that PCA-SAMNMPC tracks closer to the setpoint hence it is less conservative than both SAMNMPC and multi-stage NMPC. Therefore, the transformation of the uncertain parameter realizations in PCA-SAMNMPC improves the tracking performance of multi-stage MPC. The number of scenarios in the multi-stage MPC with a robust horizon of 2 is 81 while Fig. 8.2 shows that SAMNMPC and PCA-SAMNMPC select fewer scenarios, i.e. less than 10. This also implies that SAMNMPC and PCA-SAMNMPC are equally fast and more computationally efficient than the multi-stage MPC.

Table 8.1: Van der Pol — Accumulated costs

NMPC scheme	Accumulated cost
Multi-stage	11.4154
SAM	11.4154
PCA-SAM	7.24489

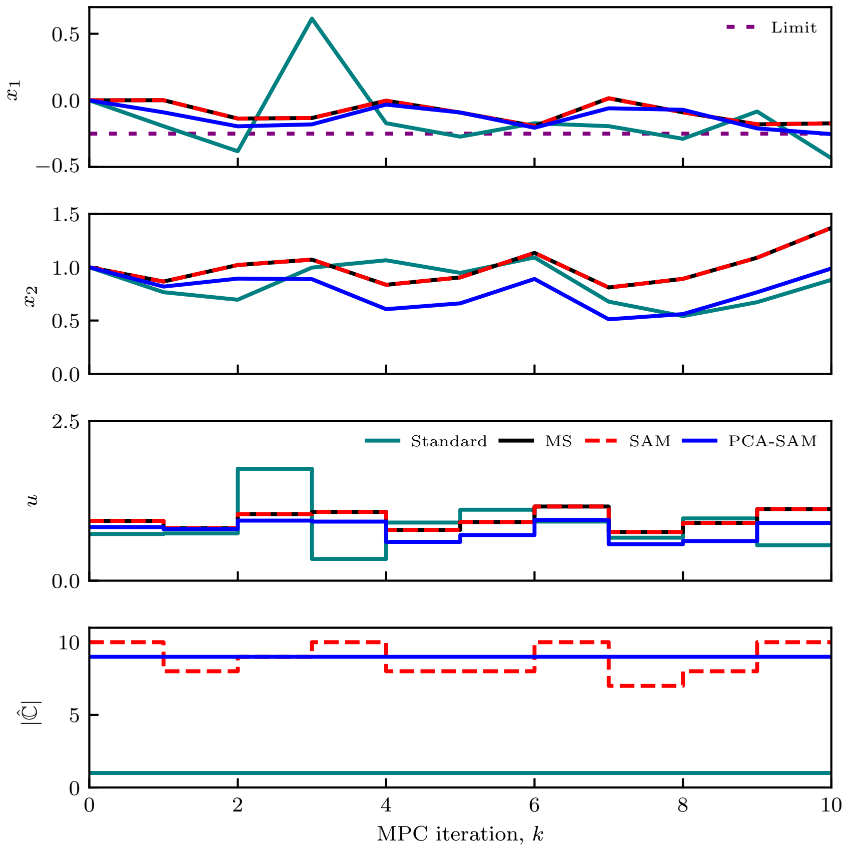

Figure 8.2: Van der Pol oscillator — Comparing the control performances of the PCA-SAMNMPC with standard NMPC, multi-stage NMPC, and SAMNMPC with a $N_R = 2$.

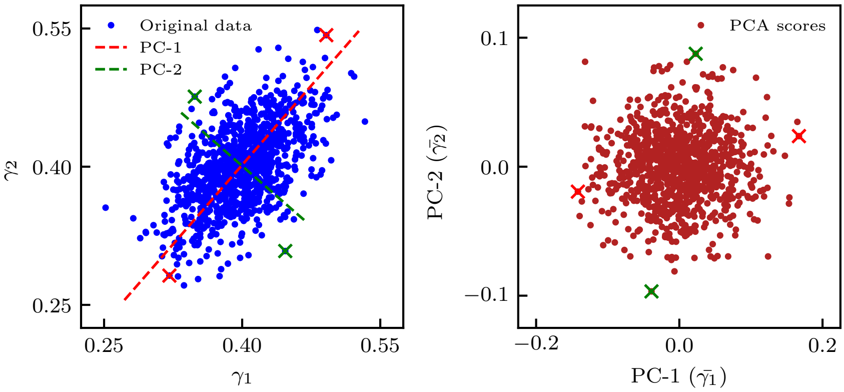

Figure 8.3: Quadtank — PCA on process synthetic dataset; left: original dataset, right: corresponding PC scores.

8.3.2 Example 2 — Quad-tank system

This is the same example as in Section 5.3.2. It is a more complex nonlinear example chosen to demonstrate that PCA-SAMNMPC has fast solution times and is not over-conservative. The quad-tank system is illustrated by Fig. 5.8.

As in the previous part, the system states x_i are the water tank levels, the inputs u_i are pump flow rates, and the uncertain parameters are the valve coefficients γ_1 and γ_2. The water levels in the four tanks are described by a set of differential equations presented in Section 5.3.2. The controller tracks setpoint levels x_1 and x_2 with minimum input usage such that the objective is $\ell = (x_1 - x_1^*)^2 + (x_2 - x_2^*)^2 + r_1 u_1^2 + r_2 u_2^2$. The levels x_3 and x_4 are bounded as shown in Table A.4, and the system experiences predefined pulses in x_1 as presented by Table A.5.

Data analysis

To obtain a dataset for controller design, the parameters γ_1 and γ_2 are random values sampled from an arbitrary multivariate probability distribution of the process. The uncertain parameter samples have a process data cloud shown in the left plot of Fig. 8.3. PCA on the dataset gives the scores plot on the right of Fig. 8.3, where:

$$\mathbf{C} = \begin{bmatrix} 0.6571 & -0.7538 \\ 0.7538 & 0.6571 \end{bmatrix}$$

The dashed lines on the right plot of Fig. 8.3 are the principal component axes where the red line represents the dominant PC (PC-1) and the green line is the second PC (PC-2). The red and green “×” points are the data points corresponding to the extreme scores along each PC.

Simulation results

A software lineup similar to that in the previous example is used except `HSL-MA57` was deployed as the linear solver. The continuous-time differential equations are transcribed into discrete-time using third-order direct collocation with $\Delta t = 10$s. Again, at each iteration, a random uncertain parameter is drawn from the probability distribution of the process. The same random uncertain parameter sequence is used in the simulation of each controller for performance comparison. Simulations are performed for both standard NMPC, multi-stage NMPC, SAMNMPC, and PCA-SAMNMPC. There are done for 150 MPC iterations for robust horizons $N_R = 1$, 2, and 3, and the results are shown in Figs. 8.4 to 8.6, respectively. When the robust horizon is one, in Fig. 8.4 the controllers exhibit similar tracking performances of the level setpoints in tanks 1 and 2 with SAMNMPC slightly deviating from the rest. Inspecting the levels in tanks 3 and 4, it is evident that all the robust controllers back off from the constraint and avoid constraint violations. The standard MPC shows, as expected, 28 instances of constraint violations i.e. $\approx 19\%$ of the operation time.

Figs. 8.5 and 8.6 show that the tracking performance of SAMNMPC degrades when the robust horizon is increased. PCA-SAMNMPC tracks closer to the setpoint, making it less conservative than both SAMNMPC and multi-stage NMPC. Data-driven transformation improves the tracking performance of the SAMNMPC while maintaining robustness against constraint violations of the bounds of x_3 and x_4. This is because the controller selects critical scenarios with parameter realizations that are obtained from

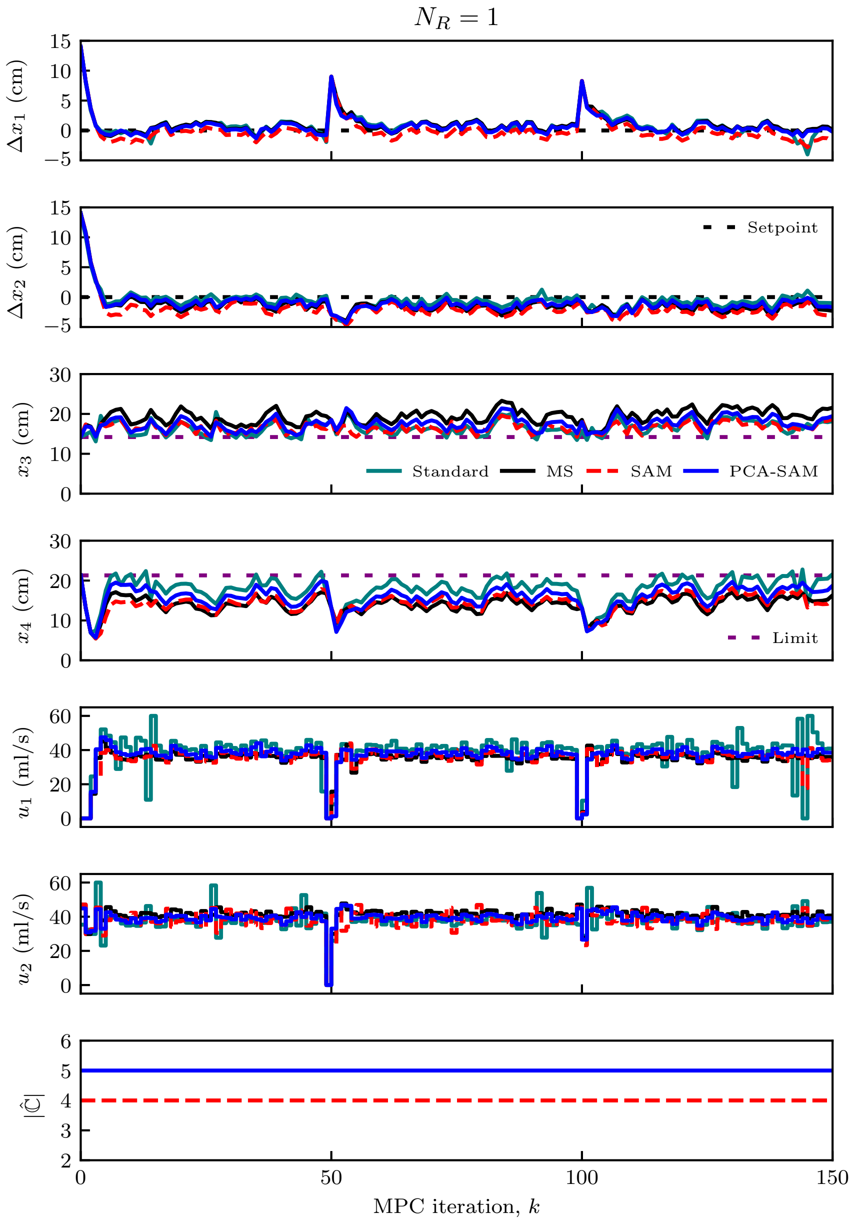

Figure 8.4: Quad-tank — Comparing the control performance of the PCA-SAMNMPC with standard NMPC, multi-stage, and SAMNMPC with $N_R = 1$.

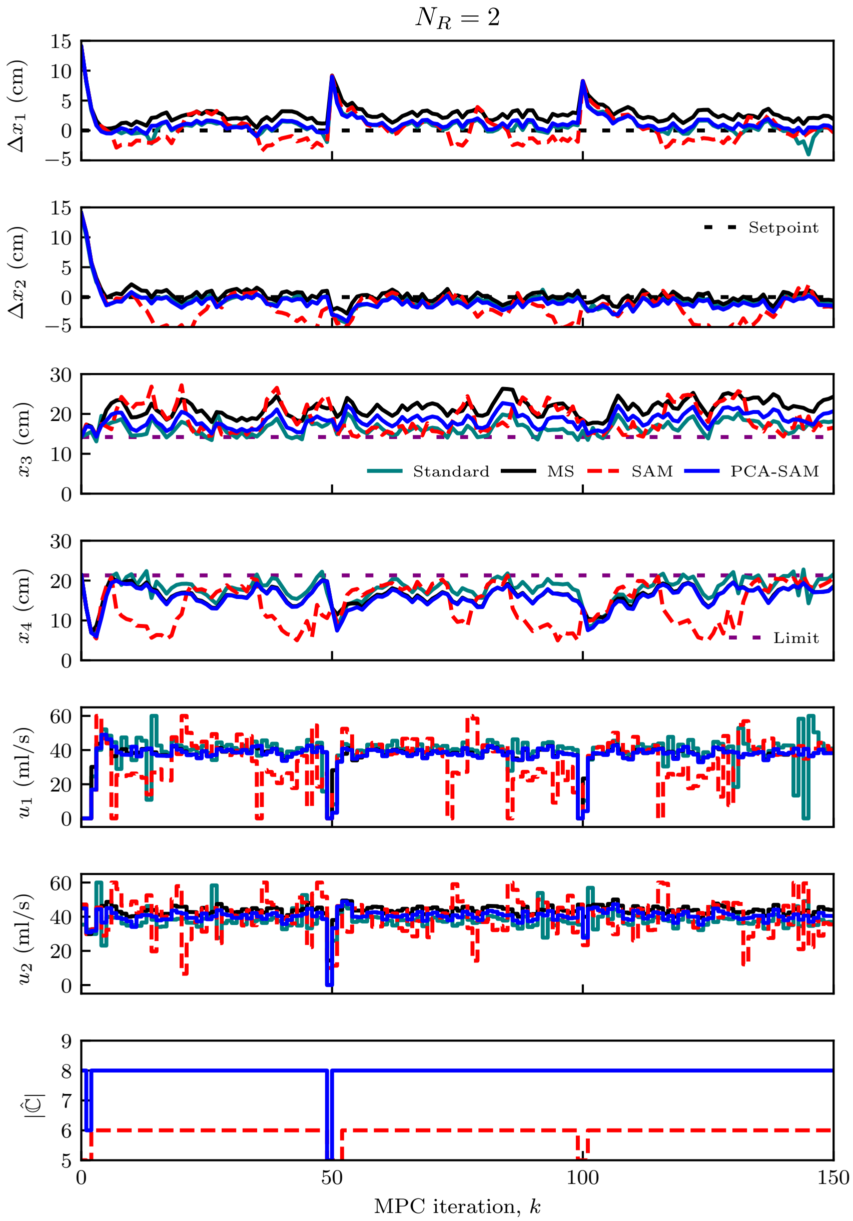

Figure 8.5: Quad-tank — Comparing the control performance of the PCA-SAMNMPC with standard NMPC, multi-stage, and SAMNMPC with $N_R = 2$.

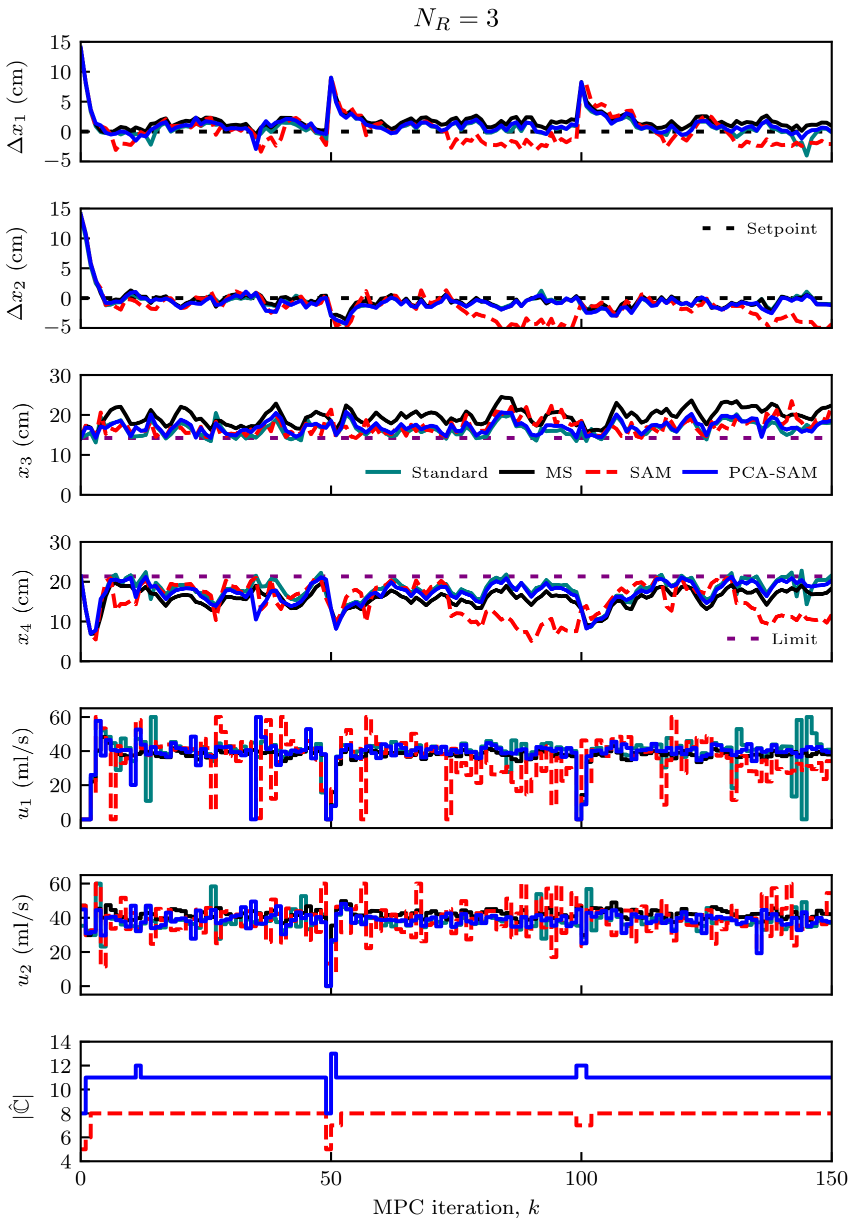

Figure 8.6: Quad-tank — Comparing the control performance of the PCA-SAMNMPC with standard NMPC, multi-stage, and SAMNMPC with $N_R = 3$.

data avoiding over-conservativeness. The SAMNMPC tracking performance is the worst, especially for robust horizons greater than one. This is probably due to the noncritical scenario cost approximation error being larger when the selected realizations lie further away from the nominal parameter. Further, the approximation error is increased when scenarios branch further forward because of a stronger nonlinear cost function with respect to the current control input. In an actual industrial process, this approximation error could be reduced by identifying the uncertain parameters more accurately from historical process data. It can be achieved in this context since data treatment and analysis for identification are all performed offline. Therefore more time could be dedicated to obtaining suitable model parameters with good accuracy.

The number of scenarios in the multi-stage MPC with robust horizons 1,2, and 3 is 9, 81, and 729, respectively while Fig. 8.2 shows that SAMNMPC and PCA-SAMNMPC select fewer scenarios, i.e. less than 5,8, and 13 for robust horizons 1,2 and 3, respectively. Again, this implies that SAMNMPC and PCA-SAMNMPC are equally fast and more computationally efficient than the multi-stage MPC. Fig. 8.7 presents a bar chart that compares mean computation times among the NMPC schemes in the simulations. It shows that PCA-SAMNMPC and the original SAMNMPC have similar solve times that are much lower than multi-stage MPC solve times. For instance when $N_R = 3$, the solve time is above 100 seconds which is impractical. This is because by the time the solution is obtained the tanks will have overflown. On the other hand, SAMNMPC and PCA-SAMNMPC have solution times below 10 seconds that are in the same order as Δt making it practically feasible. Therefore, PCA-SAMNMPC reduces conservativeness and improves the robust performance of SAMNMPC without adding computational complexity.

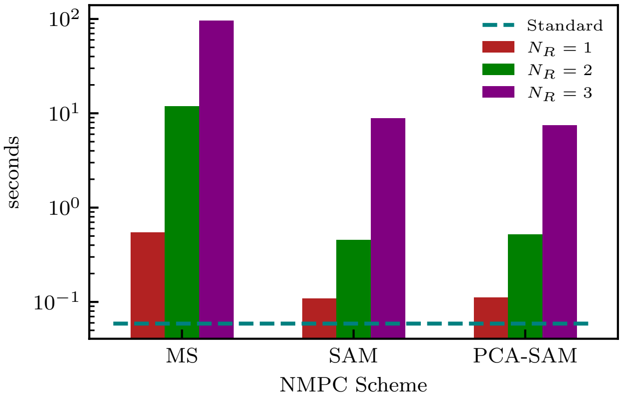

Figure 8.7: Quad-tank — Comparing the average computation time among the different NMPC schemes. Note that computation time has a logarithmic scale.

8.4 Industrial case study

8.4.1 Introduction

This case study examines the advantage of integrating a thermal energy storage (TES) tank as a short-term heat storage, managed by a data-driven multi-stage MPC. The TES tank operated robustly, will be linked to an established district heating network that includes an industrial cluster. The district heating network, situated in Mo Industry Park, generates an annual output of 85–90 GWh, primarily utilizing waste heat extracted from the off-gas produced at the Elkem Rana ferrosilicon plant. While approximately 90% of its yearly heat generation stems from reclaimed waste heat, the intermittent nature of waste heat availability is managed through peak heating boilers. These boilers come into play when the available waste heat falls short of achieving the desired water supply temperature demanded by the city. The peak heating boilers are versatile, operating on either CO-gas or electricity, with the choice determined by the availability of CO-gas and prevailing electricity prices. The flowsheet in Fig. 8.8 illustrates the district heating plant process and the proposed connection of the TES tank to the network. It also shows how water and energy flows are modeled for the system.

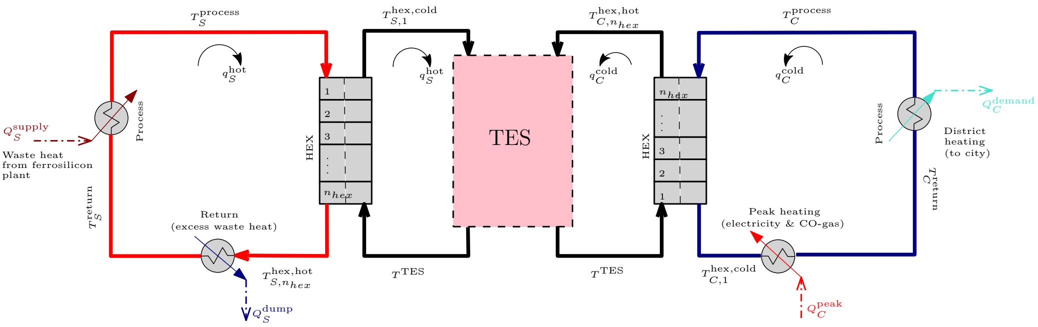

Figure 8.8: Industrial case study — Thermal network flowsheet and model illustration.

The TES stores intermittent industrial waste heat and ensures a consistent temperature of the hot water supplied to the city, accommodating a fluctuating demand profile. This study aims to showcase the benefits of employing PCA-SAMNMPC, utilizing both sensitivities and PCA on historical process data for scenario selection, instead of solely relying on PCA in the PCA-based approaches. In contrast to the two examples presented in the preceding section, the computational time is not investigated. This is due to a 1h time interval between samples providing ample time to find a solution for both multi-stage MPC schemes.

First, PCA-SAMNMPC is implemented to ensure that the TES is operated within bounds under supply and demand uncertainty. Then the control performance is compared to existing PCA with multi-stage MPC heuristic [13] that selects the parameter realizations predominantly from the dominant PC, except during peak heating hours.

8.4.2 System model

TES must be operated to minimize peak heating by making hourly decisions on how much supplied heat must be dumped, and how much peak heating is needed to meet consumer heating and the industrial effluent stream cooling requirements. These control decisions are heavily influenced by uncertainty in heat supply Q_S^{supply} and demand Q_C^{demand}. The system model is derived from mass and energy balances. For a thorough explanation of the process modeling, refer to §2.1 in [13]. The final set of model equations and the model parameter values are presented in Appendix A.3.

The hot and cold sides of the heat exchangers are modeled by $n_{\text{hex}} = 5$ cell discretizations. Therefore, the dynamic model consists of 25 states. All the states are temperatures and can be categorized into three.

- Supplier-side temperatures:

$$\mathbf{T}_S = [T_S^{\text{process}}, T_S^{\text{return}}, T_{S,j}^{\text{hex,hot}}, T_{S,j}^{\text{hex,cold}}]^\top,$$

- Consumer-side temperatures:

$$\mathbf{T}_C = [T_{C,j}^{\text{hex,hot}}, T_{C,j}^{\text{hex,cold}}, T_C^{\text{process}}, T_C^{\text{return}}]^\top, \text{ and}$$

- Thermal storage temperature: T^{TES}

where $j = \{1, \ldots, n_{\text{hex}}\}$. The control inputs vector u has three variables $u = [q_C^{\text{hot}}, Q_S^{\text{dump}}, Q_C^{\text{peak}}]^\top$ which are the consumer-side volumetric flow rates of the storage fluid, the heat dump rate for supplier, and the peak heating rate for consumer, respectively.

The system must be operated within process bounds, for example, to avoid boiling of storage liquid. Furthermore, there are return stream requirements from the industrial processes and district heating on either side of the heat network. Constraints are also enforced on the volumetric flows to avoid low flows that risk fouling in heat exchangers. There exist bounds on the heat dumping and peak heating rates due to heat transfer limitations.

The problem has an economic objective to minimize peak heat usage in the industrial cluster. Regularization terms are added to the stage cost at t_{k+i} and the stage cost is written as:

$$\ell_i = r_1 Q_{C,i}^{\text{peak}} + r_2 \left(Q_{S,i}^{\text{dump}}\right)^2 + r_3 \left(q_{C,i}^{\text{hot}}\right)^2 \tag{8.2}$$

where $r_1 = 1.0$, and $r_2 = 0.01$ and $r_3 = 10^{-5}$ are the coefficients of the regularization terms.

Let us define the process constraints that are included in the optimization problem. All the storage fluid flows $q_{C,i}$ are supposed to be between 0.001 m^3/s and 0.3 m^3/s. The supply side is assumed to generate heat through a batch cooling process, and it is essential to ensure that the return temperature $T_S^{\text{return}} \leq 85$ °C, to maintain optimal product quality in the batches. Moreover, the district heating network requires the return temperature $T_C^{\text{return}} \geq 60$ °C. Violation of the temperature constraints may result in economic penalties for the supplier or consumer involved. For an exhaustive list of the process constraints, refer to Table A.7.

8.4.3 Data analysis

The system model assumes that there is one source and one sink that exchange heat through the TES unit. There is also one peak heating source. Therefore, total waste heat, total peak heating, and total heat demand are computed for each hour to obtain the raw dataset.

Data description. The hourly heat supply and demand datasets from the years 2017 and 2018 are provided by the district heating company, Mo Fjernvarme AS. Data from 2017 is used as the training set, and the test set is data from 2018. The model does not focus on the individual units but on the total heat demand and supply to demonstrate the performance of our proposed control strategy. Further, we only consider data from the winter months (i.e. December, January, February, and March) when having a TES is relevant. The winter months have the highest heat demand profiles and the heat supply is not always higher than demand creating a need for peak heating. Hence, a need for peak heat savings using optimal TES operation.

PCA on data. The training set consists of hourly demand and supply for 121 days. Before performing PCA, the training set is visualized and outliers are detected and cleaned. The scatter plots for each hour from the training set are plotted in Fig. 8.9. There is a general trend in the hourly data that a strong correlation exists between demand and supply except during the peak heating hours (scatter plots in red). During peak heating hours, the demand is usually higher than the supply from waste heat and there is no

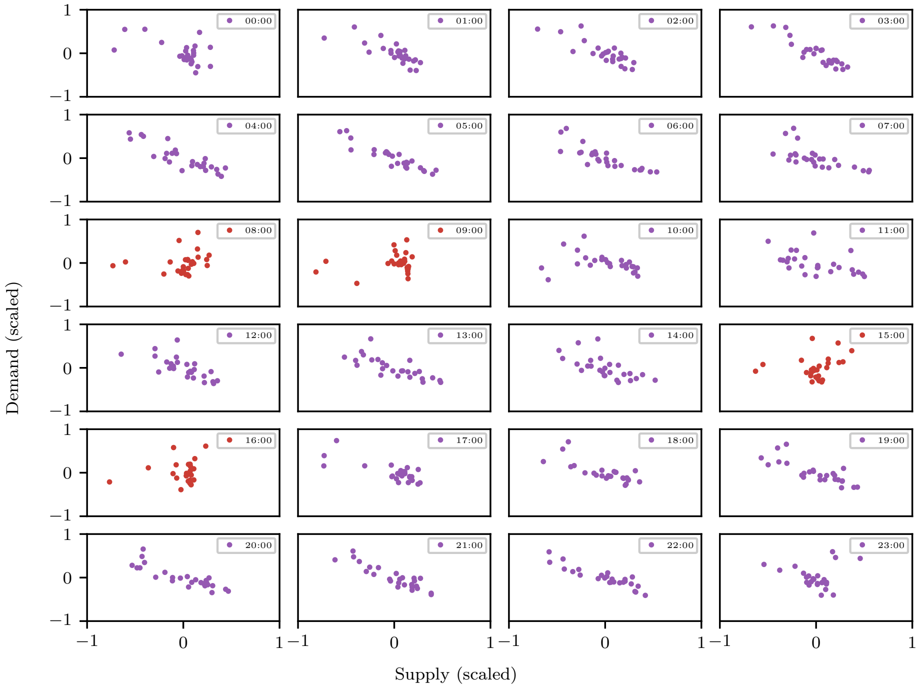

Figure 8.9: Industrial case study — Scatter plots of scaled supply and demand data for January 2017 to illustrate data directionality. Notice that there is no clear directionality during the peak heating hours (scatter plots in red).

clear correlation in the data. Outlier detection is performed on the raw dataset and the outliers are removed to obtain a clean training set. Further detail on the outlier detection is explained in Appendix C of [13].

The clean training set is standardized as shown in (7.7). Then PCA is performed on each $\mathbf{X}_{\mathbf{0},k}$ by singular-value decomposition (SVD), where $k = 0, \ldots, N-1$, and $N = 24$. The columns of $\mathbf{X}_{\mathbf{0},k}$ are the supply and demand values, in that specific order. Each row of $\mathbf{X}_{\mathbf{0},k}$ represents a day in the winter months with normal operation (i.e. a non-outlier).

The PCA projection matrices at the k-th hour, $\mathbf{C}_k$, are determined for each $k \in \{0, \ldots, N-1\}$. If the matrices $\mathbf{\Sigma}_k = \text{diag}([\sigma_S, \sigma_C])$ are the diagonal matrices of the standard deviations for each row in $\mathbf{X}_k$, then the original model parameters can be reconstructed from the PC scores as follows:

$$d_k = \mathbf{\Sigma}_k \mathbf{C}_k \overline{d}_k + d_k^0 \tag{8.3}$$

for all $k \in \{0, \ldots, N-1\}$, where $\overline{d}_k$ are the transformed model parameters.

8.4.4 Simulation results

The prediction horizon is $N = 24$ with a $\Delta t = 1\text{h}$ sample time. Simulation results for the PCA-SAMNMPC and PCA with multi-stage MPC applied to the TES system are compared. Since there is ample time between samples, there is no concern about the computational speed. The main goal is to achieve constraint satisfaction with reduced conservativeness (i.e. less peak heating).

At each MPC iteration, the scenario tree of the multi-stage MPC is designed based on the parameters identified from the PCA corresponding to that hour in Fig. 8.9. When the 24th hour ends, the hour count is reset, and the scenario tree of the subsequent MPC problem is assembled using data from 00:00 again. For the case of real-time continuous implementation, this cycle is repeated every new day i.e. every 24 hours. In this section, the simulations are run for one day and are all assumed to start from a common initial condition. Outlining initial conditions, the process temperatures,

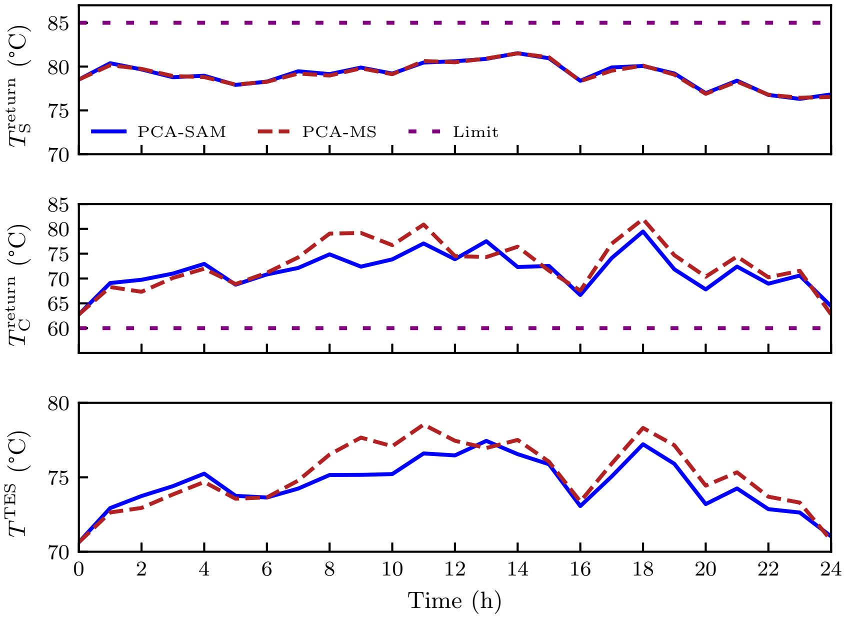

Figure 8.10: Industrial case study — The return and tank temperature profiles in the PCA with SAMNMPC (PCA-SAM) and PCA with multistage MPC (PCA-MS) formulations, for January 6, 2018.

T_S^{process} and T_C^{process}, start at 91.28 °C and 50 °C, respectively. The initial temperature within the tank, T^{TES} is 70.63 °C. These initial values result from a steady-state optimization of the system with mean heat supply and demand profiles.

The supply and demand profiles on January 6th, 2018 were used to compare the control performances of the PCA-SAMNMPC and PCA with multi-stage MPC. The scenarios were selected based on the training dataset. Fig. 8.10 illustrates the trajectories of the temperatures of the supplier return T_S^{return}, consumer return T_C^{return} and storage T^{TES} for the two multistage MPC schemes given the constraints on the return temperatures. Both the PCA-SAMNMPC and PCA with multi-stage NMPC show no constraint violations in the return temperature profiles, and both have identical tank temperature profiles. The return temperature profiles for PCA-SAMNMPC are slightly lower than in PCA with multi-stage MPC, especially during the peak heating hours. Moreover, PCA-SAMNMPC has lower operation costs than PCA with multi-stage MPC because it has lower peak heat usage Q_C^{peak} as it can be seen in Fig. 8.11. While PCA with multi-stage MPC uses a total peak heating of 124.68 MWh, PCA-SAMNMPC results in 112.65 MWh peak heating during the day. Therefore, the conservativeness of the PCA with multi-stage MPC is further reduced by including sensitivity analysis for scenario selection. The value of q_C^{hot} is always at the maximum bound i.e. 0.3 m^3/s, throughout all the simulations. This is because the system requires the transfer of large available amounts of heat as quickly as possible to the demand side through the TES. Since q_C^{hot} is always constant and with an active constraint, it is therefore not discussed further in the results.

Further simulations for each day in January 2018 were done, employing their respective supply and demand profiles. The results reveal consistent success in maintaining the system within the predefined limits for both schemes. Fig. 8.12 illustrates the return temperature profiles of the supplier and consumer throughout all days in January 2018, showcasing the effectiveness of the two multi-stage MPC schemes in keeping the uncertain process within bounds. Notably, the simulations demonstrate the applicability of the two multi-stage MPC schemes, even when the initial heat supply is lower than the heat demand, as observed on multiple days in January 2018.

The average daily use of peak heating across all days in January 2018 with PCA multi-stage NMPC was found to be 108.38 MWh, whereas, for PCA-SAMNMPC, it was 96.78 MWh. This is a 12% reduction on the monthly average peak heating cost. Considering this as a monthly average, it significantly reinforces the conclusion drawn from the daily simulation, indicating that PCA-SAMNMPC is less conservative than PCA with multi-stage MPC. Moreover, the simulations resulted in infeasible problems for PCA-SAMNMPC on 2 days in January 2018 (9th and 12th) while PCA with multi-stage NMPC exhibited that on 4 days in January 2018 (9th, 11th, 12th, and 14th). Therefore the expected frequency of constraint violations is greater for PCA multistage than for PCA-SAMNMPC. This is because PCA-SAMNMPC picks the scenarios along the PC that affect the con-

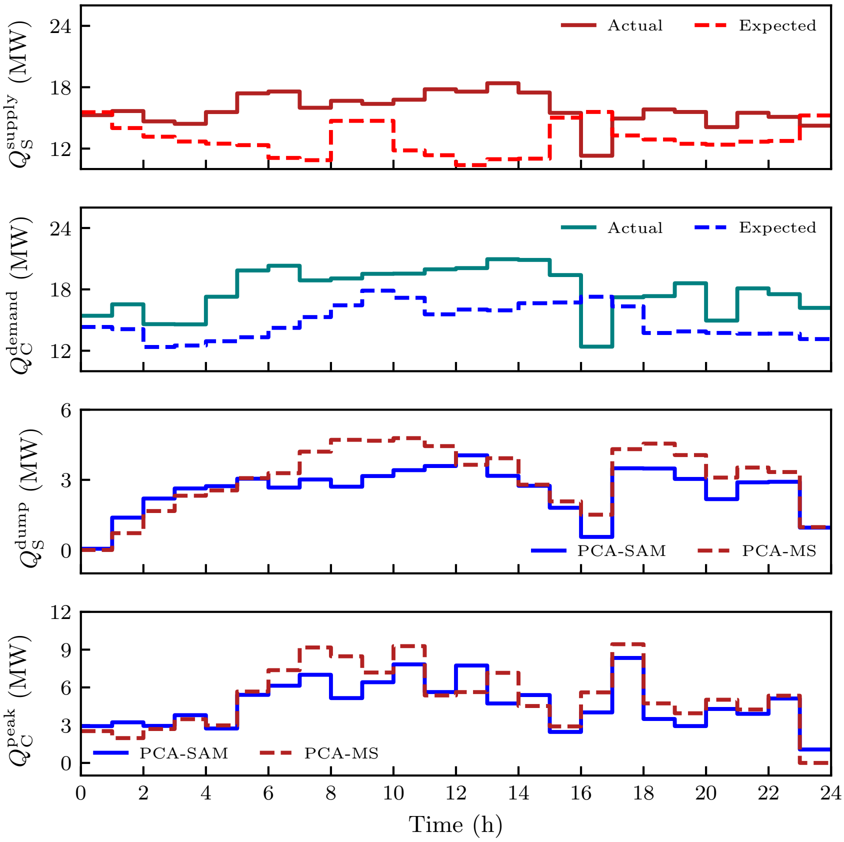

Figure 8.11: Industrial case study — The heat supply and demand profiles; and the corresponding heat dumping and peak heating profiles obtained from PCA with multistage MPC and PCA with SAMNMPC formulations for January 6, 2018. The expected profile is the nominal scenario and is the mean demand and supply profile for Jan.-Mar. and Dec. 2017

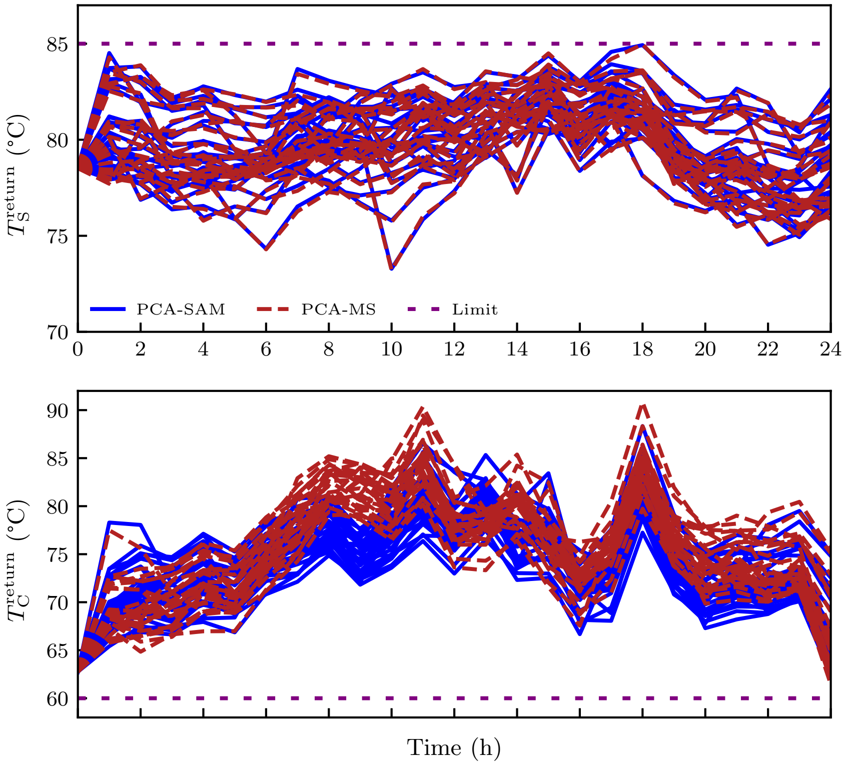

Figure 8.12: Industrial case study — The supplier and consumer return temperature profiles for the whole month of January 2018; both schemes have no constraint violations and they keep the temperatures within bounds.

straints.

8.5 Conclusion

This chapter integrates PCA and NLP sensitivity analysis to propose the PCA-SAMNMPC algorithm that improves the performance of SAMNMPC [20]. First, PCA on historical process data extracts correlations in the uncertain model parameter space and then computes sensitivities of the inequality constraints with respect to the principal components. The parameter realizations that lie in these new unit directions that will cause constraint violations are identified and used to construct the scenario tree for multistage MPC. The proposed strategy is applied to two numerical case studies and shows fast solution times with the smallest tracking error.

Moreover, this chapter investigates the performance of this strategy on the robust control of a hot water thermal energy storage unit in a district heating network. The proposed method is compared to the PCA-based approach in [13] that does not use sensitivities. The simulation results show that the proposed method improves the average economic performance. There is a 12% reduction in peak heating requirement and a 50 % reduction in the frequency of infeasible problems. Hence, in addition, an improved robustness compared to the data-driven heuristic in [13]. This is because the NLP sensitivities assist in updating parameter realizations of the critical scenarios, especially during peak heating hours when the demand and supply correlation is the weakest.

Conclusions & Future Work

9 Conclusions & Future Work

*"In nature's infinite book of secrecy
A little I can read."*

William Shakespeare (1564-1616)

This chapter summarizes the content of this thesis and presents key findings, limitations of the study, and suggestions for future work.

9.1 Summary of findings

This thesis examines the robust multi-stage MPC framework and presents new ideas to improve it for a swifter practical application in nonlinear systems control. The novel approaches aim to mitigate the limitations of the multi-stage MPC formulation while maintaining its good qualities, especially robust constraint satisfaction. The addressed limitations include computational efficiency and scenario selection strategy. Hence, this thesis is divided into two major parts.

In both parts of the thesis, nonlinear optimization theory and sensitivity analysis are employed to improve computational efficiency. The first part shows that the use of nonlinear optimization theory and sensitivity analysis can be used together with nonlinear systems stability theory to continuously adapt the length of the prediction horizon. Consequently, the framework progressively reduces the prediction horizon as the controlled system approaches the optimal steady state, and the scenario tree shrinks in size resulting in a computationally fast and robustly stable performance. The second part demonstrates that the use of multivariate data analysis is prudent for assigning the parameter realizations assembled in the scenario tree. This part also demonstrates that the combination of multivariate data analysis and sensitivity analysis to assemble the scenario tree results in further improvement in robustness and fast computational performance of multi-stage MPC.

The following is an outline of the findings from each chapter in the main parts of the thesis. Part I begins with Chapter 5 that motivates the need for multi-stage MPC approximation to improve its computational efficiency and facilitate online implementation.

The major contributions of this chapter are listed as follows:

- Proposed an adaptive horizon multi-stage MPC framework to achieve fast computation times.
- Proposed the algorithm for an online update of the prediction horizon for multi-stage MPC with a reference tracking objective.
- Proposed workflows to determine the terminal ingredients (terminal region and cost) of multi-stage MPC with robust horizon and with a fully branched scenario tree.
- Demonstrated an improved computational efficiency using simulations on two numerical examples.

In Chapter 6, the stability properties of the previously proposed adaptive horizon multi-stage MPC are presented and discussed. The conditions for recursive feasibility are outlined based on achieving robust constraint satisfaction of the closed-loop system. Further, conditions for robust stability of the proposed framework are established by proving regional input-to-state practical stability (ISpS).

The following are the main contributions of this chapter:

- Proved recursive feasibility of the adaptive horizon multi-stage MPC with a fully branched scenario tree.

- Showed that the adaptive horizon multi-stage MPC with a robust horizon is not recursively feasible but using the relaxed formulation avoids problem infeasibility.
- Proved ISpS of the adaptive horizon multi-stage MPC.
- Demonstrated the control performance of the adaptive horizon multi-stage MPC with a fully branched scenario tree that is guaranteed ISpS with the aid of a numerical example and showed robustness and a significant reduction in computational time.

Chapter 7 introduces Part II by motivating the importance of scenario selection techniques to improve the robust control performance of multi-stage MPC. This chapter elucidates the advantages of data-driven scenario selection because it gives a better representation of uncertainty. extracted. The uncertainty representation is extracted from large correlated data sets using PCA that uncovers directions of maximum variability. Parameter realizations that may result in constraint violations are expected to lie on the extremes of these directions known as principal components. PCA can result into a dimensionality reduction leading to fewer selected realizations, and consequently a reduced scenario tree size. Furthermore, with the aid of a simple optimization problem, this chapter shows that the actual propagated effect of these parameter realizations on the constraints is obtained via nonlinear programming sensitivity analysis. A further reduction in the size of the optimization problem size is shown with assistance from NLP sensitivity analysis. Hence, a combination of PCA and NLP sensitivity analysis results in robustness with reduced conservativeness and computational time. The main contributions of Chapter 7 include:

- Proposed a scenario selection strategy that is based on combining PCA and NLP sensitivity analysis.
- Compared the conservativeness and problem size of the conventional box over-approximation scenario selection approach with PCA-based scenario selection with the aid of a simple nonlinear problem.
- Compared the conservativeness and problem sizes of the PCA only,

and the PCA with NLP sensitivity analysis scenario selection approaches with the aid of the same numerical example.
- Demonstrated that evaluation of sensitivities along principal components from data results in the selection of less conservative parameter realizations.

Chapter 8 demonstrates and compares the practical implementation of the proposed scenario selection approaches from Chapter 7 for robust control using multi-stage MPC. A brief background of the computationally efficient sensitivity-assisted multi-stage NMPC (SAMNMPC) is given with its drawback of over-conservativeness that is a result of poor scenario selection. This chapter assumes that model parameters are usually strongly correlated, and can be sampled or estimated, even at a later time, to generate process data. The process data is analyzed using PCA to compute the principal components of the uncertainty space. Then the SAMNMPC formulation is expressed in terms of the principal components, a linear transformation of the original parameters. This approach is tested by simulations on two benchmark numerical examples with a synthetic process data set. In both cases, it is computationally fast and the controlled system is maintained within process limits but there is a significant reduction in conservativeness.

In addition, the proposed framework was implemented on a detailed case study with actual process data to control thermal energy storage for a district heating network under supply and demand uncertainty. The approach reduced the required average peak heating when compared to the scenario selection approach based on PCA only. In this case, the approach cuts the frequency of infeasible problems by half. Hence, the integration of PCA and NLP sensitivity analysis for scenario selection in multi-stage MPC improves robustness and reduces conservativeness and computational delay. The major contributions in this chapter are outlined below:

- Proposed the PCA-SAMNMPC framework that combines PCA and NLP sensitivities for scenario selection in multi-stage MPC.
- Compared the performances of the PCA-SAMNMPC approach with SAMNMPC, multi-stage MPC, and standard MPC on two numerical

examples to demonstrate robust constraint handling, computational speed, and low conservativeness.

- Demonstrated the practical advantage of implementing the less conservative PCA-SAMNMPC using a realistic case study where an economic benefit is realized by a reduced peak heating requirement.

9.2 Limitations of this study

A general limitation of this thesis stems from the fact that all the proposed approaches are model-based. This is a strong limitation because the key to a satisfactory control performance is model accuracy. It is challenging to obtain good models, and they are seldom relevant for the full process domain. Therefore, the methods will require strong expertise to ensure good models are obtained for all the operating points or domains.

The thesis focuses on handling parametric uncertainty only, and that assumes a correct model structure. No guarantees can be made if the model structure is different and naturally, this may occur in practice. Since this is a theoretical study, the methods assume that full-state information is always available, and acquired without any delay. This must be accounted for, if the proposed methods are to be applied to a real plant. The following are specific limitations to each of the proposed algorithms.

Adaptive horizon multi-stage MPC. This approach aims to reduce the scenario tree size by shrinking the prediction horizon using an adaptive horizon update. This should be done while also maintaining the robustness of multi-stage MPC and closed-loop stability. The proposed framework causes a reduction in problem size only when the controlled system is approaching a stable optimal equilibrium. If the controlled system is continually disturbed such that it is steered away from its equilibrium, then no computational benefit will be realized. Moreover, there is always a significant computational delay at the first iteration where the problem must be solved with a long prediction horizon.

PCA-SAMNMPC framework. This approach relies on the availability of a large data set with uncertain parameter samples. It requires that the system has model parameters that are measurable or can be estimated accurately from other measurements. This can be the case if these parameters are process disturbances for example source and sink temperatures. If the model parameters cannot be measured, and are to be estimated based on the model, for example using the moving horizon estimator, then there is an increased dependency on the correctness of the model structure. This may lead to an unexpected loss of performance of the proposed approach. Moreover, the method selects scenarios based on sensitivity analysis by assuming monotonic constraints with respect to uncertain parameters. The robustness of the controller might break if the constraints do not satisfy this property.

9.3 Recommendations for future work

In this section, several directions for future work on the topics of this thesis are presented. The recommendations include direct extensions of this work and suggestions driven by certain limitations of the proposed methodologies.

Horizon update in multi-stage MPC with economic costs. In Part I of the thesis a horizon update scheme for the multi-stage MPC with reference tracking objective was presented. A direct extension of that is to examine the performance of the proposed adaptive horizon multi-stage MPC with economic costs. Then a theoretical analysis similar to [79] may be done to investigate the conditions for recursive feasibility and robust stability properties.

Advanced step strategy at the first iteration. The adaptive horizon multi-stage MPC framework does not reduce computational delay at the initial MPC interval. This is due to the need to solve the problem with a long horizon at the start and obtain sensitivity-assisted predictions that decide to shrink the horizons of subsequent iterations. To circumvent this problem, an advanced step MPC strategy such as in [18, 104] may be integrated to shift the computational burden away from the first interval. Another possibility is to perform an advanced step with a horizon update to obtain solutions in real time.

Inverse optimal control to shrink horizons. To solve the issue of significant computational delay at the beginning of adaptive horizon multi-stage MPC implementation, an inverse optimal control approach similar to that in [105] may be applied to provide a computationally efficient approximation. The method finds a suitable objective function that makes a shorter prediction horizon problem approximately equal to a complex problem with a long prediction horizon. The approach first involves collecting closed-loop state-input pairs and then fitting a small-sized problem that gives the same control policy with approximate optimality.

Synergizing scenario selection and horizon update using sensitivity analysis. Another direct extension is to use sensitivity analysis to simultaneously select scenarios and reduce the prediction horizon. This might require performing an uneven reduction of the prediction horizon. That is, some scenarios will end up becoming longer or shorter than others, causing a software implementation challenge. In this case, one must rethink to find a more efficient methodology for the algorithm. This is because directly integrating the two algorithms implies solving both nominal and multi-stage MPC with long horizons in one interval, increasing the computational burden.

Scenario selection using other data analysis techniques. The thesis has extensively examined the PCA-based approach to obtain parameter realization for the multi-stage MPC scenario tree. Further investigation could be made on the feasibility and robust performances of other multivariate data analysis techniques to identify scenarios from large data sets.

Validation with practical implementation. All the proposed methods in this thesis are tested using simulation experiments. The next step may be to implement the strategies on a laboratory-scale plant process. This will help discover and uncover the limitations and advantages of these methods for real-time MPC implementation. Hence, further contributing to bridging the gap between theory and practice.

Appendix

A | Supporting Information

A.1 Cooled CSTR

The following Table A.1 shows the parameters for the cooled CSTR example used in this thesis. The following Table A.2 presents the limits on the state and control input variables for the cooled CSTR example.

Table A.1: CSTR — System parameters

Parameter	Value	Units
$A_{1,2}$	9.043×10^{12}	/h
A_3	9.043×10^{9}	/h
$E_{1,2}/R$	9758.3	K
E_3/R	8560.0	K
ΔH_{AB}	4.2	kJ/mol
ΔH_{BC}	−11.0	kJ/mol
ΔH_{AD}	−41.85	kJ/mol
c_p	3.01	kJ/kgK
$c_{p,J}$	2.0	kJ/kgK
ρ	0.9342	kg/m^3
A_R	0.215	m^2
V_R	10.01	m^3
T_{in}	130.0	°C
k_w	4032	kJ/hm^{2}K
m_J	5	kg
R	8.314×10^{-3}	kJ/Kmol

Table A.2: CSTR — Bounds on states and inputs

Variable	Initial condition	Minimum	Maximum	Unit
c_A	0.8	0.1	5.0	mol/l
c_B	0.5	0.1	5.9	mol/l
T_R	134.14	50	140	°C
T_J	134.0	50	180	°C
F	18.83	0.0	35	/h
$\dot{Q}_J$	−4495.7	−8500	0	kJ/h

A.2 Quad-tank system

The following Table A.3 summarizes the model parameters for the quadtank example used in this thesis. The following Table A.4 presents the limits on the state and control input variables for the quadtank example. The following Table A.5 presents the pulse changes introduced on the tank levels for the quad-tank example.

Table A.3: Quad-tank — Model parameters

Parameter	Value	Unit	Parameter	Value	Unit
A_1	50.27	cm^2	a_1	0.233	cm^2
A_2	50.27	cm^2	a_2	0.242	cm^2
A_3	28.27	cm^2	a_3	0.127	cm^2
A_4	28.27	cm^2	a_4	0.127	cm^2
γ_1	0.4 ± 0.05	–	γ_2	0.4 ± 0.05	–

Table A.4: Quad-tank — Bounds on states and inputs

Variable	Minimum	Maximum	Unit
x_1	7.5	28.0	cm
x_2	7.5	28.0	cm
x_3	14.2	28.0	cm
x_4	4.5	21.3	cm
u_1	0.0	60.0	ml/s
u_2	0.0	60.0	ml/s

Table A.5: Quad-tank — Pulse changes to state variables

k	x_1	x_2	x_3	x_4
0	28 cm	28 cm	14.2 cm	21.3 cm
50	28 cm	14 cm	28 cm	21.3 cm
100	28 cm	14 cm	14.2 cm	21.3 cm

A.3 Optimal Control Problem for the TES System

The complete formulation of the dynamic optimization problem for the industrial park system with a TES tank is as follows:

Energy cost

$$\textbf{min} \quad \int_{t_0}^{t_f} \left(\sum_{j=1}^{n_C} Q_{C_j}^{\text{peak}}(t) \right) dt$$

s.t.

Supply side balances

$$\dot{\mathbf{T}}_{S_i}(t) = \mathbf{f}_{S_i}\big(\mathbf{T}_{S_i}(t), \mathbf{q}_{S_i}(t), Q_{S_i}^{\text{supply}}(t), Q_{S_i}^{\text{dump}}(t), Q_{S_i}^{\text{TES,in}}(t)\big)$$

$$Q_{S_i}^{\text{TES,in}}(t) = \sum_{k=1}^{n_{\text{hex}}} (UA)^{\text{hex}} (T_{S_i,k}^{\text{hex,hot}}(t) - T_{S_i,k}^{\text{hex,cold}}(t))$$

Consumer side balances

$$\dot{\mathbf{T}}_{C_j}(t) = \mathbf{f}_{C_j}\big(\mathbf{T}_{C_j}(t), \mathbf{q}_{C_j}(t), Q_{C_j}^{\text{demand}}(t), Q_{C_j}^{\text{peak}}(t), Q_{C_j}^{\text{TES,out}}(t)\big)$$

$$Q_{C_j}^{\text{TES,out}}(t) = \sum_{k=1}^{n_{\text{hex}}} (UA)^{\text{hex}} (T_{C_j,k}^{\text{hex,hot}}(t) - T_{C_j,k}^{\text{hex,cold}}(t))$$

Tank balance

$$\rho c_p V^{\text{tank}} \dot{T}^{\text{TES}}(t) = \sum_{i=1}^{n_S} Q_{S_i}^{\text{TES,in}}(t) + \sum_{j=1}^{n_C} Q_{C_j}^{\text{TES,out}}(t) - Q^{\text{TES,loss}}(t)$$

$$Q^{\text{TES,loss}}(t) = (UA)^{\text{tank}} \big(T^{\text{TES}}(t) - T^{\text{amb}}(t)\big)$$

Initial state constraints

$$\mathbf{T}_{S_i}(0) = \mathbf{T}_{S_i}^{\text{init}}$$

$$\mathbf{T}_{C_j}(0) = \mathbf{T}_{C_j}^{\text{init}}$$

$$T^{\text{TES}}(0) = T^{\text{TES,init}}$$

Operating constraints - states

$$\mathbf{T}_{S_i,\text{min}} \leq \mathbf{T}_{S_i}(t) \leq \mathbf{T}_{S_i,\text{max}}$$

$$\mathbf{T}_{C_j,\text{min}} \leq \mathbf{T}_{C_j}(t) \leq \mathbf{T}_{C_j,\text{max}}$$

$$T_{\text{min}}^{\text{TES}} \leq T^{TES}(t) \leq T_{\text{max}}^{\text{TES}}$$

Operating constraints - inputs

$$\mathbf{q}_{C_j,\text{min}} \leq \mathbf{q}_{C_j}(t) \leq \mathbf{q}_{C_j,\text{max}}$$
$$Q^{\text{dump}}_{S_i,\text{min}} \leq Q^{\text{dump}}_{S_i}(t) \leq Q^{\text{dump}}_{S_i,\text{max}}$$
$$Q^{\text{peak}}_{C_j,\text{min}} \leq Q^{\text{peak}}_{C_j}(t) \leq Q^{\text{peak}}_{C_j,\text{max}}$$

Indices

$$\forall i \in \{1, \dots, n_S\}, \forall j \in \{1, \dots, n_C\}, \forall k \in \{1, \dots, n_{\text{hex}}\}$$

where the parameters and their values are presented in the following Table A.6. The following Table A.7 presents the bounds on the state and control input variables for the industrial thermal energy storage case study.

Table A.6: TES — System parameters

Symbol	Value	Unit
V^{tank}	1000	m^3
$(UA)^{\text{tank}}$	0.1	kW/K
V^{hex}	1	m^3
A^{hex}	2500	m^2
U^{hex}	1.2	$\text{kW/m}^2\text{K}$
n_{hex}	5	-
T^{amb}	-5	°C
ρ	1000	kg/m^3
c_p	4.18	kJ/kgK
q_S^{hot},q_S^{cold}	0.3	m^3/s

Table A.7: TES — Bounds on states and inputs

Variable	Minimum	Maximum	Unit
T_S^{process}, T_C^{process}	50	100	°C
T_S^{return}	50	85	°C
T_C^{return}	60	100	°C
T^{TES}	50	100	°C
q_C^{hot}, q_C^{cold}	0.001	0.3	m^3/s
$Q_{S_i}^{\text{dump}}$, $Q_{C_j}^{\text{peak}}$	0.0	10	MW

References

[1] James B. Rawlings and David Q. Mayne. *Model Predictive Control: Theory and Design.* Nob Hill Pub., 2009.

[2] Pierre O. M. Scokaert and David Q. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic control*, 43(8):1136–1142, 1998.

[3] Sergio Lucia. *Robust multi-stage nonlinear model predictive control.* PhD thesis, Techinical University of Dortmund, 2015.

[4] John R. Birge and Franois Louveaux. *Introduction to Stochastic Programming.* Springer Publishing Company, Incorporated, 2nd edition, 2011. ISBN 1461402360, 9781461402367.

[5] Sergio Lucia, Tiago Finkler, Dahn Basak, and Sebastian Engell. A new robust NMPC scheme and its application to a semi-batch reactor example. *IFAC Proceedings Volumes*, 45(15):69–74, 2012.

[6] Sergio Lucia, Tiago Finkler, and Sebastian Engell. Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty. *Journal of Process Control*, 23(9):1306–1319, 2013.

[7] Sergio Lucia, Joel A. E. Andersson, Heiko Brandt, Moritz Diehl, and Sebastian Engell. Handling uncertainty in economic nonlinear model predictive control: A comparative case study. *Journal of Process Control*, 24(8):1247–1259, 2014.

[8] Hong Jang, Jay H. Lee, and Lorenz T. Biegler. A robust NMPC scheme for semi-batch polymerization reactors. *IFAC-PapersOnLine*, 49(7):37–42, 2016. ISSN 2405-8963. doi:https://doi.org/10.1016/j.ifacol.2016.07.213. URL `https://www.sciencedirect.com/science/article/pii/S2405896316304190`. 11th IFAC Symposium on Dynamics and Control of Process SystemsIncluding Biosystems DYCOPS-CAB 2016.

[9] Sergio Lucia and Sebastian Engell. Robust nonlinear model predictive control of a batch bioreactor using multi-stage stochastic programming. In *2013 European Control Conference (ECC)*, pages 4124–4129. IEEE, 2013.

[10] Ruben Marti, Sergio Lucia, Daniel Sarabia, Radoslav Paulen, Sebastian Engell, and César de Prada. Improving scenario decomposition algorithms for robust nonlinear model predictive control. *Computers & Chemical Engineering*, 79:30–45, 2015.

[11] Dinesh Krishnamoorthy, Bjarne Foss, and Sigurd Skogestad. Real-time optimization under uncertainty applied to a gas lifted well network. *Processes*, 4(4):52, 2016.

[12] Daniel Haßkerl, Clemens Lindscheid, Sankaranarayanan Subramanian, Patrick Diewald, Alexandru Tatulea-Codrean, and Sebastian Engell. Economics optimizing control of a multi-product reactive distillation process under model uncertainty. *Computers & Chemical Engineering*, 118:25–48, 2018.

[13] Mandar Thombre, Zawadi Mdoe, and Johannes Jäschke. Data-driven robust optimal operation of thermal energy storage in industrial clusters. *Processes*, 8(2):194, 2020.

[14] Dinesh Krishnamoorthy, Bjarne Foss, and Sigurd Skogestad. A primal decomposition algorithm for distributed multistage scenario model predictive control. *Journal of Process Control*, 81:162–171, 2019.

[15] Sergio Lucia, Sankaranarayanan Subramanian, and Sebastian Engell. Non-conservative robust nonlinear model predictive control via scenario decomposition. In *Control Applications (CCA), 2013 IEEE International Conference on*, pages 586–591. IEEE, 2013.

[16] Conrad Leidereiter, Andreas Potschka, and Hans Georg Bock. Dual decomposition for QPs in scenario tree nmpc. In *2015 European Control Conference (ECC)*, pages 1608–1613. IEEE, 2015.

[17] Wachira Daosud, Paisan Kittisupakorn, Miroslav Fikar, Sergio Lucia, and Radoslav Paulen. Efficient robust nonlinear model predictive control via approximate multi-stage programming: A neural networks based approach. In *Computer Aided Chemical Engineering*, volume 46, pages 1261–1266. Elsevier, 2019.

[18] Zhou Joyce Yu and Lorenz T. Biegler. Advanced-step multistage nonlinear model predictive control: Robustness and stability. *Journal of Process Control*, 84:192–206, 2019.

[19] Flemming Holtorf, Alexander Mitsos, and Lorenz T. Biegler. Multistage NMPC with on-line generated scenario trees: Application to a semi-batch polymerization process. *Journal of Process Control*, 80:167–179, 2019. ISSN 0959-1524. doi:https://doi.org/10.1016/j.jprocont.2019.05.007. URL `https://www.sciencedirect.com/science/article/pii/S0959152418303639`.

[20] Mandar Thombre, Zhou Joyce Yu, Johannes Jäschke, and Lorenz T. Biegler. Sensitivity-Assisted multistage nonlinear model predictive control: Robustness, stability and computational efficiency. *Computers & Chemical Engineering*, 148:107269, 2021.

[21] Zawadi Mdoe, Dinesh Krishnamoorthy, and Johannes Jaschke. Adaptive horizon multistage nonlinear model predictive control. In *2021 American Control Conference (ACC)*, pages 2088–2093. IEEE, 2021.

[22] Zawadi Mdoe, Dinesh Krishnamoorthy, and Johannes Jäschke. Sta-

bility properties of the adaptive horizon multistage model predictive control. *Journal of Process Control*, 128:103002, 2023.

[23] David L. Ma and Richard D. Braatz. Worst-case analysis of finite-time control policies. *IEEE Transactions on Control Systems Technology*, 9(5):766–774, 2001.

[24] Zoltan K. Nagy and Richard D. Braatz. Robust nonlinear model predictive control of batch processes. *AIChE Journal*, 49(7):1776–1786, 2003.

[25] Moritz Diehl, Hans Georg Bock, and Ekaterina Kostina. An approximation technique for robust nonlinear optimization. *Mathematical Programming*, 107:213–230, 2006.

[26] Zoltan K. Nagy and Richard D. Braatz. Distributional uncertainty analysis using power series and polynomial chaos expansions. *Journal of Process Control*, 17(3):229–240, 2007.

[27] Ali Mesbah, Stefan Streif, Rolf Findeisen, and Richard D. Braatz. Stochastic nonlinear model predictive control with probabilistic constraints. In *2014 American Control Conference*, pages 2413–2419. IEEE, 2014.

[28] Eric Bradford and Lars Imsland. Output feedback stochastic nonlinear model predictive control for batch processes. *Computers & Chemical Engineering*, 126:434–450, 2019.

[29] Sergio Lucia and Radoslav Paulen. Robust nonlinear model predictive control with reduction of uncertainty via robust optimal experiment design. *IFAC Proceedings Volumes*, 47(3):1904–1909, 2014.

[30] Dinesh Krishnamoorthy, Sigurd Skogestad, and Johannes Jaschke. Multistage model predictive control with online scenario tree update using recursive bayesian weighting. In *2019 18th European Control Conference (ECC)*, pages 1443–1448. IEEE, 2019.

[31] Zawadi Mdoe and Johannes Jäschke. Sensitivity-based scenario selection for multi-stage MPC along principal components: Applied to robust thermal energy storage operation. *Computers & Chemical Engineering (Under Review)*, 2024.

[32] Zawadi Mdoe, Mandar Thombre, and Johannes Jäschke. Data-driven online scenario selection for multistage NMPC. In *Computer Aided Chemical Engineering*, volume 49, pages 1627–1632. Elsevier, 2022.

[33] Evren M. Turan, Zawadi Mdoe, and Johannes Jäschke. Learning convex objectives to reduce the complexity of model predictive control. *arXiv preprint arXiv:2312.02650*, 2023.

[34] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[35] John C. Doyle. Guaranteed margins for LQG regulators. *IEEE Transactions on Automatic Control*, 23(4):756–757, 1978.

[36] J. Glover and F. Schweppe. Control of linear dynamic systems with set constrained disturbances. *IEEE Transactions on Automatic Control*, 16(5):411–423, 1971.

[37] V. L. Kharitonov. Asymptotic stability of an equilibrium position of a family of systems of linear differential equations. *Differentialnye Uravneniya*, 14:2086–2088, 1978.

[38] George Zames. Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Transactions on Automatic Control*, 26(2):301–320, 1981. doi:10.1109/TAC.1981.1102603.

[39] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishenko. The Mathematical Theory of Optimal Processes. *New York*, 1962.

[40] Arthur E. Bryson and Yu-Chi Ho. *Applied Optimal Control.* USA: Taylor and Francis, 1975.

[41] 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(6):627–631, 1970.

[42] R. W. H. Sargent and G. R. Sullivan. The development of an efficient optimal control package. In *Optimization Techniques: Proceedings of the 8th IFIP Conference on Optimization Techniques Würzburg, September 5–9, 1977*, pages 158–168. Springer, 1978.

[43] Hans Georg Bock and Karl-Josef Plitt. A multiple shooting algorithm for direct solution of optimal control problems. *IFAC Proceedings Volumes*, 17(2):1603–1608, 1984.

[44] T. H. Tsang, D. M. Himmelblau, and Thomas F. Edgar. Optimal control via collocation and non-linear programming. *International Journal of Control*, 21(5):763–768, 1975.

[45] James E. Cuthrell and Lorenz T. Biegler. On the optimization of differential-algebraic process systems. *AIChE Journal*, 33(8):1257–1270, 1987.

[46] Jacques Richalet, André Rault, J. L. Testud, and J. Papon. Model predictive heuristic control. *Automatica*, 14(5):413–428, 1978.

[47] Charles R. Cutler and Brian L. Ramaker. Dynamic matrix control?? a computer control algorithm. In *Joint Automatic Control Conference*, 17, page 72, 1980.

[48] David Q. Mayne, James B. Rawlings, Christopher V. Rao, and Pierre O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[49] Uri M. Ascher and Linda R. Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*, volume 61. SIAM, 1998.

[50] Lorenz T. Biegler. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. SIAM, 2010.

[51] Lorenz T. Biegler, Arturo M. Cervantes, and Andreas Wächter. Advances in simultaneous strategies for dynamic process optimization. *Chemical Engineering Science*, 57(4):575–593, 2002.

[52] G. De Nicolao, L. Magni, and Riccardo Scattolini. On the robustness of receding-horizon control with terminal constraints. *IEEE Transactions on Automatic Control*, 41(3):451–453, 1996.

[53] Gabriele Pannocchia, James B. Rawlings, and Stephen J. Wright. Conditions under which suboptimal nonlinear MPC is inherently robust. *Systems & Control Letters*, 60(9):747–755, 2011.

[54] Shuyou Yu, Marcus Reble, Hong Chen, and Frank Allgöwer. Inherent robustness properties of quasi-infinite horizon nonlinear model predictive control. *Automatica*, 50(9):2269–2280, 2014.

[55] E. D. Sontag. Smooth stabilization implies coprime factorization. *IEEE Transactions on Automatic Control*, 34(4):435–443, 1989. doi:10.1109/9.28018.

[56] Daniel Limón, Teodoro Alamo, Davide M. Raimondo, D. Muñoz de La Peña, José Manuel Bravo, Antonio Ferramosca, and Eduardo F. Camacho. Input-to-state stability: a unifying framework for robust model predictive control. *Nonlinear Model Predictive Control: Towards New Challenging Applications*, pages 1–26, 2009.

[57] Daniel Limón, Teodoro Alamo, Francisco Salas, and Eduardo F Camacho. Input to state stability of min–max MPC controllers for nonlinear systems with bounded uncertainties. *Automatica*, 42(5):797–803, 2006.

[58] Eduardo D. Sontag and Yuan Wang. New characterizations of input-to-state stability. *IEEE Transactions on Automatic Control*, 41(9): 1283–1294, 1996.

[59] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer Science & Business Media, 2006.

[60] Jacques Gauvin. A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming. *Mathematical Programming*, 12(1):136–138, 1977.

[61] Anthony V. Fiacco. Sensitivity analysis for nonlinear programming using penalty methods. *Mathematical Programming*, 10(1):287–311, 1976.

[62] Daniel Ralph and Stephan Dempe. Directional derivatives of the solution of a parametric nonlinear program. *Mathematical Programming*, 70(1-3):159–172, 1995.

[63] Masakazu Kojima. Strongly stable stationary solutions in nonlinear programs. In *Analysis and Computation of Fixed Points*, pages 93–138. Elsevier, 1980.

[64] Tor Aksel N. Heirung, Joel A. Paulson, Jared O'Leary, and Ali Mesbah. Stochastic model predictive control — how does it work? *Computers & Chemical Engineering*, 114:158–170, 2018. ISSN 0098-1354. doi:https://doi.org/10.1016/j.compchemeng.2017.10.026. URL https://www.sciencedirect.com/science/article/pii/S0098135417303812. FOCAPO/CPC 2017.

[65] Ali Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, 36 (6):30–44, 2016. doi:10.1109/MCS.2016.2602087.

[66] David Q. Mayne, María M. Seron, and S. V. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[67] Peter J. Campo and Manfred Morari. Robust model predictive control. In *1987 American Control Conference*, pages 1021–1026. IEEE, 1987.

[68] Sergio Lucia, Tiago Finkler, and Sebastian Engell. Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty. *Journal of Process Control*, 23(9):1306 – 1319, 2013. doi:10.1016/j.jprocont.2013.08.008.

[69] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczynski. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, 2021.

[70] D. Munoz de la Peña, Alberto Bemporad, and Teodoro Alamo. Stochastic programming applied to model predictive control. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 1361–1366. IEEE, 2005.

[71] Daniele Bernardini and Alberto Bemporad. Scenario-based model predictive control of stochastic constrained linear systems. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 6333–6338. IEEE, 2009.

[72] D. Munoz de la Peña, Teodoro Alamo, Alberto Bemporad, and Eduardo F. Camacho. Feedback min-max model predictive control based on a quadratic cost function. In *2006 American Control Conference*, pages 6–pp. IEEE, 2006.

[73] Sungho Shin, Mihai Anitescu, and Victor M. Zavala. Exponential decay of sensitivity in graph-structured nonlinear programs. *SIAM Journal on Optimization*, 32(2):1156–1183, 2022.

[74] Katta G. Murty and Santosh N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. Technical Report 85-23, Dept. of Industrial & Operations Engineering, The University of Michigan, 1985.

[75] Arthur J. Krener. Adaptive horizon model predictive control. *IFAC-PapersOnLine*, 51(13):31–36, 2018.

[76] Arthur J. Krener. Adaptive horizon model predictive control and Al'brekht's method. In *Encyclopedia of Systems and Control*, pages 27–40. Springer, 2021.

[77] Devin W. Griffith, Lorenz T. Biegler, and Sachin C. Patwardhan. Robustly stable adaptive horizon nonlinear model predictive control. *Journal of Process Control*, 70:109–122, 2018.

[78] Rohan Chandra Shekhar. *Variable Horizon Model Predictive Control: Robustness and Optimality*. PhD thesis, University of Cambridge, 2012.

[79] Dinesh Krishnamoorthy, Lorenz T. Biegler, and Johannes Jäschke. Adaptive horizon economic nonlinear model predictive control. *Journal of Process Control*, 92:108–118, 2020.

[80] Hong Chen and Frank Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.

[81] K. U. Klatt and Sebastian Engell. Gain-scheduling trajectory control of a continuous stirred tank reactor. *Computers & Chemical Engineering*, 22(4-5):491–502, 1998.

[82] Johannes Jäschke, Xue Yang, and Lorenz T. Biegler. Fast economic model predictive control based on NLP-sensitivities. *Journal of Process Control*, 24(8):1260–1272, 2014.

[83] Iain Dunning, Joey Huchette, and Miles Lubin. JuMP: a modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017. doi:10.1137/15M1020575.

[84] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59 (1):65–98, 2017. doi:10.1137/141000671. URL https://doi.org/10.1137/141000671.

[85] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.

[86] STFC Rutherford Appleton Laboratory. HSL. A collection of Fortran codes for large-scale scientific computation. *http://www.hsl.rl.ac.uk*, 2019.

[87] Tobias Raff, Steffen Huber, Zoltan K. Nagy, and Frank Allgower. Nonlinear model predictive control of a four tank system: An experimental stability study. In *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, pages 237–242. IEEE, 2006.

[88] Michael Maiworm, Tobias Bäthge, and Rolf Findeisen. Scenario-based model predictive control: Recursive feasibility and stability. *IFAC-PapersOnLine*, 48(8):50–56, 2015.

[89] Sergio Lucia, Sankaranarayanan Subramanian, Daniel Limón, and Sebastian Engell. Stability properties of multi-stage nonlinear model predictive control. *Systems & Control Letters*, 143:104743, 2020.

[90] James B. Rawlings, David Q. Mayne, and Moritz Diehl. *Model Predictive Control: Theory, Computation, and Design*, volume 2. Nob Hill Publishing Madison, WI, 2017.

[91] Davide Martino Raimondo, Daniel Limón, Mircea Lazar, Lalo Magni, and Eduardo Fernández Camacho. Min-max model predictive control of nonlinear systems: A unifying overview on stability. *European Journal of Control*, 15(1):5–21, 2009.

[92] Chao Shang and Fengqi You. A data-driven robust optimization approach to scenario-based stochastic model predictive control. *Journal of Process Control*, 75:24–39, 2019.

[93] Dinesh Krishnamoorthy, Mandar Thombre, Sigurd Skogestad, and Johannes Jäschke. Data-driven scenario selection for multistage robust model predictive control. *IFAC-PapersOnLine*, 51(20):462–468, 2018.

[94] Mandar Thombre, Dinesh Krishnamoorthy, and Johannes Jäschke. Data-driven online adaptation of the scenario-tree in multistage model predictive control. *IFAC-PapersOnLine*, 52(1):461 – 467, 2019. doi:https://doi.org/10.1016/j.ifacol.2019.06.105. 12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems DYCOPS 2019.

[95] Jennifer Puschke and Alexander Mitsos. Robust feasible control based on multi-stage eNMPC considering worst-case scenarios. *Journal of Process Control*, 69:8–15, 2018.

[96] R. I. Kachurovskii. Monotone operators and convex functionals. *Uspekhi Matematicheskikh Nauk*, 15(4):213–215, 1960.

[97] Chao Ning and Fengqi You. A data-driven multistage adaptive robust optimization framework for planning and scheduling under uncertainty. *AIChE Journal*, 63(10):4343–4369, 2017.

[98] Chao Shang, Xiaolin Huang, and Fengqi You. Data-driven robust optimization based on kernel learning. *Computers and Chemical Engineering*, 106(Supplement C):464 – 479, 2017.

[99] Chao Ning and Fengqi You. Data-driven decision making under uncertainty integrating robust optimization with principal component analysis and kernel smoothing methods. *Computers & Chemical Engineering*, 112:190–210, 2018.

[100] Runda Jia, Shulei Zhang, Zhiqi Li, and Kang Li. Data-driven tube-based model predictive control of an industrial thickener. In *2022 4th International Conference on Industrial Artificial Intelligence (IAI)*, pages 1–6. IEEE, 2022.

[101] Ian T. Jolliffe. Principal component analysis and factor analysis. In *Principal Component Analysis*, pages 115–128. Springer, 1986.

[102] Vassilios Vassiliadis. *Computational solution of dynamic optimization problems with general differential-algebraic constraints.* PhD thesis, Imperial College London (University of London), 1993.

[103] Purt Tanartkit and Lorenz T. Biegler. Stable decomposition for dynamic optimization. *Industrial & Engineering Chemistry Research*, 34 (4):1253–1266, 1995.

[104] Victor M. Zavala and Lorenz T. Biegler. The advanced-step NMPC controller: Optimality, stability and robustness. *Automatica*, 45(1): 86–93, 2009.

[105] Arezou Keshavarz, Yang Wang, and Stephen Boyd. Imputing a convex objective function. In *2011 IEEE International Symposium on Intelligent Control*, pages 613–619. IEEE, 2011.

ISBN 978-82-326-7986-7 (printed ver.)
ISBN 978-82-326-7985-0 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU
Norwegian University of
Science and Technology