NTNU – Trondheim
Norwegian University of
Science and Technology

Novel Approaches in Robust Multistage Nonlinear Model Predictive Control

Mandar Thombre

Doctoral thesis for the degree of philosophiae doctor
Trondheim, December 2020

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

Acknowledgments

It feels strange to conclude this thesis in the midst of a global pandemic. Its been a challenging year, and this thesis is due in no small part to the encouragement, support and contributions from several people.

First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Johannes Jäschke for giving me the opportunity to work on this PhD, and for his continuous support and guidance throughout. It has been an enriching experience and I am deeply grateful for his patience and advice. Johannes not only kept pointing me in the right direction, but also gave me the freedom to develop my own ideas. I also cannot thank him enough for suggesting and facilitating my research stay at Carnegie Mellon University.

I would like to thank my co-supervisor Prof. Lorenz T. Biegler for the priceless mentoring in the latter part of my PhD. I am extremely thankful to him for hosting me at Carnegie Mellon University for 6 months. He is a constant source of great ideas, and I have learned an lot from my many discussions with him. I also immensely enjoyed working with him during his recent sabbatical in Trondheim.

I thank my co-supervisor Prof. Sigurd Skogestad for his wealth of knowledge and for always being supportive. He has helped create a very vibrant and fun environment at the PSE group. I also thank Prof. Heinz Preisig, who facilitated my undergraduate exchange stay at NTNU back in 2014. His mentoring during the exchange stay and subsequently during my master thesis was a major reason why I chose to pursue a PhD.

Many thanks to Dr. Brage Rugstad Knudsen from Sintef Energy for the fruitful collaboration on the thermal energy storage project, and for helping procure relevant data from the industry. On that note, I would like to thank Mo Fjernvarme AS for providing useful industrial data that has been crucial in some of the work that went into this thesis. I gratefully acknowledge the financial support from the FME HighEFF, which is financed by the Research Council of Norway, and which enabled the collaborations with Sintef Energy and Mo Fjernvarme.

A special thanks to Zawadi Mdoe and Sandeep Prakash, who I had the opportunity to co-supervise as master thesis students. I got as much out of the supervision as they (hopefully) did, and both have contributed to some of the work that went into this thesis. I wish them both all the very best for their ongoing PhD endeavors.

I have greatly enjoyed working in the PSE group these past years, and a lot of credit for that goes to my friends and colleagues. Thanks to Cristina, Zawadi, David, Haakon, Adriana and Julian for the nice times we had in the K4-239 office. Cheers to Dinesh, Adriaen, Timur, Eka, Tamal, Jose, Allyne, Ana, Pedro, Bahareh, Fabienne, Lucas, Andrea, Robert, Sigve, Tobias, Cansu, Carol, Halvor, Rizwan, Saket, Simen, Peter, Evren and all the others for the many laughs, breaks and random conversations. Shoutout to my friends in the USA - Saif, David, Jennifer, Vibhav, Joyce, Tom, Yeonsoo, Tarang and the rest for a great time in Pittsburgh.

I would like to thank my PhD evaluation committee: Prof. Sergio Lucia, Dr. Devin Griffith and Prof. Sebastien Gros for their valuable time in reading and evaluating this thesis.

Finally, my thoughts go to my family and friends in India, without whom this thesis would not have been possible. I am forever indebted to my parents and my brother for their unconditional love and support, and their unfaltering belief in me even during the difficult times. This thesis is dedicated to them.

Mandar Thombre
Trondheim, Norway
December 2020

Abstract

Model predictive control has achieved remarkable success across a range of industrial applications, thanks to its capacity for handling multivariable constrained control problems. To account for more accurate nonlinear process models, the framework has been extended to nonlinear model predictive control. To ensure optimal performance under uncertainty, a class of methods classified under robust (N)MPC has received widespread attention in the past couple of decades. Many of these methods, however, suffer from a high degree of conservatism, while also having significant limitations in their practical applicability. More recently, the robust multistage NMPC, based on a scenario tree evolution of uncertainty, has emerged as a promising approach that offers robustness with low conservatism.

A major issue with this approach, however, is that it scales poorly with increasing dimensionality of the uncertainty. The construction of the scenario tree is such that the computational complexity of the optimization problem grows rapidly for larger problems. Another issue with the approach is that conventional scenario tree constructions rely on heuristics that may not be sufficiently tailored to the application at hand, and may thus lead to unnecessary conservatism.

This thesis addresses the above issues by extending the robust multistage NMPC framework with novel approaches and algorithms. In the first part of the thesis, a data-driven approach to scenario tree construction is proposed. The proposed approach leverages correlations in the available uncertainty data to seek a more compact scenario tree, thus reducing conservatism. A dynamic scenario tree update strategy is considered, where one is always up-to-date with the latest uncertainty information. The data-driven approach is applied to a detailed thermal energy storage case study to achieve robust optimal operation under varying supply and demand of heat. In this case study, the uncertainty data is based upon an industrial data set sourced from a district heating company, allowing for extensive data analysis for scenario selection.

The second part of the thesis presents an approximation strategy for robust multistage NMPC that significantly improves its computational efficiency. This strategy is based on finding those scenarios in the scenario tree that are most likely to cause constraint violations, and explicitly optimizing only over these scenarios to achieve robustness. The remaining scenarios are accounted for implicitly with a sensitivity-based approximation. This sensitivity-assisted multistage NMPC (samNMPC) approach is shown to have similar results to conventional multistage NMPC in terms of robustness and optimal performance, but at a fraction of the computational cost. A theoretical analysis based on soft-constrained formulations paves the way for showing recursive feasibility and robust stability properties of the samNMPC method. Finally, the samNMPC approach is extended with a path-following algorithm (sampfNMPC) to improve its approximation accuracy even further. The sampfNMPC approach is shown to offer an almost identical result to the conventional multistage NMPC in terms of robustness and setpoint tracking.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Chemical process plants are operated under a wide variety of operating conditions, product specifications and safety limits, often following highly complex dynamics. In a world with ever-increasing energy demands, it becomes vital to optimize these processes from an environmental as well as an economic perspective. However, a key challenge in this endeavor is that a large majority of modern real-world processes lack perfect system information and have to contend with significant uncertainties. To this end, advanced process control methods seek to achieve optimal operation in the presence of uncertainty.

A powerful tool that has been widely used for control and optimization in the chemical process industry is model predictive control (MPC), mainly because of its ability to handle complex multivariable systems under process constraints. Based on model predictions, MPC computes an optimal control trajectory that minimizes a certain cost function over a prediction horizon [1]. Plant dynamics are often highly nonlinear, and hence the nonlinear counterpart of MPC (NMPC) has received attention. Inevitably, the performance of such a model-based controller is affected by how well the model describes the real system, as well as the process disturbances that affect the system dynamics. The presence of plant-model mismatch or noise can cause the system to violate constraints or even be unstable. Although a standard NMPC implementation provides some inherent robustness against uncertainty, this is not enough when the uncertainty is pronounced. As such, robust NMPC approaches that rigorously handle the uncertainty have received attention in the past few decades [2].

There are, however, significant challenges in implementing many of these robust NMPC approaches. Some methods are highly conservative, while some others are computationally very expensive. Some are unsuitable for nonlinear systems, while some others do not scale well for large problems. A brief background of robust NMPC approaches is given in Section 2.1. One robust NMPC approach that overcomes many of these challenges, and has gained popularity in recent years, is the multistage NMPC proposed by Lucia [3], based on the concepts of multistage stochastic programming. In the paradigm of stochastic programming, the decision variables may be either of the two forms: “here-and-now” variables representing decisions that have to be taken before the realization of uncertainty - these use *a priori* information about the uncertainty; and “wait-and-see” variables representing decisions that have to be taken after the uncertain data becomes known - these can be used to hedge against future realizations of the uncertainty [4]. These latter decision variables allow for *recourse* action that help in reducing the conservativeness of the “here-and-now” decision variables. This is the central idea in the multistage NMPC approach, where future control decisions represent the recourse action to the uncertainty evolution in response to the current control decision. A key feature of this approach is that it offers robustness in terms of constraint feasibility without being overly conservative i.e. without the control decisions being too cautious. Multistage NMPC has been shown to provide robust constraint satisfaction with a low level of conservatism on a wide range of applications, including semi-batch polymerization [5, 6, 7, 8], batch bioreactor [9], hydrodesulphurization [10], gas lifted wells in oil and gas production [11], multi-product distillation [12] and penicillin fermentation [13].

In robust multistage NMPC, the uncertainty is modeled such that it propagates through time in the form of a scenario tree, with each scenario representing a distinct realization of the uncertainty across the prediction horizon. By design, each scenario is comprised of the combinations of discrete realizations of the uncertain parameters. Naturally, the choice of these discrete realizations affects the performance of the controller, particularly with respect to how conservative it is. A consequence of the scenario tree formulation is also that the number of scenarios grows exponentially with the number of uncertain parameters, as well as with the number of discrete realizations of each uncertain parameter. In addition, the scenarios also expand exponentially further down the prediction horizon. All of these factors contribute to an increasing problem size that quickly becomes computationally expensive to solve, causing a delay in the availability of the solution. As noted by [14], computational delay in implementing control actions may lead to system instability or suboptimal solutions.

The topics of this thesis focus on two important aspects related to multistage NMPC:

1. selecting scenarios in the scenario tree that better describe the uncertainty, and
2. improving the computational efficiency by addressing the exponential growth limitation of the scenario tree.

The first aspect of scenario selection is related to that of system uncertainty description. For online control of nonlinear systems, a common approach for treating uncertainty is to employ first-order approximations of the process model with respect to the uncertain parameters [15, 16, 17]. An alternate formulation based on probabilistic chance constraints uses polynomial chaos expansion to propagate the uncertainty through the nonlinear system model [18, 19, 20]. For multistage NMPC, the heuristic suggested by [6] to build up the scenario tree is to take combinations of the extreme values of the uncertain parameter ranges as the discrete realizations in each stage. However, this approach may lead to solutions with higher conservativeness. To reduce this conservativeness, a range reduction approach for parametric uncertainty was suggested by [21], making use of dynamic experiment design for guaranteed parameter estimation. A recursive Bayesian weighting approach is considered in [22] to update the scenarios in order to reduce conservativeness. In the first part of this thesis, a data-driven scenario selection approach based on sampled uncertainty data is considered within multistage NMPC to obtain less conservative but robust solutions.

The second aspect of computational efficiency in multistage NMPC can be tackled in two ways: decomposition methods and approximation methods. Decomposition methods exploit the inherent structure of the multistage NMPC problem by decoupling all the scenarios and solving the smaller subproblems separately. To this end, algorithms based on primal decomposition [23] and dual decomposition [10, 24, 25] have been proposed. On the other hand, approximation methods seek to replace the larger problem with a smaller problem without losing its main features. Examples include the work by [26], where cost-to-go functions of different scenarios are approximated by neural networks and applied to a semi-batch reactor. An online scenario generation approach that approximates the multistage NMPC with a much smaller scenario tree is proposed in [27]. This method is based on finding the worst-case realizations of uncertainty with respect to constraint feasibilities. In [28], an algorithm based on nonlinear programming (NLP) sensitivities is proposed, that can compute fast approximations of the multistage NMPC solutions. In the second part of this thesis, a computationally fast, sensitivity-based approximation strategy is proposed, that prunes the scenario tree by quickly identifying scenarios most likely to cause constraint violations.

1.2 Scope of the thesis

The goal of this thesis is to extend the multistage NMPC framework with approaches that better describe the uncertainty information and significantly improve its computational efficiency, without sacrificing its robust performance. The proposed approaches make use of statistical data-analysis, control theory and nonlinear programming theory to further the practical applicability of robust multistage NMPC. They allow for fast, real-time implementation of robust NMPC, and the corresponding optimization problems are reasonably straightforward to formulate. These extensions to robust multistage NMPC are thus well suited for implementation on large-scale problems. The performances of the these novel approaches are evaluated with the help of various case studies, and are shown to have key advantages over the standard NMPC and conventional multistage NMPC methods.

1.3 Thesis structure and main contributions

As alluded before, this thesis consists of two main parts and is organized as follows: Chapter 2 gives a brief background of robust multistage NMPC, the formulations of the standard and multistage NMPC problems, a short recap of the relevant NLP properties, and the basics of the interior-point algorithm for solving NLPs.

Part I of this thesis is about a data-driven scenario selection approach in multistage NMPC. In Chapter 3, a data-driven approach based on principal component analysis (PCA) is proposed to dynamically select the scenarios. When time-varying uncertainty is considered, PCA can be performed online to select new scenarios whenever the uncertainty data is updated. The results, demonstrated on a simple two-plant system with a thermal storage tank, show that the solution obtained is less conservative with the data-driven scenario selection strategy compared to the conventional approach. This chapter is largely adapted from [29].

In Chapter 4, the data-driven approach is extended to a detailed case-study of a thermal energy storage system to achieve robust optimal operation. The uncertainties in the system are the profiles for supply and demand of heat, the data for which is sourced from a district heating company. In addition to scenario selection, PCA is also used to detect outliers in the industrial data. The results show that the data-driven approach is successfully able to keep the system from violating any operating constraints, as opposed to the standard NMPC. It is shown that the approach is robust even in the presence of significant deviations between the predicted and actual heat profiles. The results in this chapter are published in [30].

The thesis includes a minor contribution in the form of Appendix D, where the case of the thermal energy storage system is investigated from a design optimization perspective. These results are published in [31]. However, since this analysis is unrelated to the robust multistage NMPC schemes presented in the rest of the thesis, it is only included in the Appendix.

Part II deals with a sensitivity-based approximation strategy for multistage NMPC to address the exponential growth in problem size. Chapter 5 presents an approximate sensitivity-assisted multistage NMPC (samNMPC) scheme that reduces the problem size by dividing the scenario set into so-called “critical” and “noncritical” scenarios, with the former composed of the worst-case realizations of the uncertain parameters. In this approach, the optimization is sought explicitly over the critical scenarios, while noncritical scenarios are included implicitly through NLP sensitivity-based approximations in the objective function. A key advantage of the proposed approach is that the problem size is independent of the number of constraints and scales only linearly with the length of the robust horizon. This allows for faster computations with longer robust horizons that more rigorously account for future uncertainty. The samNMPC approach is applied to two case studies for tracking setpoints, and the results show that it compares favorably in performance and robustness to ideal multistage NMPC, but with a significant reduction in computational cost.

In Chapter 6, the conventional multistage NMPC and samNMPC problems are shown to be recursively feasible and robustly stable with rigorous proofs. In particular, the given proofs consider formulations of the NMPC problems with soft inequalities, and are also applicable for the robust horizon assumption. The content of Chapters 5 and 6 is based on [32].

Chapter 7 extends the samNMPC framework with a path-following approach, termed sampfNMPC. Here, a reduced NLP comprising only of critical and nominal scenarios is solved first, and then the full multistage NMPC solution is sought by path-following from the reduced NLP solution along all the noncritical scenario uncertainties. The path-following algorithm is based on solving a sequence of predictor-corrector quadratic programs (QPs) that approximate the NLPs. Applied to a case study for tracking setpoints, the sampfNMPC algorithm is shown to offer an almost identical performance to the conventional multistage NMPC approach in terms of tracking and robustness. The content of this chapter is based on the draft [33].

The thesis concludes with some summarizing remarks and provides an outlook for future research directions.

Chapter 2

Robust Multistage Nonlinear Model Predictive Control

This chapter gives a brief background leading up to the development of robust multistage NMPC, and then introduces the formulations of standard and robust multistage NMPC. These formulations form the basis for discussing the novel approaches in robust multistage NMPC shown in later chapters. Also included is a brief introduction to NLP properties and the interior-point algorithm used in solving large-scale NLPs.

2.1 Brief background

MPC is a modern control technique that has its roots in optimal control theory. System dynamics are typically modeled as ordinary differential equations (ODEs) or differential and algebraic equations (DAEs). These equations are in the continuous-time form and their corresponding optimal control problem formulations are thus infinite-dimensional. In general, these infinite-dimensional problems are not straightforward to solve. For the special case of linear systems with a quadratic cost function, an analytical solution can be obtained by solving the Riccati differential equation. This is known as the linear quadratic regulator (LQR). Such analytical solutions are generally not possible for nonlinear optimal control problems, and thus they have to be solved numerically.

Approximate numerical solutions of these problems can be sought in two different ways: direct and indirect approaches [34]. The indirect approach comes from the application of Pontryagin's maximum principle [35], where optimality conditions are first solved and then discretized to obtain the solution. On the other hand, the direct approach involves first discretizing the optimal control problem into a finite-dimensional NLP which is then solved to get the numerical solution. Examples of the latter include multiple shooting [36] and collocation on finite elements [37, 38]. Note that the direct collocation method has been used throughout this thesis for simultaneous optimization of the relevant optimal control problems.

Practical implementation of NMPC algorithms thus relies on recasting the optimal control problems as NLPs. A discretized mathematical model of the system is used to predict its evolution and, based on this, a fully discretized NLP is solved to compute a sequence of control actions across a finite prediction horizon. Only the first control action in this sequence is implemented to the plant. To account for the mismatch between the model and the real system, state feedback information is incorporated by repeatedly solving the NLP with updated state information, and with a receding prediction horizon.

As noted before, standard NMPC offers some inherent robustness against uncertainty - see [39, 40, 41], for instance. However, a more rigorous treatment is needed for general nonlinear systems, prompting research into robust NMPC approaches. In [2], two different ways are noted to achieve robustness in MPC: a direct way wherein robust contraction constraints are enforced to guarantee stability, and an indirect way wherein the objective function and uncertainty description are specified in a way such that the optimal control actions lead to robust stability.

The tube-based MPC, proposed for linear systems by [42], is based on the direct way. This framework was extended to nonlinear systems in [43]. Here, a nominal controller computes a nominal trajectory and a so-called ancillory controller forces the evolution of the uncertain system to be within a tube centered around the nominal trajectory. Although tube-based control can guarantee robustness, it is usually highly conservative and does not ensure optimal performance under uncertainty. Moreover, the ancillory control law is difficult to compute for nonlinear systems, limiting the practical applicability of the tube-based robust NMPC.

Alternatively based on the indirect way, the min-max MPC [44] presents a robust MPC strategy where the optimal control trajectory is computed such that it minimizes the cost of the worst-case realization of the uncertainty. However, this approach ignores available future recourse actions that may counteract the uncertainty, and thus can lead to overly conservative or infeasible results. To remedy this, feedback min-max MPC was proposed in [45], where closed-loop optimization is sought over different control *policies* for different realizations of the uncertainty. Since feedback is explicitly accounted for, this controller leads to lower conservatism and avoids infeasibility. Combining aspects of stochastic programming, the evolution of uncertainty is described here in the form of a scenario tree. This framework was extended for nonlinear systems in [3], and led to the development of the robust multistage NMPC approach that is the focus of this thesis. The multistage NMPC formulation is discussed later in this chapter.

2.2 Standard NMPC formulation

Consider a system with dynamics described by the discrete-time mapping:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k) \tag{2.1}$$

where $\mathbf{x}_k \in \mathbb{X} \subset \mathbb{R}^{n_x}$ are the state variables, $\mathbf{u}_k \in \mathbb{U} \subset \mathbb{R}^{n_u}$ are the control variables, and $\mathbf{d}_k \in \mathbb{D} \subset \mathbb{R}^{n_d}$ represents the time-varying uncertainty in the model. The sets $\mathbb{X}$ and $\mathbb{U}$ are the domains for the state and control variables, respectively, whereas $\mathbb{D}$ is the bounded uncertainty set. The function $\mathbf{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \to \mathbb{R}^{n_x}$, along with $\mathbf{f}(0,0,0) = 0$, represents the nominal model of the system.

In the standard NMPC controller, the model uncertainty is not explicitly accounted for. At time t_k, the current state $\mathbf{x}_k$ is obtained from plant measurements and the following NLP is solved:

$$\begin{aligned}
\min_{\mathbf{z}_l, \mathbf{v}_l} \quad & \phi(\mathbf{z}_N, \mathbf{d}^0_{N-1}) + \sum_{l=0}^{N-1} \varphi(\mathbf{z}_l, \mathbf{v}_l, \mathbf{d}^0_l) && \text{(2.2a)}\\
\text{s.t.} \quad & \mathbf{z}_{l+1} = \mathbf{f}(\mathbf{z}_l, \mathbf{v}_l, \mathbf{d}^0_l) \qquad l = 0, \dots, N-1 && \text{(2.2b)}\\
& \mathbf{z}_0 = \mathbf{x}_k && \text{(2.2c)}\\
& \mathbf{z}_l \in \mathbb{X}, \mathbf{v}_l \in \mathbb{U}, \mathbf{z}_N \in \mathbb{X}_f && \text{(2.2d)}
\end{aligned}$$

where N is the length of the prediction horizon, $\mathbf{z}_l$ and $\mathbf{v}_l$ are the state and control variable vectors, respectively at time t_{k+l}, and the value of the uncertain model parameter is fixed at a nominal $\mathbf{d}_l^0$ for all time steps.

The objective function (2.2a) is composed of the stage cost $\varphi : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \rightarrow \mathbb{R}$, and the terminal cost $\phi : \mathbb{R}^{n_x} \times \mathbb{R}^{n_d} \rightarrow \mathbb{R}$. Constraints (2.2b) represent the dynamic model used in the controller. The controller is initialized with the current state $\mathbf{x}_k$ at t_k, as shown in (2.2c), and (2.2d) represents the bound constraints on the state and control variables. Note that the set $\mathbb{X}_f \subset \mathbb{X}$ is the terminal region, and is typically used to ensure recursive feasibility of the finite-horizon NMPC controller.

At each time step k, problem (2.2) solves for a predicted state trajectory $\mathbf{x}_{[k,k+N]}$ and a corresponding sequence of control inputs $\mathbf{u}_{[k,k+N-1]}$ across the prediction horizon $[k, k+N]$. From the obtained optimal sequence of control inputs, the first stage control input $\mathbf{u}_k = \mathbf{v}_0^*$ is applied to the plant. This can also be represented in the form of a feedback control law $\mathbf{u}_k = \mathbf{h}(\mathbf{x}_k)$, where $\mathbf{h} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}$. After injecting $\mathbf{u}_k$, the plant evolves from t_k to t_{k+1} according to (2.1). The updated state $\mathbf{x}_{k+1}$ at t_{k+1} is used to solve problem (2.2) at the next time step, and the procedure repeats. It is assumed that the states $\mathbf{x}_k$ can be measured at each sampling time.

Typically, a receding prediction horizon is used as the controller moves forward in time. This receding horizon nature, along with the incorporation of state feedback information, allows standard NMPC to offer a limited degree of robustness [39, 40]. However the plant-model mismatch arising due to uncertainty causes deteriorating performance in standard NMPC, particularly with respect to constraint satisfaction.

A key performance metric for any NMPC scheme is the computational delay, which is the time difference between obtaining updated state information from the plant at t_k and applying the computed control input $\mathbf{u}_k$ to the plant. To minimize this delay, it is important to be computationally fast in solving NMPC problem formulations such as (2.2). A comprehensive discussion on standard NMPC can be found in [1].

2.3 Robust multistage NMPC formulation

In contrast to standard NMPC, robust NMPC methods rigorously account for the model uncertainty. In presence of plant-model mismatch, the evolution of the state trajectory at time step k depends on the actual realization of the uncertain parameter $\mathbf{d}_k \in \mathbb{D}$. As such, the sequence of control inputs $\mathbf{u}_{[k,k+N-1]}$ should correspond to a cone of state trajectories $\{\mathbf{x}_{[k,k+N-1]}\}_{\mathbb{D}}$ [46]. The min-max MPC solves for a single control profile that applies to all realizations of the uncertainty, including the worst-case realization [44].

However, optimizing over a single control profile is overly conservative and disregards the fact that feedback is available. In other words, it does not explicitly take into account that new uncertainty information will be available in the future and the future control inputs can take recourse action to negate the effect of the current uncertainty. With this notion of feedback, it may be prudent to optimize over different control policies for different realizations of uncertainty (see [47, 48]). More precisely, a cone of control profiles $\{\mathbf{u}_{[k,k+N-1]}\}_{\mathbb{D}}$ needs to be computed.

This problem can be made tractable by discretizing the uncertainty set $\mathbb{D}$, which converts the cone of state trajectories into discrete scenarios. The future evolution of the uncertainty can be modeled in the form of a scenario tree as shown in [45], and closed-loop optimization sought over the different scenarios, thereby reducing conservativeness compared to the min-max approach. This is the main idea of multistage MPC, which was further expanded upon in [6] in the context of nonlinear systems to propose the multistage NMPC.

Consider that the uncertainty set $\mathbb{D}$ is discretized into a set $\mathbb{M}$ of discrete realizations. A commonly used heuristic is to use the combinations of $\{\mathsf{max}, \mathsf{nominal}, \mathsf{min}\}$ values of each uncertain parameter to build the scenario tree. The set of discrete realizations of uncertainty is thus:

$$\mathbb{M} = \{d_1^{\mathsf{max}}, d_1^{\mathsf{nom}}, d_1^{\mathsf{min}}\} \times \cdots \times \{d_{n_d}^{\mathsf{max}}, d_{n_d}^{\mathsf{nom}}, d_{n_d}^{\mathsf{min}}\} \tag{2.3}$$

where n_d is the dimensionality of the uncertain parameter vector, and $|\mathbb{M}| = 3^{n_d}$. Figure 2.1 shows the evolution of a fully branched scenario tree with 27 scenarios ($n_d = 1$, $N = 3$). The current state at t_k is the root node of the scenario tree. At t_{k+1}, there are 3 possible states corresponding to the 3 discrete realizations of the uncertainty. The scenario tree continues branching further along the prediction horizon, with 9 possible states at t_{k+2}, and 27 possible states at t_{k+3}. A scenario is defined as a sequence of states from the root node to the leaf node at the end of the prediction horizon. Thus there are 27 discrete scenarios in the scenario tree shown in Figure 2.1. In general, the total number of scenarios in a fully branched scenario tree is $|\mathbb{M}|^N$.

It is apparent that for longer prediction horizons, the number of scenarios grows exponentially large, and it becomes computationally infeasible to solve the resulting optimiza-

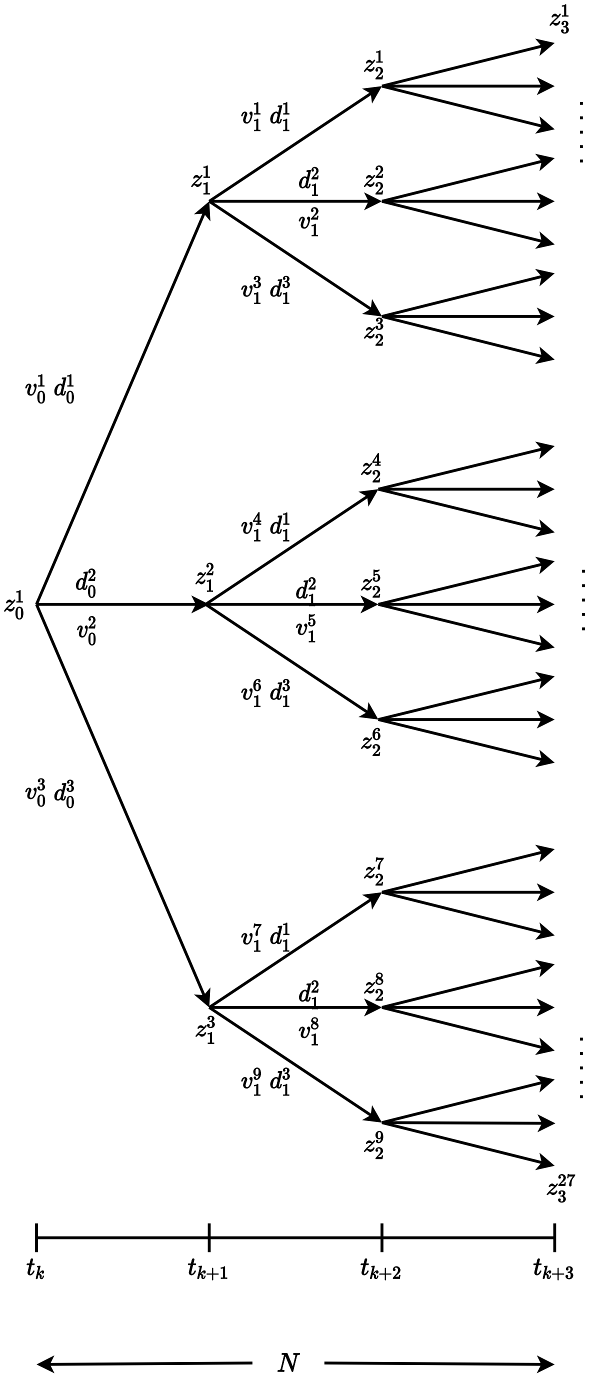


Figure 2.1: Fully branched scenario tree with $n_d = 1$ and $N = 3$.

tion problem. To resolve this, a robust horizon of length N_r was suggested in [6], wherein the branching of the scenario tree is stopped after t_{k+N_r}, and the uncertain parameters stay at their t_{k+N_r} values for the rest of the prediction horizon. Figure 2.2 shows a truncated scenario tree with $N_r = 2$, and 9 scenarios. The number of scenarios is $|\mathbb{M}|^{N_r}$, which is much lower than the fully branched scenario tree provided $N_r \ll N$, albeit with the caveat that the truncated scenario tree does not account for every possible evolution of the uncertainty up to N. This is justified on the basis that only the immediate control input is applied to the plant and the next control inputs are recomputed anyway in a receding horizon implementation.

The resulting optimization problem to be solved with current state $\mathbf{x}_k$ at t_k is formulated as follows:

$$\begin{aligned}
\min_{\mathbf{z}_l^c, \mathbf{v}_l^c} \quad & \sum_{c \in \mathbb{C}} \omega^c \Big(\phi(\mathbf{z}_N^c, \mathbf{d}_{N-1}^c) + \sum_{l=0}^{N-1} \varphi(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^c) \Big) && \text{(2.4a)} \\
\text{s.t.} \quad & \mathbf{z}_{l+1}^c = \mathbf{f}(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^c) \quad l = 0, \ldots, N-1 && \text{(2.4b)} \\
& \mathbf{z}_0^c = \mathbf{x}_k && \text{(2.4c)} \\
& \mathbf{v}_l^c = \mathbf{v}_l^{c'} \quad \{(c, c') \mid \mathbf{z}_l^c = \mathbf{z}_l^{c'}\} && \text{(2.4d)} \\
& \mathbf{z}_l^c \in \mathbb{X}, \mathbf{v}_l^c \in \mathbb{U}, \mathbf{z}_N^c \in \mathbb{X}_f, \mathbf{d}_l^c \in \mathbb{D} && \text{(2.4e)} \\
& \mathbf{d}_{l-1}^c = \mathbf{d}_l^c \quad l = N_r, \ldots N-1 && \text{(2.4f)} \\
& \forall c, c' \in \mathbb{C}
\end{aligned}$$

where $\mathbb{C}$ is the set of all scenarios, ω^c is the probability of each scenario, and $\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^c$ represent the vectors of state variables, control variables and uncertain parameters at stage l and scenario c. The objective function in (2.4a) is the weighted sum of the cost across all the scenarios at $\mathbf{x_k}$, with ω^c being the probability associated with each scenario. The equation (2.4f) imposes that the uncertain parameters remain constant after the robust horizon.

Equation (2.4d) represents the non-anticipativity constraints (NACs) which impose that all control inputs corresponding to branches of the same parent node in the scenario tree, are equal. This is because only one control input $\mathbf{u}_k = \mathbf{v}_0$ can be injected into the plant at t_k, irrespective of how $\mathbf{d}_k$ evolves. In other words, one cannot anticipate how the state trajectory is going to evolve from a particular node before a control decision is taken at the node. In Figure 2.1 for instance, $\mathbf{v}_0^1 = \mathbf{v}_0^2$ and $\mathbf{v}_1^1 = \mathbf{v}_1^2$ are NACs (however $\mathbf{v}_1^1$ and $\mathbf{v}_1^4$, for example, are not coupled by non-anticipativity because their parent nodes

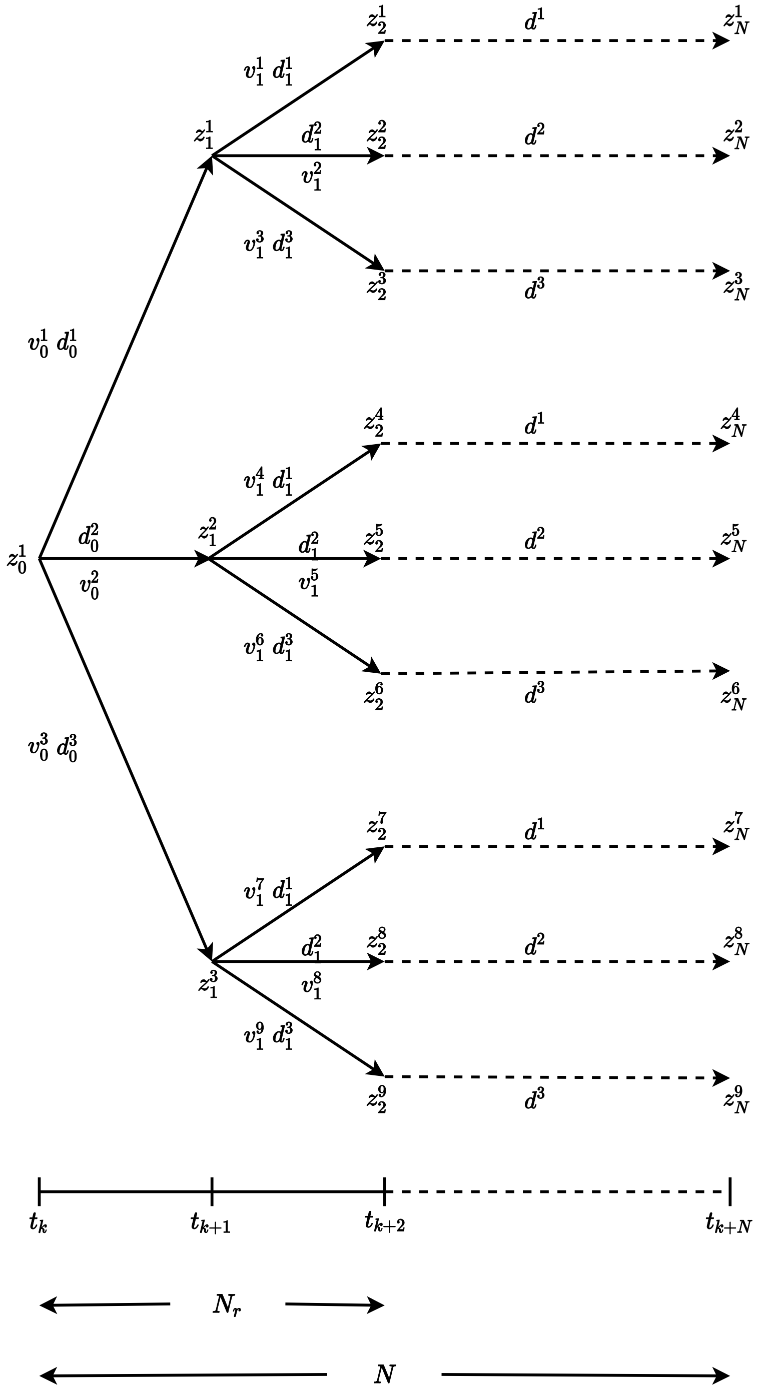


Figure 2.2: Robust horizon assumption: scenario tree with $n_d = 1$ and $N_r = 2$.

are different). Note that the number of NACs grows exponentially with the number of scenarios.

It must be noted that in robust multistage NMPC, the scenario tree grows exponentially not only with the length of the robust horizon, but also with the number of uncertain parameters. Hence, for complex systems with multiple uncertain parameters, it is common to assume a very small robust horizon (for example $N_r = 1$) to keep the optimization problem tractable.

2.4 Nonlinear programming properties

Since the multistage NMPC problem (2.4) is parametric in its uncertainty $\mathbf{d}_k$, its properties are investigated with respect to $\mathbf{d}_k$ by rewriting the problem in the more compact general NLP form:

$$\begin{aligned} \min_{x} \quad & F(x;p) \\ \text{s.t.} \quad & c(x;p) = 0 \\ & x \geq 0 \end{aligned} \tag{2.5}$$

where x are all the variables in (2.4), and p are all the uncertain parameters $\mathbf{d}$ of (2.4). The solution of (2.5) is given by a KKT point, which satisfies the Karush-Kuhn-Tucker conditions for (2.5):

Definition 1. *(KKT, [49]) KKT conditions for* (2.5) *are given by:*

$$\begin{aligned} &\nabla F(x^*;p) + \nabla c(x^*;p)\lambda - \nu = 0 \\ &c(x^*;p) = 0 \\ &0 \leq \nu \perp x^* \geq 0 \end{aligned} \tag{2.6}$$

for some multipliers (λ, ν)*, where* x^* *is a KKT point. The Lagrangian function of* (2.5) *is defined as* $\mathcal{L}(x, \lambda, v; p) := F(x;p) + \lambda^T c(x;p) - \nu^T x$*.*

A constraint qualification (CQ) is required so that a KKT point is necessary for a local minimizer of (2.5) [49]. For (2.5), the following CQ is widely invoked:

Definition 2. *(LICQ, [49]) The linear independence constraint qualification (LICQ) holds at* x^* *when the gradient vectors*

$$\nabla c(x^*;p) \textit{ and } \nabla x_j^*; \quad j \in J \textit{ where } J = \{j | x_j^* = 0\} \tag{2.7}$$

are linearly independent. Here the set J is known as the active set of bounds.

The LICQ implies that the set of multipliers (λ, ν) satisfying the KKT conditions are unique. A weaker, generalized form of the LICQ is as follows:

Definition 3. *(MFCQ, [49]) For* (2.5)*, the Mangasarian-Fromovitz constraint qualification (MFCQ) holds at x^* if and only if*

- $\nabla c(x^*; p)$ *is linearly independent and the singular values of* $\nabla c(x^*; p)$ *are bounded away from zero.*
- *There exists a vector q such that*

$$\begin{aligned} \nabla c(x^*; p)^T q &= 0 \\ q_j > 0, \quad j &\in J. \end{aligned} \tag{2.8}$$

The MFCQ implies that the set of multipliers (λ, ν) is a compact convex polytope [50].

Further, the KKT point is a local minimum if the following sufficient second order conditions apply.

Definition 4. *(SSOSC, [51]) The KKT point x^* with multipliers (λ, ν) is a strict local optimum if there exists some $\epsilon > 0$ and the following strong second order sufficient conditions (SSOSC) hold at x^*:*

$$q^T \nabla_{xx} \mathcal{L}(x^*, \lambda, \nu;\ p) q \ \geq \epsilon > 0 \quad \textit{for all } q \neq 0 \tag{2.9}$$

such that

$$\begin{aligned} \nabla c_i(x^*; p)^T q = 0, &\quad i = 1, .., n_c \\ q_j = 0, &\quad \textit{for } \nu_j \ \geq \epsilon > 0, j \in J. \end{aligned} \tag{2.10}$$

Definition 5. *(GSSOSC, [52]) The generalized strong second order sufficient condition (GSSOSC) holds at x^* when the SSOSC holds for all multipliers (λ, ν) that satisfy the KKT conditions of* (2.5).

Finally, for the active set of bounds, the following definition is given:

Definition 6. *(SC, [49]) At a KKT point x^* with multipliers (λ, ν), the strict complementarity condition (SC) is defined by $\nu_j + x_j^* > 0$ for each $j \in J$.*

2.5 Interior-point algorithm

Interior-point methods represent an attractive way of handling NLPs with a large number of inequality constraints. An efficient implementation of the interior-point algorithm for handling large-scale NLPs is `IPOPT` [53]. The `IPOPT` algorithm substitutes the inequality constraints in (2.5) with a barrier function in the objective and solves a sequence of problems, indexed by m with $\lim_{m\to\infty}\mu_m \to 0$:

$$\begin{aligned} \min_{x} \qquad & F(x;p) - \mu_m \sum_{i=1}^{n_x} ln(x_i) \\ \text{s.t.} \qquad & c(x;p) = 0 \end{aligned} \tag{2.11}$$

For a negligibly small $\mu_m > 0$, the Lagrangian of (2.11) can be denoted as:

$$\mathcal{L}(x, \lambda, \nu;\, p) := F(x;p) + \lambda^T c(x;p) - \nu^T x \tag{2.12}$$

and the KKT conditions of (2.11) are:

$$\nabla_x \mathcal{L}(x, \lambda, \upsilon;\, p) = \nabla_x F(x;p) + \nabla_x c(x;p)\lambda - \upsilon = 0 \tag{2.13a}$$

$$c(x;p) = 0 \tag{2.13b}$$

$$XVe = \mu e \tag{2.13c}$$

where $X = diag(x)$, $V = diag(\upsilon)$ and $e^T = [1, \ldots, 1]$.

The KKT conditions (2.13) are solved using the Newton's method to obtain the search direction. Linearizing (2.13) around a given current iterate $[x_k, \lambda_k, \nu_k]^T$ leads to the following KKT system:

$$\begin{bmatrix} \nabla_{xx}\mathcal{L}(s_k(\mu;p)) & \nabla_x c(s_k(\mu;p)) & -I \\ \nabla_x c(s_k(\mu;p))^T & 0 & 0 \\ V(\mu;p) & 0 & X(\mu;p) \end{bmatrix} \begin{bmatrix} d_k^x \\ d_k^\lambda \\ d_k^\nu \end{bmatrix} = - \begin{bmatrix} \nabla_x \mathcal{L}(s_k(\mu;p)) \\ c(x_k(\mu;p)) \\ X_k V_k e - \mu e \end{bmatrix} \tag{2.14}$$

where $[d_k^x, d_k^\lambda, d_k^\nu]^T$ is the search direction. As $\mu_m \to 0$, the solutions of (2.11) approach the solution of the original NLP (2.5). The primal-dual solution vector is

$$s(\mu;p) = \begin{bmatrix} x(\mu;p) \\ \lambda(\mu;p) \\ \upsilon(\mu;p) \end{bmatrix}$$

Part I

Multistage Nonlinear Model Predictive Control with Data-Driven Scenario Selection

Chapter 3

Data-driven Online Adaptation of the Scenario Tree in Multistage Nonlinear Model Predictive Control

In this chapter, we consider systems where the uncertainty set $\mathbb{D}$ can change from one time step to the next. An online data-driven scenario tree adaptation strategy is proposed for systems with such time-varying uncertainty information, that can be performed dynamically in the multistage NMPC implementation. The idea is to use a scenario selection approach based on principal component analysis (PCA) whenever the uncertainty set $\mathbb{D}$ changes, and to dynamically change the length of the robust horizon N_r in anticipation of any predicted change in $\mathbb{D}$. This chapter is adapted from [29], and extends upon some preliminary work done in [54].

The following sections elaborate on the proposed strategy, and show its application on a simple thermal energy storage (TES) example. In particular, it is shown that the PCA-based approach leads to faster computations and solutions that are less conservative than the conventional scenario selection approach.

3.1 Motivating the need for data-driven scenario selection

The convention in multistage NMPC is to assume that the uncertainty set $\mathbb{D}$ is known *a-priori*, and that it can be well represented by discrete scenarios such as in (2.3). However, the selection of scenarios that build the scenario tree is important in the practical implementation of multistage NMPC.

Stochastic programming methods assume that the uncertain parameters follow a probability distribution function (PDF) within a finite set of values [4]. Conventional scenario-based stochastic programming methods involve a two step process:

1. using statistical methods to estimate the PDF from a given finite set of data samples representing the uncertain parameters, and

2. discretizing the obtained PDF to generate scenarios to be used in the optimization problem.

The main issue with this process is that the first step aims to achieve maximum estimation accuracy without directly considering the optimization problem. Consequently, the PDF is itself subject to uncertainty. In this context, the paradigm of so-called distributionally robust optimization methods has been developed to deal with the issue of uncertainty-affected PDFs [55, 56].

Alternatively, one can skip the PDF estimation step and go directly from data to scenarios, i.e. the discrete scenarios can directly be chosen from the available data samples. After all, the different data samples represent the discrete measurements of uncertainty in the system. Ideally, then, any set of selected scenarios should be a subset of this data set for the best representation of uncertainty.

Having decided on selecting scenarios directly from available data, the next question to consider is *which* data samples to select as scenarios. As noted before, the size of the multistage NMPC problem (2.4) increases exponentially with increasing number of scenarios. Hence, in order to be computationally efficient, it is important to capture maximum (and the most representative) uncertainty information with minimum number of scenarios.

In systems with multidimensional parametric uncertainty, the uncertain parameters often exhibit correlations. This is especially true for chemical process systems where disturbances in temperatures or flow rates in different sections of the plant are usually correlated. Sampling methods like the monte carlo or the latin hypercube sampling emphasize

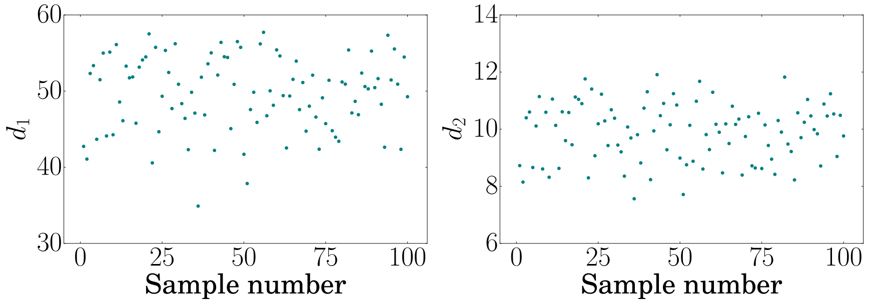


Figure 3.1: Univariate plots of sampled data for d_1 and d_2

randomness of sampling to maximize information, but ignore such correlations. Similarly, selecting scenarios according to a heuristic like the one shown in (2.3) also assumes that the parameters are independent of each other. Therefore, these scenario selection methods may not be the best if we want to exploit the uncertainty correlations to reduce the number of scenarios.

To overcome this, multivariate data-analysis methods can be used to detect any hidden correlations within the available data samples. The scenarios chosen using these multivariate methods explicitly take into account the interdependence between the parameters. Dimensionality reduction methods such as PCA explain the parametric variation in a data set in fewer dimensions. This means that lesser number of scenarios are able to effectively describe the parametric variation in the system, leading to a compact scenario tree formulation in multistage NMPC. It must be stated that the combination of MPC with PCA has been proposed previously [57, 58, 59], albeit not in the context of multistage NMPC or scenario selection.

In addition to improving computational efficiency, a further advantage of a compact scenario tree formulation is that the corresponding multistage NMPC formulation leads to less conservative solutions. Note that reducing conservativeness through uncertainty range reduction has also been considered in [21, 22].

3.1.1 Principal component analysis for scenario selection

Consider a system with two uncertain parameters ($n_d = 2$), that are represented by their respective data sets as shown in Figure 3.1. Based on the conventional scenario selection heuristic (2.3), the extreme and nominal values of the parameters are considered. Another similar heuristic to select scenarios, which we term as the "BOX" approach, is based on the following set of discrete realizations:

$$\mathbb{M}^{BOX} = \left(\{d_1^{\mathsf{max}}, d_1^{\mathsf{min}}\} \times \cdots \times \{d_{n_d}^{\mathsf{max}}, d_{n_d}^{\mathsf{min}}\}\right) \cup \mathbf{d}^{\mathsf{nom}} \tag{3.1}$$

However, such an approach interprets the parameter data independently and gives no indication of whether the two parameters are correlated to each other.

PCA employs a mathematical procedure that transforms a data set with multiple, possibly correlated, parameters into a lesser number of uncorrelated parameters, characterized by principal components. To be precise, PCA can be obtained through the eigenvalue decomposition of the data covariance matrix, or alternatively through the singular value decomposition of the (centered) data matrix (see [60] for further details on the PCA algorithm). Essentially, it is an orthogonal linear transformation of the data set into a new coordinate system with each new axis representing a principal component. The first principal component points in the direction of maximum variance within the data set. Subsequent principal components account for as much of the remaining variance as possible, in decreasing order. This dimensionality reduction helps explain the parametric variation in the data using smaller number of components.

Consider a data matrix $\mathbf{X} \in \mathbb{R}^{n_o} \times \mathbb{R}^{n_d}$, where rows of the matrix represent observations and columns represent the (possibly correlated) parameters. To remove arbitrary biases from the measurements, the data is mean-centered and scaled, resulting in the data matrix $\mathbf{X}_{sc} \in \mathbb{R}^{n_o} \times \mathbb{R}^{n_d}$. Performing PCA on $\mathbf{X}_{sc}$ results in the output $\mathbf{Y} \in \mathbb{R}^{n_o} \times \mathbb{R}^{n_{d'}}$, with $n_{d'} \leq n_d$, according to:

$$\mathbf{Y} = \mathbf{X}_{sc}\mathbf{P} \tag{3.2}$$

where, $\mathbf{P} \in \mathbb{R}^{n_d} \times \mathbb{R}^{n_{d'}}$ is the projection matrix with each column representing a principal component. In other words, each column of $\mathbf{P}$ contains the coefficients that project the original data point to the new coordinate system ($\mathbf{Y}$) of $n_{d'}$ principal components. These are also referred to as *loadings*. The matrix $\mathbf{Y}$ is called the *scores* matrix. The score of a data point along a principal component represents the distance of that data point from the mean along the direction of that principal component.

Scenarios can be chosen by leveraging information from this transformed data set. Since the principal components are orthogonal to each other, scenarios can be chosen along the direction of these principal components to obtain maximum uncertainty infor-

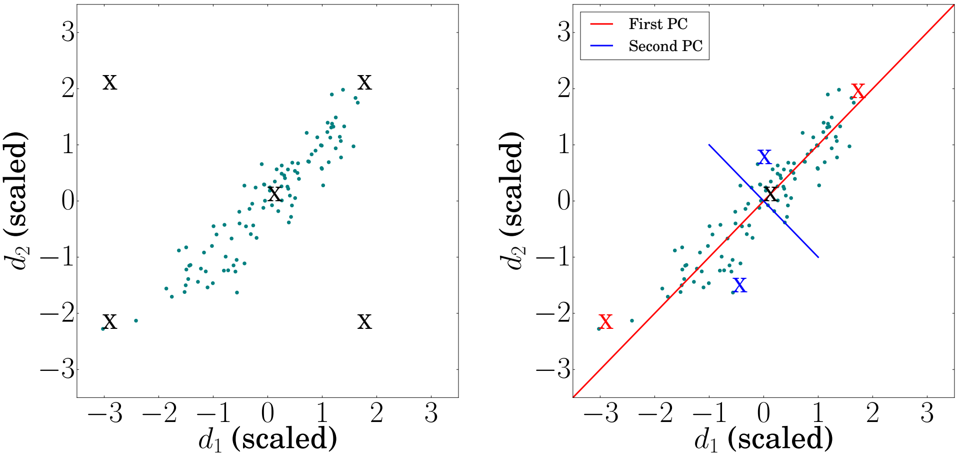


Figure 3.2: Multivariate plot of the sampled data with the BOX (left) and PCA (right) scenario selection. Selected scenarios are marked by ‘X’.

mation. Selecting scenarios along $n_{d'}$ principal components explicitly accounts for any hidden linear correlations between the original n_d uncertain parameters. In particular, we select scenarios that correspond to the maximum and minimum scores along the principal components. This heuristic is based on the following set of discrete realizations:

$$\mathbb{M}^{PCA} = \{d'^{\text{max}}_{PC(1)}, d'^{\text{min}}_{PC(1)}\} \cup \cdots \cup \{d'^{\text{max}}_{PC(n_{d'})}, d'^{\text{min}}_{PC(n_{d'})}\} \,\cup\, \mathbf{d}^{\text{nom}} \tag{3.3}$$

where d' are in the transformed axes. The scenarios are obtained projecting these coordinates back on the original axes. The benefit of using this scenario selection approach becomes apparent when the uncertain parameters are shown on a multivariate plot. As shown in Figure 3.2, the scenarios selected by PCA are much more representative of the sampled data than those selected by the BOX approach. This is because PCA is able to detect correlations in the sampled data and explain the uncertainty more compactly with the principal components. A more compact uncertainty representation leads to reduced conservativeness of the multistage NMPC solution. The corresponding scores plot is shown in Figure 3.3.

Note that the PCA may result in principal components such that some components dominate over the others, in terms of how much data variability they explain. We then propose to select scenarios only along these dominant principal components, since the chosen scenarios can then account for maximum variation in the uncertainty in fewer di-

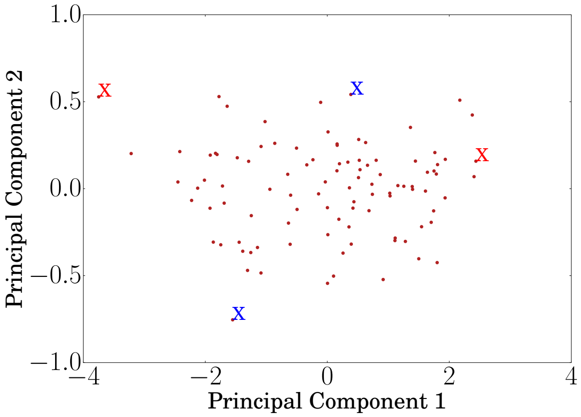


Figure 3.3: Scores plot obtained from PCA. The maximum and minimum scores along the two principal components are marked by 'X's.

mensions. For instance, the plot on the right in Figure 3.2 shows scenarios (marked by red and blue 'X's) selected along both the principal components. Instead, since it can be seen clearly that the first principal component is dominant[1], scenarios can be chosen only in that direction (marked by red 'X's). Thus, instead of choosing 4 scenarios, only 2 scenarios can encompass most of the parametric variation in the data shown in Figure 3.2, without any significant loss in explained variability. Reducing the number of scenarios in this manner can thus make the size of resulting multistage MPC problem significantly smaller, reducing the computational effort.

The number of principal components considered to be dominant can rely on a suitable heuristic for explained variance. In general, out of the obtained $n_{d'}$ principal components, the first n_{d^*} components that explain at least a certain fraction ξ of variance can be considered. A high value of ξ can be chosen (for example 80% or 90%) depending on the application.

3.2 Dynamically adjusting the scenario tree

In this chapter, we take into account that the uncertainty set $\mathbb{D}$, or the sampled data it represents, may change during the operation of the system. That is, the uncertainty may have very different characteristics during different points in time. Consequently the scenarios

[1] For the data shown in Figure 3.2, the first principal component explains 96.4$\%$ of the variance.

selected from $\mathbb{D}$ also change with time.

To reflect these changes, we propose to update the scenario tree dynamically to acknowledge this time-varying nature of $\mathbb{D}$. An online PCA-based scenario selection approach is employed whenever new uncertainty data becomes available. Moreover, we propose that if the parametric variation can be explained by a small number of "dominant" principal components (shown by PCA), it suffices to select the scenarios only along these components to sufficiently explain the uncertainty. The resulting PCA-based multistage NMPC formulation can be stated as:

$$\min_{\mathbf{z}_l^c, \mathbf{v}_l^c} \quad \sum_{c \in \mathbb{C}_k^{PCA}} \omega^c \Big(\phi(\mathbf{z}_N^c, \mathbf{d}_{N-1}^c) + \sum_{l=0}^{N-1} \varphi(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^c) \Big) \tag{3.4a}$$

$$\text{s.t.} \quad \mathbf{z}_{l+1}^c = \mathbf{f}(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^c) \quad l = 0, \dots, N-1 \tag{3.4b}$$

$$\mathbf{z}_0^c = \mathbf{x}_k \tag{3.4c}$$

$$\mathbf{v}_l^c = \mathbf{v}_l^{c'} \quad \{(c, c') \mid \mathbf{z}_l^c = \mathbf{z}_l^{c'}\} \tag{3.4d}$$

$$\mathbf{z}_l^c \in \mathbb{X}, \mathbf{v}_l^c \in \mathbb{U}, \mathbf{z}_N^c \in \mathbb{X}_f, \mathbf{d}_l^c \in \mathbb{D}_k \tag{3.4e}$$

$$\mathbf{d}_{l-1}^c = \mathbf{d}_l^c \quad l = N_r, \dots N-1 \tag{3.4f}$$

$$\forall c, c' \in \mathbb{C}_k^{PCA}$$

where $\mathbb{C}_k^{PCA}$ is the set of scenarios, formed from the set of discrete uncertainty realizations $\mathbb{M}_k^{PCA}$, taken from the uncertainty set $\mathbb{D}_k$. Note that $\mathbb{C}_k^{PCA}$, $\mathbb{M}_k^{PCA}$ and $\mathbb{D}_k$ can change from one time step to next.

Further, we propose to extend the length of the robust horizon dynamically in anticipation of a predicted change in uncertainty data set. This can provide an additional hedge against uncertainty, since the "latest" information is constantly being used to select the scenarios further along the scenario tree. If it is known *a priori* that new sampled data will represent uncertainty information at a future time step in the horizon, the robust horizon can be accordingly modified to take this into account. Farther way from the point of data update, a shorter robust horizon can be used to reduce computational burden. As the predicted point of data update comes closer, the robust horizon can be extended to include the new scenarios reflecting the update. For instance, consider the scenario tree shown in Figure 2.2. Here $N_r = 2$ at time t_k. Now, if new data samples explaining the uncertainty for time t_{k+2} become available at time t_k, then the robust horizon can be increased from

$N_r = 2$ to $N_r = 3$ at time t_k, in order to accommodate the extra branching (based on newly available sampled data) at time t_{k+2}. This would of course lead to consideration of additional scenarios (in this case, $3^3 = 27$ scenarios), which increases the computational effort in solving the problem. Thus a trade-off needs to be made between incorporating extra information about future uncertainty and increasing computational complexity, depending on the given application.

It is important to note in this framework that it is the new *data set* describing the expected uncertainty at t_{k+2} (i.e. $\mathbb{D}_{k+2}$) that becomes known at t_k. The exact future *realization* of the uncertainty at t_{k+2} (i.e. $\mathbf{d}_{k+2}$) is obviously not known at t_k. However, with the knowledge of $\mathbb{D}_{k+2}$ at t_k, we can extend the robust horizon from $N_r = 2$ to $N_r = 3$ at t_k itself, in order to capture this extra information. We can then perform PCA online for the $\mathbb{D}_{k+2}$ data set, again at t_k itself, to get the extra branches in the scenario tree resulting from this change in the robust horizon from $N_r = 2$ to $N_r = 3$. In this way, the updates of scenarios using PCA and changes in the length of the robust horizon can be performed online at any time step where new information about future uncertainty data presents itself.

3.3 Case study

We consider a hypothetical two-plant TES system, with one plant being the supplier of heat (S) and the other being the consumer (C). A thermal storage tank acts as a buffer between the two plants to facilitate the energy exchange. The tank interacts with the two plants via heat exchangers, as shown in Figure 3.4. Further, the tank can directly be heated up via a limited, local heating source that is considered inexpensive. If the energy in the tank is insufficient to meet the energy requirements on the demand side (for example, during peak demands), the consumer plant C has to purchase the excess energy from the market. Energy from the market, usually in the form of peak heating sources like fossil fuel burners or electricity, is much more expensive than the local heating source. The objective is to operate the system such that the total cost of energy purchase is minimized.

The process temperatures from the supplier and consumer plants are $T_S^{process}$ and $T_C^{process}$ respectively. Temperatures on the hot and cold sides of heat exchangers on both sides are T_S^{hot}, T_S^{cold}, T_C^{hot} and T_C^{cold} respectively. The tank temperature is T^{TES}, whereas the ambient temperature is T^{amb}. The flow rates on either side of the heat exchangers are

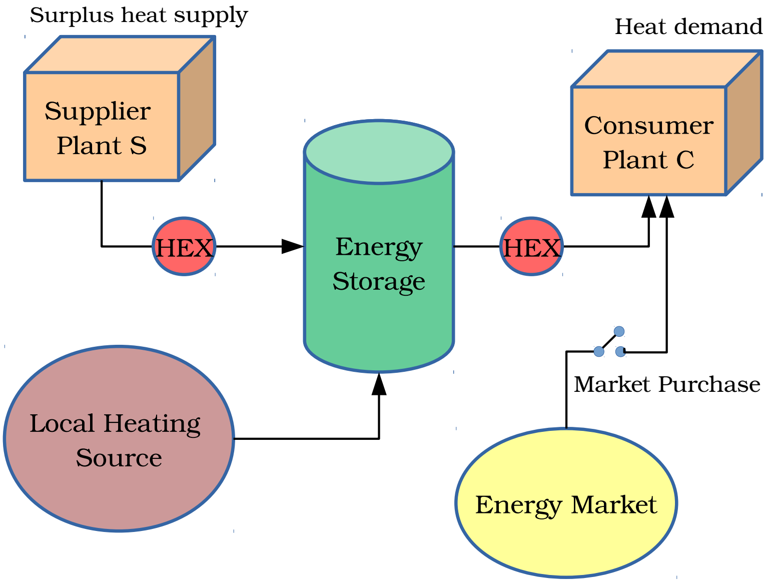


Figure 3.4: Illustration of a simple energy storage system.

q_S^{hot}, q_S^{cold}, q_C^{hot} and q_C^{cold} respectively. Local heat supply is Q^{TES} and the peak heating sourced from the market is Q_C^{peak}. Consumer side demand is denoted by Q_C^{demand}. Table 3.1 shows the states, inputs and uncertainties in the system. Further details regarding the process model and data are given in Appendix A. We consider hot water as the fluid for heat exchange.

As noted, the uncertainties in the system are the process temperatures of the supplier and the consumer. Such temperature data from a process is usually logged. A period of 24 hours is considered for this uncertainty data. Further, we consider that the process temperature distributions are different for three distinct phases of the day - 12 am to 8 am (night phase), 8 am to 4 pm (day phase) and 4 pm to 12 am (evening phase). We use synthetic data for the temperature distributions for these phases. The plants operate at higher temperatures during the day phase and lower temperatures during the evening and night phases. Further, it is reasonable to expect that these plant temperatures are correlated to each other, since the periods of high and low activity in process plants in an industrial cluster are similar. The scatter plot of the two process temperatures is shown in Figure 3.5. For implementing the multistage NMPC (2.4), the economic stage cost is considered to be

Table 3.1: Simple TES model: states, inputs, and uncertainties.

Symbol	Description	Unit
States		
T_S^{hot}, T_S^{cold}	Supplier-side temperatures	$^\circ C$
T_C^{hot}, T_C^{cold}	Consumer-side temperatures	$^\circ C$
T^{TES}	Temperature of the TES	$^\circ C$
Inputs		
q_S^{hot}, q_S^{cold}	Volumetric flow rates on the supplier-side HEX	m^3/s
q_C^{hot}, q_C^{cold}	Volumetric flow rates on the consumer-side HEX	m^3/s
Q^{TES}	Local heating for TES	kW
Q_C^{peak}	Peak heating rate for consumer	kW
Uncertainties		
$T_S^{process}$	Supplier process temperature	$^\circ C$
$T_C^{process}$	Consumer process temperature	$^\circ C$

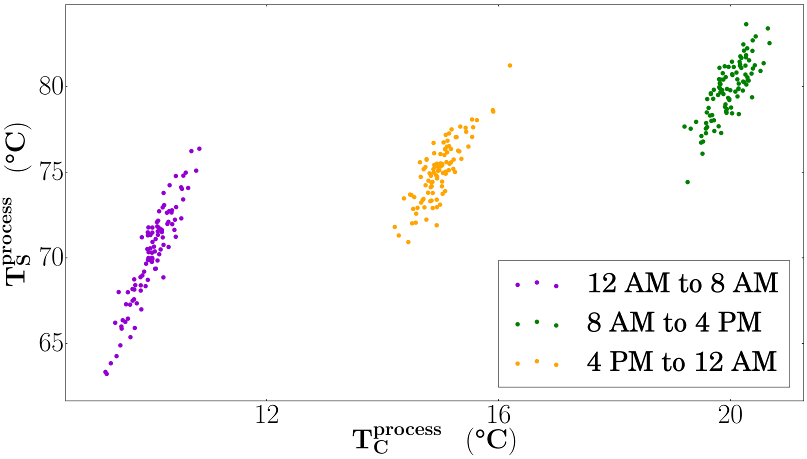


Figure 3.5: Uncertainty data - process temperatures from the plants for the different phases. It can be seen that the temperatures are correlated differently at different phases of the day.

the cost of purchased energy, which we want to minimize [2]:

$$\psi_l = P^{TES}(Q^{TES})_l + P^{peak}(Q_C^{peak})_l \tag{3.5}$$

Model parameters are given in Table A.1, and bounds for the states and inputs are given in Table A.2.

The process dynamics are discretized using third-order Radau collocation over finite elements. The multistage NMPC problem is formulated with $N = 24$ hours (finite elements) and $N_r = 1$ hour, with control action changing every hour. `JuMP` [62] (version 0.18.2), a modeling tool within the framework of `Julia` [63] (version 0.6.2) programming language, is used to implement the multistage NMPC problem. The resulting NLP is solved using `IPOPT` [53], with MA27 [79] as the linear solver. The results are divided into the following two parts:

1. Comparison of a scenario selection between the data-driven PCA approach and the conventional BOX approach, with a constant $N_r = 1$.

2. Studying the effect of dynamically adjusting N_r closer to an anticipated change in the uncertainty data, while using PCA for scenario selection.

3.3.1 Data-driven vs conventional scenario selection

For comparison, the dynamic scenario selection is done with two methods. In the first method, the conservative BOX approach is used, selecting scenarios as the four corner points from the box that encompasses all the uncertainty data over each 8-hour phase during the day. Essentially, these scenarios represent the combinations of the minimum and maximum of the data set along each dimension, along with the nominal scenario, as shown in heuristic (3.1). As noted, the BOX method does not account for any correlations between the uncertain inlet temperatures.

In the second, PCA method, the scenarios are chosen by performing PCA over the data sets that are relevant for the corresponding phases of the day (shown by different colors in Figure 3.5). The PCA results in two principal components for each data set, with the first principal component explaining 96.51% of the total variability for the night phase data,

[2] For practical implementation purposes, we assume that our prediction horizon is long enough and do not consider terminal conditions (see [61], for example).

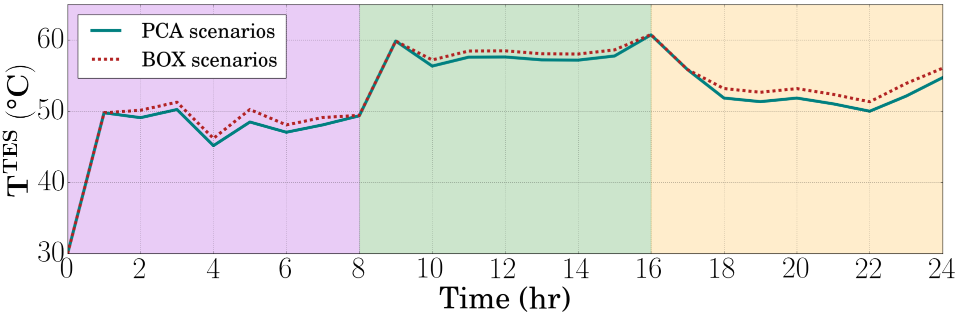


Figure 3.6: The TES tank temperature across the 24-hour period, for the BOX and PCA methods.

89.72% for the day phase data, and 91.84% for the evening phase data. Since the first principal components explain a large fraction of the variance in the data, two scenarios are selected corresponding to the minimum and maximum scores along this principal component only. Following the heuristic (3.3), a third scenario is chosen to represent the mean value along each dimension.

We consider the uncertainty to be time-varying over the MPC horizon, where the "true" realization of the uncertainty in the plants is chosen randomly for each hour from the corresponding data set. The simulation is considered for the 24 hour period from 12 am to 12 am. The demand is constant at 5 MW throughout the day, except for 7 am to 9 am and 3 pm to 5 pm, when there is a peak demand of 10 MW.

The results of the optimization are shown in Figures 3.6 and 3.7. The colored background shades in the plots relate to the corresponding phase of the day, and have a color coding that matches the one in Figure 3.5 It can be clearly seen that for the BOX method, the solution is more conservative. Figure 3.6 shows that the TES tank is heated to a higher temperature for satisfying the same demand profile across the day. As shown in Figure 3.7, the local heating of the tank is more with the BOX method compared to the PCA method. Similarly, the data-driven approach leads to lower purchases of the expensive energy from the market during peak demands (7 am to 9 am and 3 pm to 5 pm).

Moreover, the simulations were repeated 30 times for the time-varying uncertainty case. The "true" set of uncertain parameters in each simulation was a randomly chosen subset of the available data set. The performance for each simulation run was evaluated based on the integrated objective function, which sums up the values of the objective cost for all stages and scenarios. The results are presented in Figure 3.8, where it can be

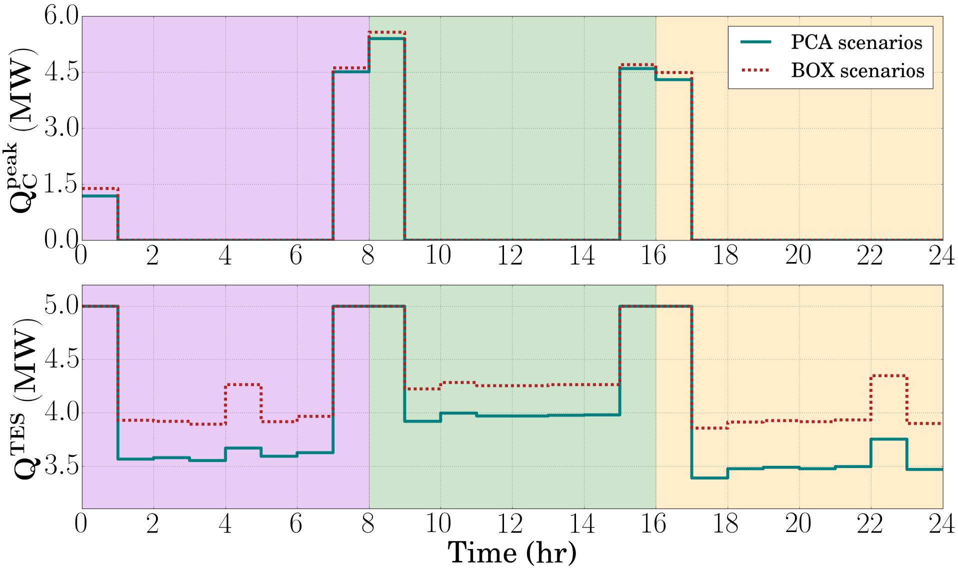


Figure 3.7: The energy purchased from the market and the local heating source across the 24-hour period, for the BOX and PCA methods.

seen that the PCA method outperforms the BOX method with a lower cost. Note that we consider only 3 scenarios in the PCA-based method, compared to 5 scenarios in the BOX method. To this end, the integrated objective costs are divided by the respective number of scenarios chosen for each method for a fairer comparison. The reason for a lower cost is that the scenarios chosen via PCA encompass the uncertainty more compactly, and are thus likely to be closer to the actual realization of the uncertainty.

Since the PCA-based method enjoys a much smaller problem size, the computation

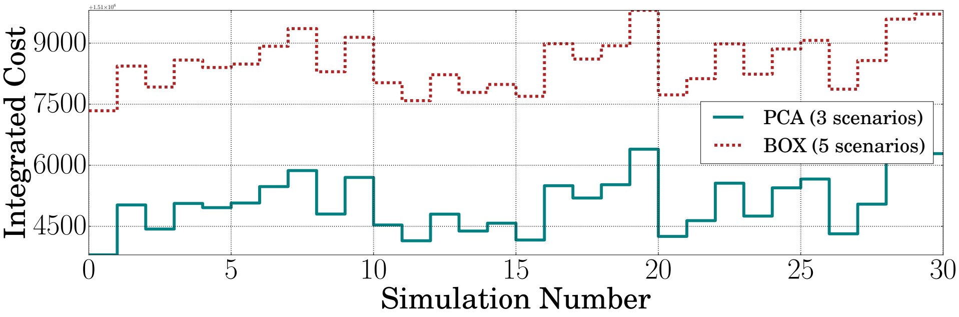


Figure 3.8: The averaged integrated cost for 30 different simulation runs.

time for the BOX method is significantly greater than for the PCA method. The BOX and the PCA methods result in NLPs with 4320 and 2592 variables respectively. Comparing the mean computation times for both methods reveals that the PCA-based scenario selection is about 48% faster than the conventional approach, on average.

3.3.2 Dynamically adjusting the robust horizon

Here, the simulations are run such that the multistage NMPC is implemented using two N_r cases. In the first case, the robust horizon is kept constant throughout the simulation at $N_r = 1$. This is referred to as the constant N_r case.

In the second case, the robust horizon is dynamically adjusted (switched) from $N_r = 1$ to $N_r = 2$, *one hour before* the night-to-day and day-to-evening phase changes. This change is implemented only for the corresponding next one time step, and is subsequently reduced back to $N_r = 1$ for later time steps. Thus, $N_r = 2$ is used for the simulation at 7 am and 3 pm. This is because it is known that, at these times, the uncertainty data will be updated in one hour due to change in phase; and the robust horizon length of 2 hours reflects this. This is denoted as the dynamic N_r case.

The scenario selection is done via PCA for both cases. Moreover, the energy demand is considered to be constant at 5 MW throughout the 24-hour period (i.e. no peak demand). The results are shown in Figures 3.9 and 3.10.

It can be seen that by dynamically extending the robust horizon (dynamic N_r case), the optimization anticipates the upcoming rise in process temperatures by preempting the local tank heating at 7 am (shown by a higher Q^{TES} in the dynamic N_r case than in the constant N_r case). This can also be seen from the tank temperature profile, where the temperature of the tank rises higher in the dynamic N_r case at 7 am than in the constant N_r case. Consequently, Q_C^{peak} at 8 am is smaller in the dynamic N_r case, leading to lower cost. During the 4 pm phase change, the temperatures are dropping anyway so market purchase is unnecessary. In other words, the tank has enough energy during this phase to satisfy the energy demand of the consumer. This leads to the same temperature and heating profiles in the evening phase for both constant and dynamic N_r cases.

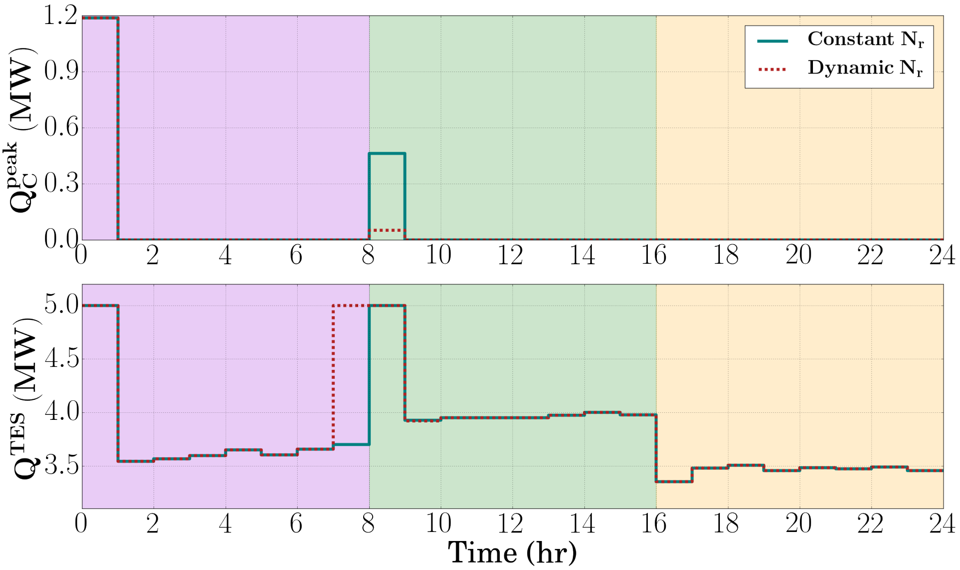


Figure 3.9: The TES tank temperature across the 24-hour period, for the constant and dynamic N_r cases.

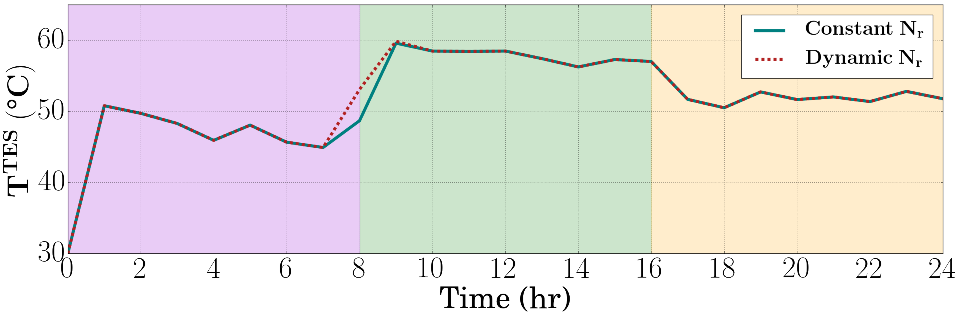


Figure 3.10: The energy purchased from the market and the local heating source across the 24-hour period, for the constant and dynamic N_r cases.

3.4 Conclusion

The TES case study demonstrates that the same energy demand profile can be satisfied by heating the tank *less* if the scenario selection is data-driven and dynamic. Not only does the tank operate at a lower temperature, but the cost of operation is also significantly lower.

In addition, extending the robust horizon dynamically leads to the consideration of future changes in process temperatures by the multistage NMPC algorithm. This prompts preemptive control action so that the tank is heated up in anticipation even before the uncertainty data changes. The result is that the expensive market purchase is reduced when the energy is demanded at a higher process temperature.

To conclude, we have demonstrated that an online PCA-based, dynamic scenario tree adaptation approach leads to solutions that are less conservative while still hedging against the uncertainty. Moreover, the approach involves solving an optimization problem of a smaller size since fewer scenarios, chosen only along the dominant principal component, are needed to describe the uncertainty.

Chapter 4

Data-Driven Robust Optimal Operation of Thermal Energy Storage in Industrial Clusters

The industrial sector is responsible for releasing large quantities of waste heat to the environment - via cooling media, exhaust gases, and hot equipment surfaces, among others. This is especially true for chemical, minerals and metals, pulp and paper, and the food industries [64]. Recovering, storing, and reusing this surplus heat when required is a means of improving industrial energy efficiency and lowering its overall environmental impact. Moreover, waste heat recovery presents a low-cost energy resource with the potential to realize significant energy savings, improving the competitiveness of the sector.

Surplus heat recovery is an especially attractive proposition for industrial clusters since it presents an opportunity for flexible energy exchange within the cluster, with plants representing both sources and sinks of surplus heat. Today, recovered high-grade heat is mostly reused within the process itself in order to reduce exergy losses, but low-grade surplus heat below 200 $^{\circ}C$ provides recovery potential for external energy exchange [65]. Such low-grade heat integration is thus highly beneficial for energy exchange in industrial clusters, since participating plants can fulfill their low-grade heat demands from within the cluster itself.

A prominent operational challenge for surplus heat exchange between multiple plants is the temporal decoupling between the availability of surplus heat and its demand [66]. From the various available technologies [67], thermal energy storage is a viable option to handle this issue. Surplus heat is supplied to, and extracted from, a TES unit at different temperatures, allowing for a diverse set of plants to participate in the energy exchange. TES units offer operating flexibility to heat recovery by creating a buffer between the supply and demand of surplus heat, thereby reducing peak energy requirements. With TES, however, the dynamic operational aspects become extremely important.

Optimal operation and control of TES units has been studied previously within different application areas. For instance, optimal operation of TES units in buildings for reducing peak loads was considered by [68]. In [69, 70], an MPC scheme is implemented on a TES system for energy-efficient buildings. Another application of MPC on a TES system for multi-energy district boilers is investigated in [71]. Classical control techniques like the PID-control have also been investigated, for instance in TES for concentrated solar polar [72]. An optimal control scheme was proposed by [73] for surplus heat exchange using TES in industrial clusters. A review of various types of TES and their applications can be found in [74, 75], whereas a review of optimization and control for TES systems is given in [76].

A TES unit within an industrial cluster setting presents a unique challenge from an operator's perspective, in that there are various system uncertainties to contend with. In [77], it is noted that supply of low-grade surplus heat is usually unstable because of temperature fluctuations in the processes producing it. Similarly, there is volatility in consumer-side heat demands and process temperatures. These uncertainties in the supply and demand profiles of surplus heat affect efficient utilization of the TES unit. Disregarding these uncertainties in the system can lead to control solutions that are not only suboptimal, but also infeasible. The plant-model mismatch that arises due to these uncertainties may result in control solutions that cause critical constraint violations. In an industrial cluster, this may translate to larger heat-acquisition costs to satisfy consumer demands, leading to high carbon emissions if fossil fuel-based peak heating is involved. Operating constraints relating to safety or process specifications may also be violated. Such constraint violations usually come with a significant economic penalty, and so may not be acceptable to the stakeholders in the industrial cluster.

In this chapter, we argue that for optimally operating a TES unit in an industrial cluster, uncertainty in heat supply and demand must be taken into account rigorously. To this end, we propose the use of multistage NMPC to achieve robust operation. The availability of historical data from industrial processes motivates the need for data-driven approaches for scenario selection in multistage NMPC. We show in this chapter that the inherent correlations present in the heat supply and demand profiles can be exploited using the data-driven PCA approach presented in Chapter 3.

Chapter 3 addresses the computational efficiency and reduced conservativeness aspect of data-driven scenario selection in multistage NMPC. Here we address the robustness aspect, and consider its application on a more detailed TES system using industrial heat supply and demand data sourced from a district heating company in northern Norway. In particular, a comparison is made between a standard NMPC formulation where only the nominal uncertainty is considered, and the multistage NMPC scheme with data-driven scenario selection. We demonstrate that the latter formulation leads to an efficient utilization of the TES unit by avoiding violations of critical operating constraints specified by the stakeholders in the industrial cluster. The results presented in this chapter haven been published in [30].

4.1 Case study: system description and modeling

Figure 4.1 illustrates the topology of an industrial cluster with n_S suppliers and n_C consumers of heat, along with a TES as buffer. The TES unit is considered to be a hot water tank. The supplier plants are sources of energy, supplying excess heat to the TES. On the other hand, the consumer plants are heat sinks, extracting energy from the TES to fulfill their energy demands. The heat exchange between the supplier/consumer plants and the TES happens with the help of heat exchangers, with hot water being the energy carrier. The temperatures of the process streams in the supplier and consumer plants drive the heat transfer.

In cases of high volatility with sharp fluctuations in heat demand on the consumer side, it may happen that the energy from the TES is not enough to satisfy any extra demand. A common practice for such an extra-demand scenario is to purchase electricity externally from the market, or to burn fossil fuels using boilers, to heat up the required process streams. Not only are these peak heating sources of energy expensive, significantly increasing the operating costs in the cluster, but they also lead to higher carbon emissions.

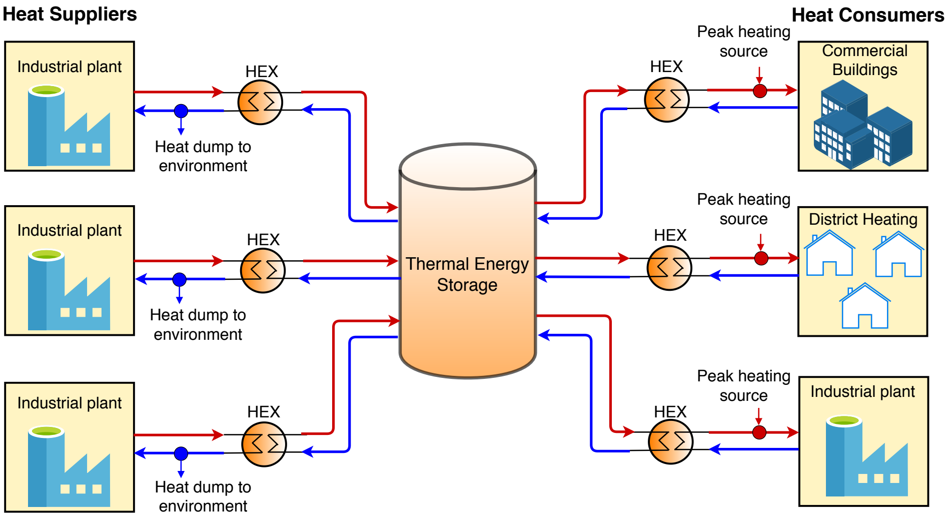


Figure 4.1: Topology of an industrial cluster with multiple energy suppliers and consumers, with a TES unit as buffer. The red lines represent the hot streams and the blue ones cold streams.

The source of surplus heat in the suppliers is assumed to be the cooling of certain batch processes. As such, the suppliers can dump heat to the environment to meet their cooling demands.

We focus on modeling the dynamics of the heat exchange between the TES, and one pair of supplier and consumer plants. The heat exchangers are considered to have counter-current flow, and are discretized in space into a series of n_{hex} cells. Further, we consider that the supplier and consumer plants are close enough to each other such that the heat losses in the pipes due to time delays are assumed to be negligible.

The incoming surplus heat from the supplier is denoted as Q_S^{supply}, and the demand from the consumer is Q_C^{demand}. This heat has to be delivered through the supply-side heat exchanger to the tank. The temperature at which this heat is supplied is denoted by $T_S^{process}$ and the return temperature (after exiting the heat exchanger) is denoted by T_S^{return}. The analogous temperatures on the consumer side are $T_C^{process}$ and T_C^{return}. On the return stream of the supplier, there is a provision to dump heat so as to reduce the return temperature to a desired level, denoted by Q_S^{dump}. Similarly on the consumer side, there is a provision to heat up the return stream using the expensive peak heating to satisfy required demand, denoted by Q_C^{peak}. The temperatures on the hot and cold sides of both heat exchangers are denoted by $T_{S,k}^{hex,hot}$, $T_{S,k}^{hex,cold}$, $T_{C,k}^{hex,hot}$ and $T_{C,k}^{hex,cold}$. Here

$k \in \{1, 2, \ldots, n_{hex}\}$ is the cell index of the corresponding heat exchanger. The volumetric flow rates on the hot and cold sides of both heat exchangers are denoted by q_S^{hot}, q_S^{cold}, q_C^{hot} and q_C^{cold}.

Typically, a large hot water tank will have a temperature gradient along its height leading to different densities at the top and the bottom of the tank. In this work, however, modeling of such thermal stratification is not considered. Instead, the tank is assumed to have uniform mixing throughout its volume, such that the temperature within the tank is same as the tank outlet temperature, denoted by T^{TES}. Since the temperature ranges considered are below the boiling temperature of water, we assume an unpressurized tank. Moreover, T^{amb} is the ambient temperature. The detailed model equations are given in Appendix B.

Dynamic operational optimization of the energy storage requires a central cluster operator to apply optimal control inputs to the system such that the heating and cooling requirements of the plants in the industry cluster is satisfied. Table 4.1 shows the states, inputs, and uncertainties in the system. The volumetric flow rates on the supplier side, q_S^{hot} and q_S^{cold}, are considered to be fixed, and are not degrees of freedom in the system. The operator should thus determine how much heat to dump on the supplier side, Q_S^{dump}, and how much heat to acquire through the peak heating sources, Q_C^{peak}, on the consumer side. Additionally, the operator must determine how much heat can be extracted from the TES by adjusting the volumetric flow rates through the consumer-side valves, q_C^{hot} and q_C^{cold}.

The economic objective for the operator is to minimize the use of expensive peak heating sources in the industrial cluster. For implementations such as standard NMPC (2.2) and multistage NMPC (2.4), this can be formulated as the stage cost:

$$\psi_l = \left(Q_C^{peak}\right)_l \tag{4.1}$$

The cost (4.1) is an L_1 cost function, which assists in getting sparse solutions [78]. It also ensures that peak heating is only used in cases where the heat demands cannot be fully met by the TES. With a purely economic objective, the idea is for the cluster operator to primarily give directives on how much heat to dump to the surroundings and/or how much peak heating to use, at every hour. We assume that there is a lower regulatory level (e.g., PID control) to handle other process disturbances on a faster time scale - e.g., in temperatures, mass flows, etc.

Table 4.1: Detailed TES model: states, inputs, and uncertainties.

Symbol	Description	Unit
States		
$T_S^{process}$, T_S^{return}	Supplier-side process and return temperatures	$^\circ C$
$T_C^{process}$, T_C^{return}	Consumer-side process and return temperatures	$^\circ C$
$T_{S,k}^{hex,hot}$, $T_{S,k}^{hex,cold}$	Supplier-side HEX temperatures	$^\circ C$
$T_{C,k}^{hex,hot}$, $T_{C,k}^{hex,cold}$	Consumer-side HEX temperatures	$^\circ C$
T^{TES}	Temperature of the TES	$^\circ C$
Inputs		
q_C^{hot}, q_C^{cold}	Volumetric flow rates on the consumer-side HEX	m^3/s
Q_S^{dump}	Heat dump rate for supplier	kW
Q_C^{peak}	Peak heating rate for consumer	kW
Uncertainties		
Q_S^{supply}	Surplus heat supply from supplier	kW
Q_C^{demand}	Surplus heat demand from consumer	kW

An important point to note is that we only consider the application of both the NMPC strategies during dynamic operation of the TES, and not during edge-cases like when the TES is empty at the start. To begin with an empty TES would be equivalent to a start-up phase where all the supplied heat goes to heating up the tank, and all the demanded heat is fulfilled via peak heating. In this case, there are not enough degrees of freedom to improve performance or robustness via NMPC, especially over a shorter horizon.

4.2 Case study: data description

We consider a case study with one supplier and one consumer of heat, exchanging heat via a TES tank. The values of the various system parameters are given in Table B.1. A tank volume of $V^{TES} = 1000\ m^3$ is considered. This approximately corresponds to a cylindrical tank with a height of 10 m and a diameter of 11 m. The heat losses in heat exchangers, and in the pipes carrying the hot water, are considered to be negligible. The operating bounds on the various system variables are given in Table B.2.

The heat supply and demand data used in this case study is based on the 2017 data from Mo Fjernvarme AS, a district heating company which is part of the Mo Industrial Park in northern Norway. The surplus heat in Mo Fjernvarme is sourced from a smelting plant

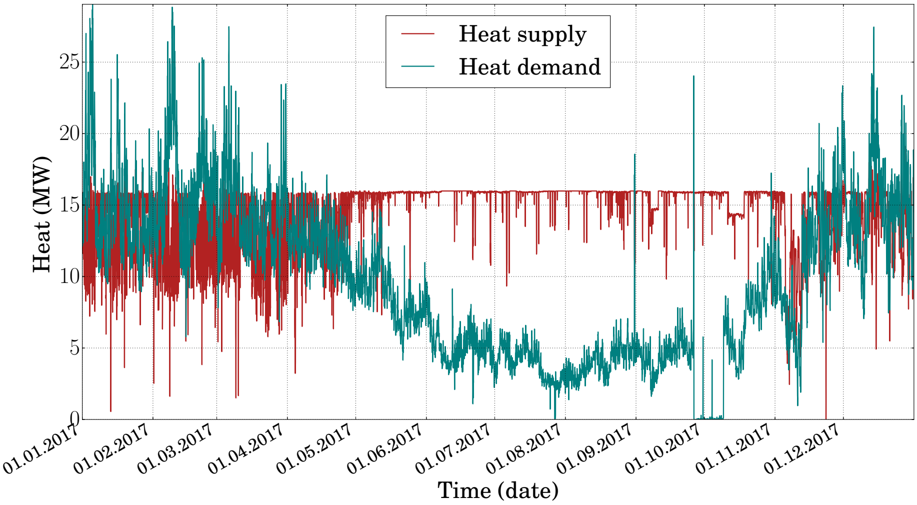


Figure 4.2: The hourly supplied and demanded heat flow rates for 2017.

within the industrial park in form of flue gases. The demand, on the other hand, comes from the town of Mo i Rana and from within the industrial park itself. For simplicity, we do not separately model the various processes within Mo Industrial Park in this work. As such, the models that describe our case study are not directly related to the industry park layout or processes. We instead focus on the overall heat supply and demand data from the industry park for data analysis, along with the detailed TES model shown in Appendix B, to demonstrate the effectiveness of our proposed approach.

The hourly heat supply and demand for 2017 is shown in Figure 4.2. For the spring, summer, and fall months of April to November, it can be seen that the demand is lower. However, for the winter months of January–March and December, there is large variation in the demand compared to a relatively smaller variation in supply. Since optimal operation is considered on a daily basis, a TES unit is not relevant for periods when supply is always higher, since the demand can be completely satisfied by the supplied heat. However, for the winter months, an optimal operation strategy would allow for making TES storage and discharge decisions in order to reduce the peak heating. Moreover, for diurnal thermal storage there must be a period in the day at which supply is greater than demand. Figure 4.3 shows the aggregated heat supply and demand for each month of 2017.

Apart from dimensionality reduction, PCA can be used in data preprocessing to detect

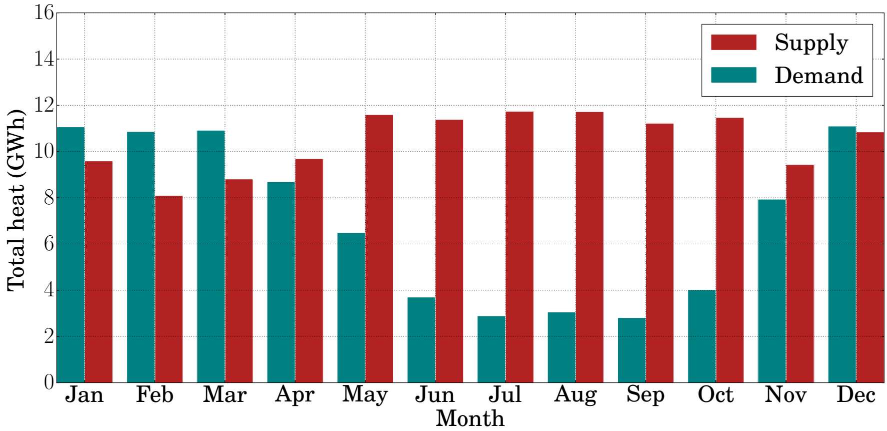


Figure 4.3: The total supplied and demanded heat for each month of 2017.

outliers. These represent instances where the system is either sampled wrongly or is not in normal operation. Statistically, outliers are considered to not be a part of the general sampling population, and are excluded from our analysis to avoid unnecessary conservativeness in scenario selection. We analyzed the demand and supply data the winter months of January–March and December to detect any outliers. For a total of 121 winter days with hourly variation in heat flows, PCA is performed separately in both sets of supply and demand data. Our analysis shows that there were many outliers in the data during the month of December, and hence this month is discarded (see Appendix B for more details on outlier detection). Of the remaining winter months, January shows the least difference between the total supplied heat and the total heat demand, making it a suitable candidate to inspect daily storage. Hence, we focus on the month of January in this work.

Since hourly heat data is available, the January data includes 31 sample points (for 31 days), corresponding to each of the 24 hours of the day. However, our analysis in Figure B.2 shows that the 2nd, 3rd, 4th, 5th, and the 13th days of the month are outliers in terms of either the heat supply or demand. Hence, these are excluded from the analysis. The scatter plot of the data points for each of the 24 hours is shown in Figure 4.4. For most of the day, a linear correlation can be seen for the heat supply and demand. The exceptions are the morning hours of 8–10 am, and the afternoon hours of 3–5 pm. These

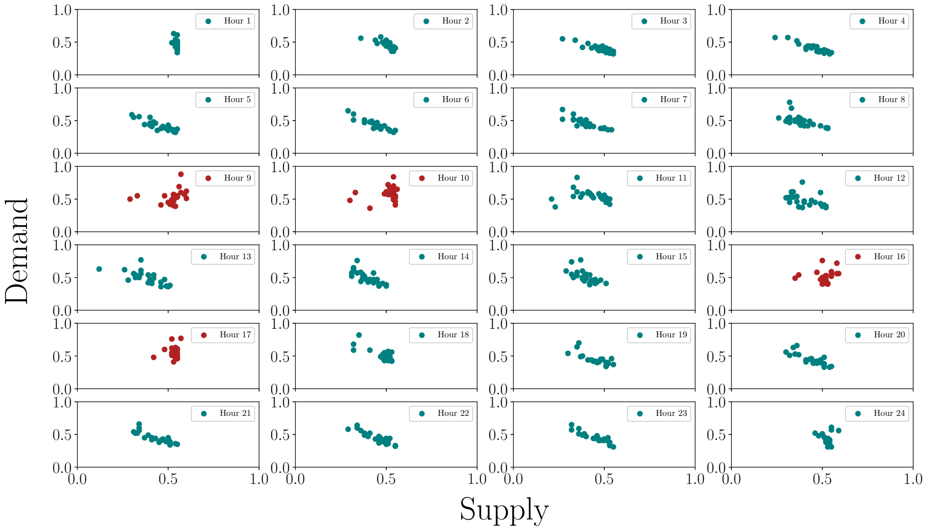


Figure 4.4: Hourly scatter plot of heat supply and demand (normalized) in January 2017. The plots in blue show a linear trend in the data. The plots in red correspond to the peak demand hours, and do not show the linear correlation.

are the peak demand hours, as can be expected of a district heating system. The supply and demand data for these hours are thus not linearly correlated to each other. Figure 4.5 shows the mean hourly demand trend averaged over the winter months of January–March and December. Combined with the analysis in Figure 4.4, it is evident that 8–10 am and 3–5 pm are the expected peak heating hours for January.

The aim is to apply the optimal control strategy for operation during a typical January 2018 day, based on the available January 2017 data shown in Figure 4.4. Applying standard NMPC (2.2) or multistage NMPC (2.4) requires a prediction of the heat supply and demand across the prediction horizon, which is taken to be 24 hours. These values are taken to be the means of the data corresponding to each of the scatter plots in Figure 4.4. Additionally, the multistage NMPC also requires scenario selection for each hour of operation. This scenario selection can be done by performing PCA on the corresponding scatter plots in Figure 4.4, as explained in Chapter 3.1.1. For the non-peak demand hours, since there is a strong linear correlation between the supply and demand, the scenarios are chosen only along the dominant principal component. This implies a total of 3 scenarios, corresponding to the minimum and maximum scores, along with the mean value. For the

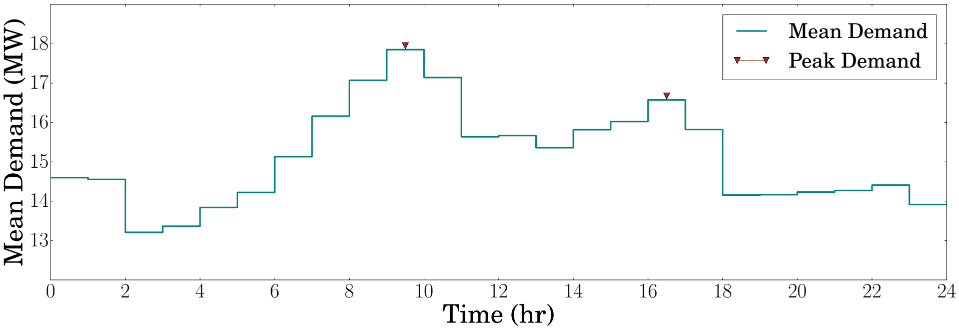


Figure 4.5: Mean hourly demand data for winter months (January–March and December) in 2017 showing expected peak demand hours between 8–10 am and 3–5 pm.

peak hours of 8–10 am and 3–5 pm, the correlation is not strong enough, and thus the scenarios are chosen along the first two principal components to more accurately encompass the uncertainty. The number of scenarios is 5, corresponding to the extreme scores along the two principal components, along with the mean.

4.3 Case study: results

We use third-order Radau collocation over finite elements to approximate the state equations. The inputs and the uncertain parameters are discretized into finite elements, but are considered to be piecewise constant within each element. We consider a prediction horizon of $N = 24\ h$. The finite-dimensional NLP is formulated using 24 finite elements, implying that control action is taken every hour. This also implies that the uncertain parameters are assumed to evolve on an hourly basis, as is also the case with the available industrial data.

The scenarios representing the uncertainty are chosen according to PCA from their respective January 2017 data sets. As explained before, we have $|\mathbb{M}| = 3$ discrete realizations of the uncertainty for every stage of the multistage NMPC problem, in case of the non-peak demand hours. Similarly, we have $|\mathbb{M}| = 5$ discrete realizations for the peak hours. We assume a robust horizon of $N_r = 1$ in this study, giving us either 3 or 5 total scenarios in the multistage NMPC problem, depending on the hour of operation. Further, we consider equal weights for all scenarios.

To evaluate the effectiveness of the PCA-based multistage NMPC formulation (2.4) for this case study, we compare it with a standard NMPC formulation (2.2) where only the nominal uncertainty is considered. To simulate the "true" plant data, we use the same system model as that in the NLP, but the heat supply and demand data are taken from a typical day in January 2018. For comparison, identical data representing the true values of heat supply and heat demand is used in the plant simulation, for both the standard and multistage NMPC methods.

The NLPs (2.2) and (2.4) are modeled using the tool JuMP [62] (version 0.18.5), within the framework of the Julia [63] (version 0.6.2) programming language . The solver used within this framework is IPOPT [53] (version 3.12.8), which uses interior-point algorithms to solve the NLPs. The MA27 [79] linear solver is used within IPOPT.

We focus on the operating constraints in the system for this demonstration. The supply of heat on the supplier side is assumed to come from a batch cooling process, which requires that the return temperature T_S^{return} does not exceed 85 $°C$. As such, any temperatures above the mandated limit may lead to inferior product quality in the batches. Similarly, the district heating network on the consumer side needs the return temperature T_C^{return} to be above 60 $°C$. Therefore, any constraint violation on these return temperatures carries an economic penalty for the respective supplier or consumer.

To investigate the robustness of the methods, the actual heat supply and demand profiles of January 06, 2018 are considered. The expected values in both the NMPC formulations come from the 2017 data, as do the scenarios of multistage NMPC method. The difference in the 2017 and 2018 data creates the plant–model mismatch, and is shown in Figure 4.6.

Figure 4.7 shows the tank temperature, supplier return temperature and consumer return temperature profiles obtained by the application of the two control strategies, given the constraints on the return temperatures. It can be seen that the temperature profiles are higher with multistage NMPC in general. The multistage NMPC keeps the tank heated up and discharges it less than the standard NMPC, even though there is a higher demand. This conservativeness is because the multistage NMPC respects the constraint on consumer-side return temperature T_C^{return} to keep it above 60 $°C$. In contrast, the standard NMPC violates this operating constraint during 8 out of the 24 hours of operation, since it prioritizes the economic objective of demand satisfaction at the cost of constraint violation. The multistage NMPC, having accounted for such a scenario in its scenario tree formulation, anticipates that recourse action is possible in the future time steps. As a consequence,

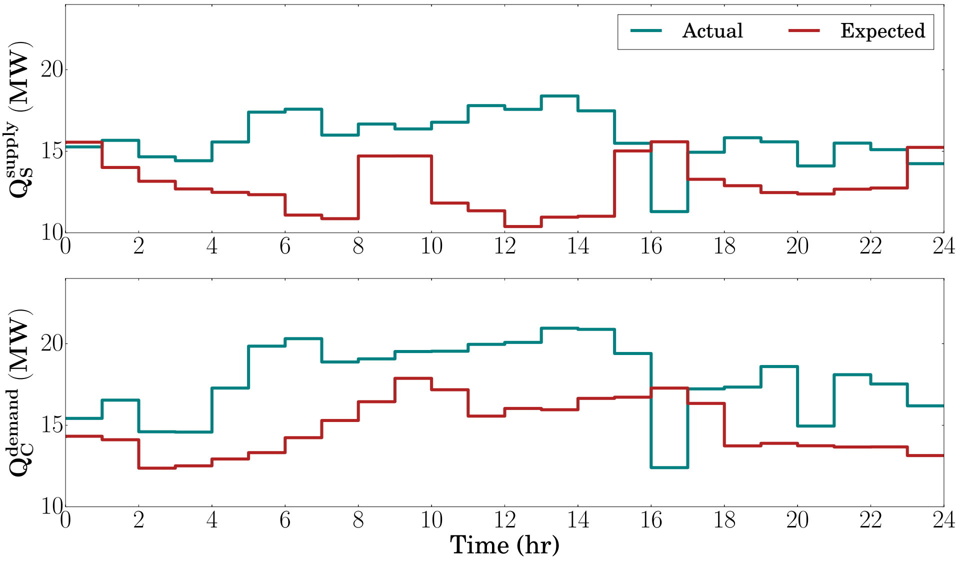


Figure 4.6: The actual and expected heat supply and demand profiles. The actual profiles are for January 06, 2018; the expected profiles represent the average across all days in January 2017.

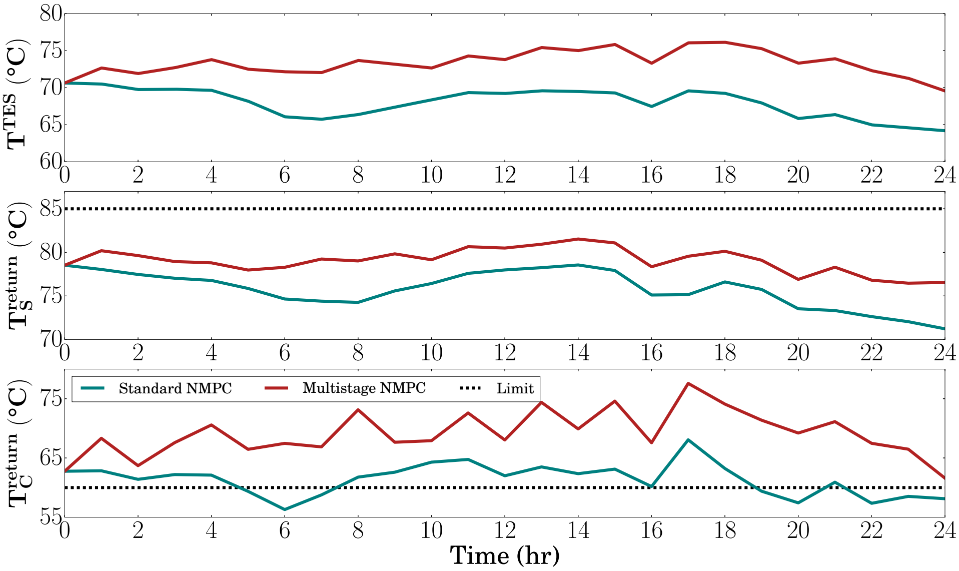


Figure 4.7: The supplier return, consumer return, and tank temperature profiles with the standard and multistage NMPC approaches for January 06, 2018. The constraint limits are shown in dashed black lines.

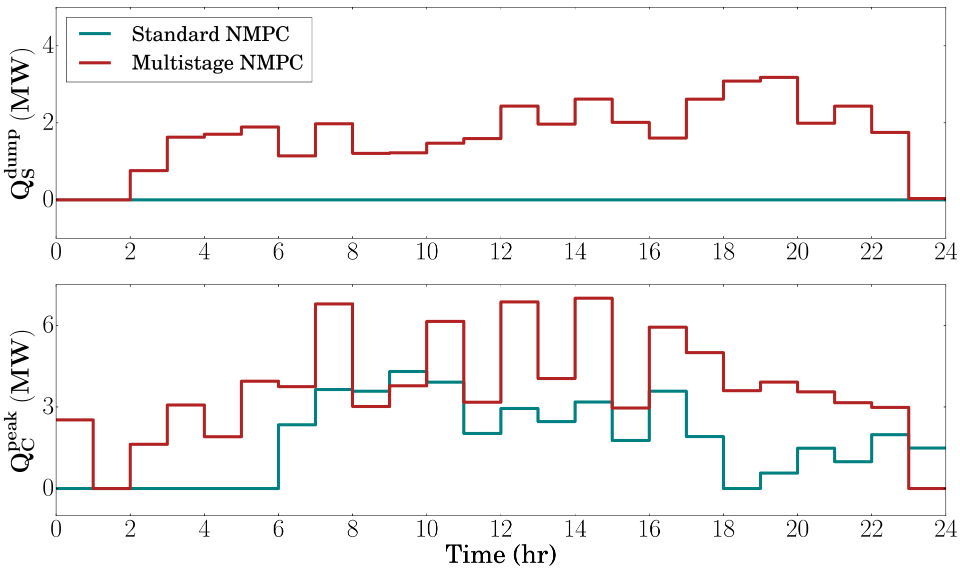


Figure 4.8: The heat dumping and peak heating profiles obtained from standard and multistage NMPC approaches for January 06, 2018.

T_C^{return} is always above its limit in order to satisfy consumer demand, and does not violate it.

Multistage NMPC, however, has a higher cost of peak heating than standard NMPC, as shown in Figure 4.8. While standard NMPC suggests a total peak heating of 42.15 MWh, multistage NMPC results in 88.73 MWh of peak heating across the day. This can be thought of as the "cost of robustness" required by multistage NMPC. Since T_C^{return} can only drop so much, the remaining heat demand has to be satisfied through additional peak heating. The supplier also has to dump a lot of heat compared to standard NMPC, since the temperature of the return stream has to be brought down below 85 $°C$. Another factor for this is the temperature difference between the supplier and the TES, which becomes too small to transfer heat.

The standard NMPC is oblivious to these differences in the heat supply and demand, and proposes control solutions that require lesser peak heating. The plant–model mismatch that arises from this uncertainty, however, causes the states to violate their operating constraints. Although the standard NMPC suggests lower peak heating requirements compared to multistage NMPC, this is counterproductive because it is not robust and ends up violating the operating constraint for a significant operating period. The economic penalty

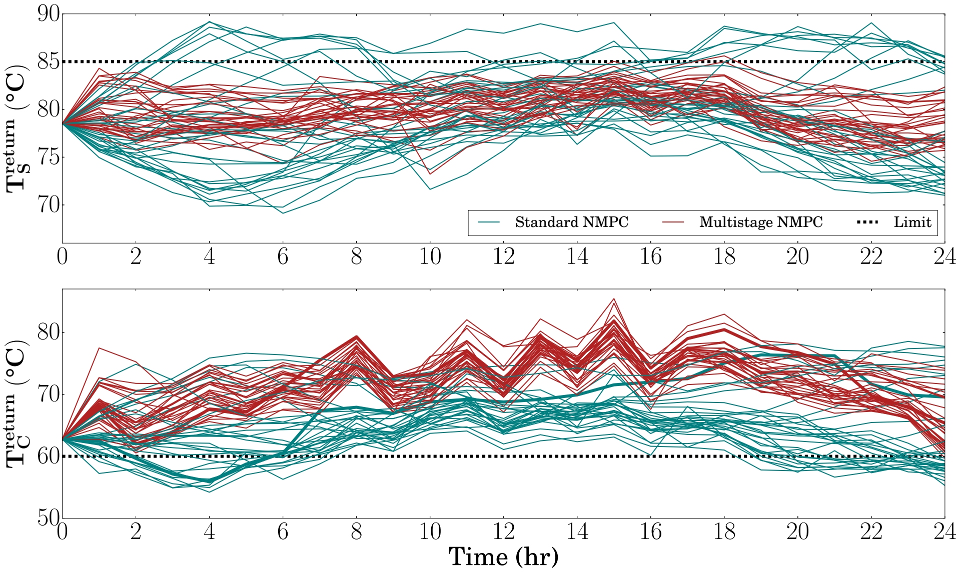


Figure 4.9: The supplier and consumer return temperature profiles for the whole month of January 2018; standard NMPC leads to frequent constraint violations, multistage NMPC keeps the temperatures within bounds, shown in dashed black lines.

of such suboptimal operation may be significantly higher than any savings achieved via lower peak heating.

We repeated the simulations for all 31 days of January 2018 using the corresponding heat supply and demand profiles, and found that multistage NMPC is consistently successful in keeping the system within the specified limits, whereas the standard NMPC frequently results in constraint violations. This can be seen in Figure 4.9, which shows the supplier and consumer return temperature profiles for all the days in January 2018. These simulations show the applicability of the method even when the heat supply at the start of the horizon is lower than the heat demand, as was the case for multiple days in January 2018.

The robustness offered by multistage NMPC comes with a higher peak heating cost. The average daily use of peak heating across all days in January with standard NMPC is found to be 17.27 MWh, whereas for multistage NMPC it is 73.11 MWh. We also noted the frequency of constraint violations in both cases. The supplier and consumer-side return temperature constraints are violated during a daily average of 3.2 hours and 3.6 hours respectively for standard NMPC. The corresponding values for multistage NMPC are 0.06 hours and 0.03 hours respectively. This shows that the cluster operator is better off employing the multistage NMPC strategy even though it has higher peak heating cost. This is because the long periods of unprofitable operation in standard NMPC may not be acceptable to the stakeholders, resulting in higher overall costs. Multistage NMPC thus helps to keep the overall costs down, while also making the system robust against uncertainties in heat supply and demand.

4.4 Conclusion

In this chapter, the application of the PCA-based multistage NMPC strategy is investigated for an industrial cluster system with a hot water TES unit. The case study presented a model with one supplier and one consumer of surplus heat, whereby the heat exchange happens through a TES tank. We looked at the challenge of effectively handling the uncertainty in heat supply and demand for this system, with the multistage NMPC scheme. A PCA-based scenario selection strategy is applied to quantify the uncertain supply and demand of heat, and the robustness of multistage NMPC is demonstrated by comparing it to a standard NMPC formulation. We use additional data analysis to detect outliers in the data and to predict trends in the heat profiles.

The results demonstrate that although the multistage NMPC is more conservative than the equivalent standard MPC formulation in terms of peak heating requirements (the “cost of robustness”), it is much better at keeping the system within specified bounds on the system states. We consider operating constraints on the return temperatures on the supplier and consumer plants, and find that the standard NMPC strategy leads to frequent, non-trivial, constraint violations. The multistage NMPC, on the other hand, is able to steer the system while respecting these operating constraints. We argue that large economic penalties for constraint violations justify the use of the multistage NMPC strategy over standard NMPC, as it is more effective in handling the uncertain heat supply and demand. In a nutshell, we demonstrate that implementing the proposed robust control strategy can result in an energy-efficient utilization of the TES unit for surplus heat exchange, not only providing cost-savings to the industrial cluster as a whole, but also benefiting its individual stakeholders.

Acknowledgment

We thank Mo Fjernvarme AS for providing industrial data on heat supply and demand used in this chapter.

Part II

Sensitivity-assisted Multistage Nonlinear Model Predictive Control

Chapter 5

Sensitivity-assisted Multistage Nonlinear Model Predictive Control: Robustness and Computational Efficiency

The major challenge in multistage NMPC is that the computational size of the problem grows rapidly with the scenario tree. In particular, the problem size grows exponentially with:

1. the number of uncertain parameters
2. the number of discrete realizations for each uncertain parameter
3. the length of the prediction horizon

The problem quickly becomes too expensive computationally, and this poses a challenge for real-time implementation of the NMPC. To expedite computations, various approximation strategies have been proposed. Most implementations of multistage NMPC apply robust horizon, where the branching of scenarios is stopped after a certain number of time steps in the prediction horizon. Because the problem size still grows exponentially with the length of robust horizon, it is typically restricted to one or two time steps in most applications. In the Part I of this thesis, a data-driven method of selecting scenarios was considered, where the uncertainty information is captured with fewer scenarios using multivariate data analysis, helping with the computational speed.

To improve computational efficiency in NMPC, several sensitivity-based approaches have been proposed (see [80, 81] for recent overviews). For instance, in [82], sensitivity calculations are used to dynamically update the NMPC horizon length in order to be computationally fast. The advanced-step NMPC proposed by [83] precomputes a set of solutions offline, and makes a quick sensitivity-based correction online to reduce computational effort. This concept was extended to a parallelizable advanced-step multistage NMPC algorithm proposed in [28].

In this chapter, we develop an efficient approximation to the multistage NMPC problem (2.4) by dividing the scenarios into two sets: a (small) set of so-called *critical* scenarios, and a (larger) set of *noncritical* scenarios. Under the assumption that inequality constraints in the system are strictly monotonic with respect to the uncertain parameters, the critical scenarios are considered to be those most likely to violate these inequality constraints. Thus, the critical scenarios are composed of the uncertainty realizations that lead to such constraint violations. We solve an optimization problem with a smaller scenario tree comprising of these critical scenarios. Further, to account for the remaining uncertainty, we include the costs of the noncritical scenarios in the objective function based on a sensitivity-based linear approximation. This approach, which we call the sensitivity-assisted multistage NMPC or samNMPC, directly addresses the issue of exponential growth rate of the problem size. In particular, the problem size is independent of the number of uncertain parameters, and also the number of discrete realizations for each uncertain parameter. Moreover, it scales only linearly with the length of the robust horizon, which allows us to efficiently approximate very large scenario tree representations for ideal multistage NMPC [1]. We also propose a parallelizable algorithm based on Schur complement decomposition that can be used to speed-up the sensitivity calculations on the linear algebra level.

The samNMPC algorithm is implemented on two illustrative case studies: a setpoint tracking of species concentration in a CSTR, and setpoint tracking of water levels in an interconnected four tank system. Its performance is compared with the standard NMPC and

[1] In Part II of this thesis, we refer to the multistage NMPC problem (2.4) as “ideal” multistage NMPC in order to differentiate it from our proposed sensitivity-assisted multistage NMPC.

ideal multistage NMPC formulations, in terms of robustness and computational efficiency. This chapter is largely based on the work done in [32], and extends some preliminary work done in [84, 85].

5.1 Nonlinear programming sensitivity

To see how the primal-dual solution s^* of the parametric NLP problem (2.5) changes as parameter p is perturbed from $p = p_0$ to $p = p$, we cite the following property:

Theorem 1. *(NLP Sensitivity) If $F(\cdot)$ and $c(\cdot)$ of the parametric NLP problem* (2.5) *are twice continuously differentiable in a neighborhood of the nominal primal and dual solution $s^*(p_0)$ and this solution satisfies the LICQ, SSOSC and SC, then,*

- *$s^*(p_0)$ is an isolated local minimizer of* (2.5) *at p_0 and the associated Lagrange multipliers are unique.*
- *For p in a neighborhood of p_0 there exists a unique, continuous and differentiable function $s^*(p)$ which is a local minimizer satisfying SSOSC and LICQ for* (2.5) *at p.*
- *There exists a positive Lipschitz constant L_S such that $|s^*(p)-s^*(p_0)| \leq L_S|p-p_0|$ where $|.|$ is the Euclidian norm.*
- *There exists a positive Lipschitz constant L_J such that the optimal cost values $F(p)$ and $F(p_0)$ satisfy $|F(p) - F(p_0)| \leq L_J|p - p_0|$.*

Proof. See [86, 87]. ■

Applying Theorem 1 and the Implicit Function Theorem to differentiate (2.13) leads to the following linear system for sensitivity of s:

$$\mathcal{M}(s(\mu; p_0))\Delta s = -\mathcal{N}(s(\mu; p_0); p) \tag{5.1}$$

where

$$\mathcal{M}(s(\mu; p_0)) = \begin{bmatrix} \nabla_{xx}\mathcal{L}(s(\mu; p_0)) & \nabla_x c(s(\mu; p_0)) & -I \\ \nabla_x c(s(\mu; p_0))^T & 0 & 0 \\ V(\mu; p_0) & 0 & X(\mu; p_0) \end{bmatrix}$$

is the KKT matrix, and

$$\mathcal{N}(s(\mu;p_0);p)^T = \begin{bmatrix} \nabla_x \mathcal{L}(s(\mu;p_0);p) \\ c(x(\mu;p_0);p) \\ 0 \end{bmatrix}$$

with

$$s(0;p) = s(\mu;p_0) + \Delta s + O(\|p - p_0\|^2) + O(\mu) \tag{5.2}$$

When LICQ, SSOSC, and SC are satisfied at $s(\mu;p_0)$, $\mathcal{M}(s(\mu;p_0))$ is nonsingular and the sensitivity steps Δs can be computed according to (5.1) using a cheap backsolve if the factorized form of $\mathcal{M}(s(\mu;p_0))$ is available. The first-order approximate solution at p can be calculated

$$\tilde{s}(p) = s^*(\mu;p_0) + \Delta s \tag{5.3}$$

Comparing this approximate solution at p with the true optimal solution of (2.5) at p, we have the following:

$$|s(p) - \tilde{s}(p)| = O(\|p - p_0\|^2) \tag{5.4}$$

Assuming the differentiability and Lipschitz continuity of $s(p)$, a positive Lipschitz constant L_s exists such that

$$|s(p) - \tilde{s}(p)| \leq L_s \|p - p_0\|^2 \tag{5.5}$$

The most expensive step in an NLP method like the interior-point algorithm is to formulate and factorize the KKT matrix. The main advantage of calculating the sensitivity step with (5.1) is that we avoid the expensive solution of the NLP (2.5) at p, which is crucial in the implementation of NMPC.

5.2 Sensitivity-assisted multistage NMPC scheme

For an efficient implementation of a multistage NMPC controller, the scenario tree should be suitably constructed to represent the uncertainty in the system. The limitation in ideal multistage NMPC is the exponential scaling of scenarios with the number of uncertain parameters, which creates a bottleneck for computational tractability. The main idea of sensitivity-assisted multistage NMPC (samNMPC) is to emulate the performance of ideal multistage NMPC with a reduced scenario tree.

Out of the {max, nominal, min} realizations of an uncertain parameter, we find the "worst-case" realization that is most likely to cause violation of an inequality constraint [27]. The combination of these worst-case realizations of each uncertain parameter forms a "critical" scenario for that inequality constraint. Going through all the inequality constraints, we form the set of critical scenarios that is used to build the scenario tree in samNMPC. Moreover, to account for the "noncritical" scenarios, we approximate them by using their associated NLP-sensitivity steps in the stage and terminal costs in the objective function. We first present the problem formulation of samNMPC, followed by the discussion of how to get the critical scenarios and the sensitivity steps for noncritical scenarios. The samNMPC formulation, at time step k, is:

$$\begin{aligned} \min_{\substack{\mathbf{z}_l^c, \mathbf{v}_l^c \\ c \in \widehat{\mathbb{C}} \cup \{0\}}} \quad & \sum_{c \in \widehat{\mathbb{C}} \cup \{0\}} \omega^c \Big(\phi(\mathbf{z}_N^c, \mathbf{d}_{N-1}^c) + \sum_{l=0}^{N-1} \varphi(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^c) \Big) + \\ & \sum_{c \in \bar{\mathbb{C}}} \omega^c \phi(\mathbf{z}_N^0 + \Delta \mathbf{z}_N^c, \mathbf{d}_{N-1}^c) + \\ & \sum_{c \in \bar{\mathbb{C}}} \omega^c \sum_{l=0}^{N-1} \varphi(\mathbf{z}_l^0 + \Delta \mathbf{z}_l^c, \mathbf{v}_l^0 + \Delta \mathbf{v}_l^c, \mathbf{d}_l^c) \end{aligned} \tag{5.6a}$$

$$\begin{aligned} \text{s.t.} \quad & \mathbf{z}_{l+1}^c = \mathbf{f}(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^c) \quad l = 0, \dots, N-1 && (5.6b) \\ & \mathbf{z}_0^c = \mathbf{x}_k && (5.6c) \\ & \mathbf{v}_l^c = \mathbf{v}_l^{c'} \quad \{(c, c') \mid \mathbf{z}_l^c = \mathbf{z}_l^{c'}\} && (5.6d) \\ & \mathbf{z}_l^c \in \mathbb{X}, \mathbf{v}_l^c \in \mathbb{U}, \mathbf{z}_N^c \in \mathbb{X}_f, \mathbf{d}_l^c \in \mathbb{D} && (5.6e) \\ & \mathbf{d}_{l-1}^c = \mathbf{d}_l^c \quad l = N_r, \dots N-1 && (5.6f) \\ & \forall c, c' \in \widehat{\mathbb{C}} \cup \{0\} \end{aligned}$$

where $\widehat{\mathbb{C}}$ and $\bar{\mathbb{C}}$ are the critical and noncritical scenario sets, respectively. The scenario $\{0\}$ represents the nominal scenario. To reiterate the notation, ω^c is the probability of each scenario, and $\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^c$ represent the vectors of state variables, control variables and uncertain parameters at stage l and scenario c. The sets $\mathbb{X}$, $\mathbb{X}_f$, $\mathbb{U}$ and $\mathbb{D}$ are the state variable domain, terminal region, control variable domain, and the bounded uncertainty set, respectively.

The decision variables in the problem are the state and control variables associated with only the nominal and critical scenarios, rendering a smaller problem formulation than the ideal multistage problem (2.4). The equality and inequality constraints (5.6b) – (5.6f) are imposed only for the nominal and critical scenarios. As shown in (5.6a), the noncritical scenarios are approximated with their NLP-sensitivity steps $\Delta\mathbf{z}^c$ and $\Delta\mathbf{v}^c$ in the objective function.

Problem (5.6) is a partially linearized version of ideal multistage NMPC problem (2.4), since the noncritical scenarios appear only as linear sensitivity steps in the objective function. The samNMPC formulation thus optimizes an expected performance over critical and noncritical scenarios. This is much like the ideal multistage NMPC formulation, where the expected cost is the weighted sum of the costs over all scenarios. The samNMPC thus provides an approximation of the ideal multistage NMPC with a reduced problem size.

5.2.1 Selecting critical scenarios

Critical scenarios are composed of worst-case uncertainty realizations that are most likely to violate inequality constraints (typically state bounds $\mathbf{z}_l \in \mathbb{X}$). Let $\mathbf{g}(\mathbf{z}_l, \mathbf{v}_l, \mathbf{d}_l) \leq 0$ represent the vector of inequality constraints in the NMPC formulation (5.6) at t_k, where $\mathbf{g} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \to \mathbb{R}^{n_g}$. If each individual inequality is indexed as $g_j(\cdot, \cdot, \cdot) \leq 0$, the critical scenarios can be found by solving the following optimization problem at t_k with a fixed control trajectory $\mathbf{v}_l$, for each inequality constraint with $l = 1, \ldots, N$ and $l' = 0, \ldots, l-1$:

$$\max_{\mathbf{d}_{l'}} \quad g_j(\mathbf{z}_l, \mathbf{v}_l, \mathbf{d}_l) \tag{5.7a}$$

$$\text{s.t.} \quad \mathbf{z}_{\bar{l}+1} = \mathbf{f}(\mathbf{z}_{\bar{l}}, \mathbf{v}_{\bar{l}}, \mathbf{d}_{\bar{l}}) \quad \bar{l} = l', \ldots, l-1 \tag{5.7b}$$

$$\mathbf{z}_0 = x_k \tag{5.7c}$$

Problem (5.7) is solved around a reference trajectory $(\mathbf{z}_l, \mathbf{v}_l)|_{ref}, \ l = 1, \ldots, N$. From (5.7) we observe that we have the implicit relationship that:

$$g_j(\mathbf{z}_l, \mathbf{v}_l, \mathbf{d}_l) = g_j(\mathbf{z}_l(\mathbf{d}_{l'}), \mathbf{v}_l, \mathbf{d}_l) = g_j(\mathbf{d}_{l'}) \leq 0, \ l' + 1 \leq l = 1, \ldots, N. \tag{5.8}$$

Moreover, we assume that the g_j model is strictly monotonic with respect to $\mathbf{d}_l$, i.e. the sensitivities of inequality constraints $\mathbf{g}$ with respect to the uncertain parameters $\mathbf{d}_l$ do not change sign across the trajectory of the parametric solution of the process model. The solution of (5.7) can then be found by linearizing process dynamics $\mathbf{f}$ and the inequality constraints $\mathbf{g}$ around the reference trajectory.

We now concatenate the uncertain parameters and states, and define $\mathbf{d}^T = [\mathbf{d}_0^T, \mathbf{d}_1^T, \ldots, \mathbf{d}_{N_r}^T]$ and $\mathbf{z}^T = [\mathbf{z}_0^T, \mathbf{z}_1^T, \ldots, \mathbf{z}_N^T]$. To find the sensitivities $\frac{d\mathbf{g}}{d\mathbf{d}} \in \mathbb{R}^{n_g \times n_d N_r}$ around the reference trajectory, we can then write:

$$\frac{d\mathbf{g}}{d\mathbf{d}}^T = \nabla_{\mathbf{z}}\mathbf{g}^T(\frac{d\mathbf{z}}{d\mathbf{d}})^T + \nabla_{\mathbf{d}}\mathbf{g}^T \tag{5.9}$$

Note that if the inequality constraints represent the state bounds $\mathbf{z}_l \in \mathbb{X}$, then $\nabla_{\mathbf{d}}\mathbf{g} = 0$, and $\nabla_{\mathbf{z}}\mathbf{g}$ is a sparse matrix with nonzero elements 1 or -1, corresponding to the upper or lower bounds respectively.

To compute the sensitivity $\frac{d\mathbf{z}}{d\mathbf{d}}$, let $\mathbf{c}(\mathbf{z}_l, \mathbf{d}_l) = 0$ represent the equality constraints in (5.7), with fixed $\mathbf{v}_l$. Differentiation by the Implicit Function Theorem yields:

$$\nabla_{\mathbf{z}}\mathbf{c}^T(d\mathbf{z}) + \nabla_{\mathbf{d}}\mathbf{c}^T(d\mathbf{d}) = 0$$

$$\frac{d\mathbf{z}}{d\mathbf{d}}^T = -(\nabla_{\mathbf{z}}\mathbf{c}^{-T})\nabla_{\mathbf{d}}\mathbf{c}^T. \tag{5.10}$$

Substituting in (5.9) leads to:

$$\frac{d\mathbf{g}}{d\mathbf{d}}^T = -\nabla_{\mathbf{z}}\mathbf{g}^T(\nabla_{\mathbf{z}}\mathbf{c}^{-T})\nabla_{\mathbf{d}}\mathbf{c}^T + \nabla_{\mathbf{d}}\mathbf{g}^T \tag{5.11}$$

Here we choose the solution of the standard NMPC problem solved at t_k as the reference trajectory. By doing so, the terms $\nabla_{\mathbf{z}}\mathbf{c}$ and $\nabla_{\mathbf{d}}\mathbf{c}$ are obtained from the Jacobian matrix at the optimal solution obtained by solving the standard NMPC problem. This allows for the efficient computation of critical scenarios even for longer robust horizons, since $\nabla_{\mathbf{d}}\mathbf{c}$ can be readily obtained by parameterizing the standard NMPC problem in the uncertain parameters $(\mathbf{d}_l)_{l=1,\cdots,N_r}$.

We further assume that the worst-case realization of the uncertain parameter $(d_m)_{m=1,\cdots,n_d}$ lies on either of its extremes d_m^{max} or d_m^{min} [88]. Combining this assumption with strict monotonicity, the analytical solution of (5.7) can be stated for $l \in N_r$ and $m = 1, \cdots, n_d$:

$$\begin{aligned} d_{l,m}^{wc} &= \arg\max_{\mathbf{d}_l \in \mathbb{D}} \quad \frac{d\mathbf{g}}{d\mathbf{d}}^T \mathbf{d} \\ &= \begin{cases} d_{l,m}^{min}, & \text{if } \dfrac{d(g_j)}{d(d_{l,m})}|_{(\mathbf{z}_l, \mathbf{v}_l)|_{ref}} \leq 0 \\ d_{l,m}^{max}, & \text{otherwise} \end{cases} \end{aligned} \tag{5.12}$$

The result that worst-case realizations of the uncertainty are always on the extremes follows directly from the strict monotonicity assumption. Note that this is a strong assumption - it certainly holds when the dynamics and input disturbances are linear in the plant, but may be violated for processes that are nonlinear. Finding the worst-case realizations in the general nonlinear case requires the solution of an NP-hard optimization problem [89]. As such, it may not be possible to compute these worst-case problems in the background except for smaller systems.

Nevertheless, the monotonicity assumption leads to a straightforward computation of the critical scenarios $\widehat{\mathbb{C}}$ in the general framework of our samNMPC algorithm, and we only need to apply (5.12) for the inequality constraints that are active or are close to the state trajectory. Indeed, if an element $|\dfrac{d(g_j)}{d(d_{l,m})}| \leq \epsilon$, i.e. the constraint g_j is insensitive to the uncertain parameter $d_{l,m}$, then the corresponding critical scenarios can be ignored. Thus, the number of critical scenarios is bounded by the number of active inequality constraints, which in practice is much smaller than the fully branched scenario tree. The number of critical scenarios does *not* scale with the number of uncertain parameters, since only the worst-case realizations of the uncertainties are relevant.

At each iteration of the samNMPC algorithm, a standard NMPC problem is solved to get the reference trajectory $(\mathbf{z}_l, \mathbf{v}_l)|_{ref}$, and the critical scenarios are updated dynamically. An illustration of a reduced scenario tree is shown in Figure 5.1, where there are only 3 critical scenarios (in addition to the nominal scenario), as opposed to 9 scenarios in a fully branched scenario tree. The constraints corresponding to the critical scenarios are included in the final NLP of samNMPC, as shown in (5.6).

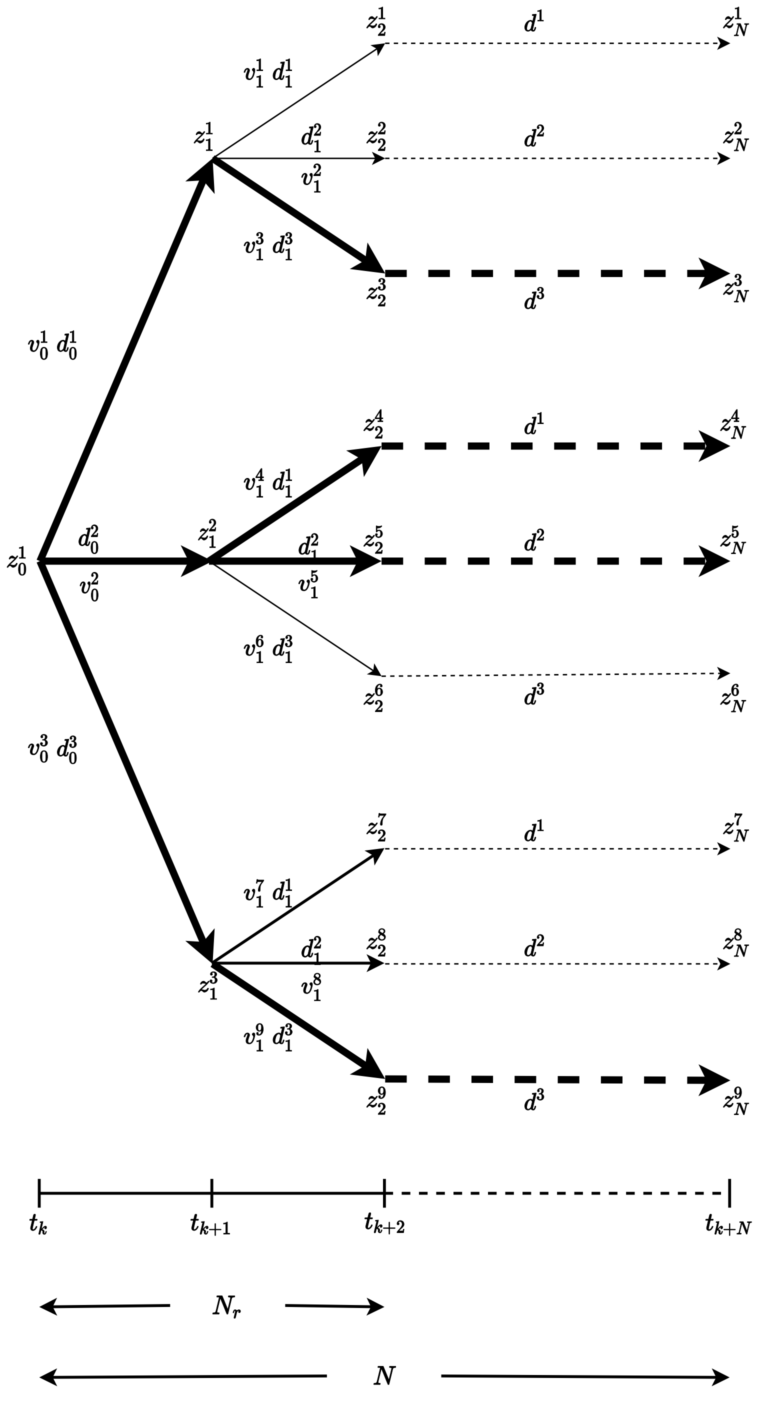


Figure 5.1: Example of a reduced scenario tree with $n_d = 1$ and $N_r = 2$. The nominal and critical scenarios are shown with the thick lines.

5.2.2 Computing sensitivity steps for noncritical scenarios

To get the sensitivity steps for the noncritical scenarios, we start with the scenario decoupled KKT matrix $\mathcal{M}$ of the ideal multistage NMPC formulation (2.4) (following the notation of Chapter 5.1):

$$\begin{bmatrix} \mathbf{W}_0 & & & & \mathbf{A}_0 & & & & \widetilde{\mathbf{N}}_0 \\ & \mathbf{W}_1 & & & & \mathbf{A}_1 & & & \widetilde{\mathbf{N}}_1 \\ & & \ddots & & & & \ddots & & \vdots \\ & & & \mathbf{W}_{n_c} & & & & \mathbf{A}_{n_c} & \widetilde{\mathbf{N}}_{n_c} \\ \mathbf{A}_0^T & & & & & & & & \\ & \mathbf{A}_1^T & & & & & & & \\ & & \ddots & & & & & & \\ & & & \mathbf{A}_{n_c}^T & & & & & \\ \widetilde{\mathbf{N}}_0^T & \widetilde{\mathbf{N}}_1^T & \dots & \widetilde{\mathbf{N}}_{n_c}^T & & & & & \end{bmatrix} \tag{5.13}$$

where $|\mathbb{C}|$ is the number of scenarios and $n_c = |\mathbb{C}| - 1$ is the last index of the scenarios. $\mathbf{W}_c = \nabla_{x^c x^c}\mathcal{L}(x, \lambda, \nu) + X_c^{-1}V_c$ is the augmented Hessian for scenario c, with $X_c = diag(x^c)$ and $V_c = diag(\nu^c)$. The Jacobian matrix of each scenario is decomposed into two parts, with respect to the non-NAC constraints $\mathbf{A_c}$, and with respect to the NAC constraints $\widetilde{\mathbf{N}}_c$:

$$\mathbf{A}_c = \nabla_{x^c} c_i(x) \quad \forall i \in \bar{\mathcal{M}} \tag{5.14a}$$

$$\widetilde{\mathbf{N}}_c = \nabla_{x^c} c_j(x) \quad \forall j \in \widehat{\mathcal{M}} \tag{5.14b}$$

where $\bar{\mathcal{M}}$ and $\widehat{\mathcal{M}}$ are the index sets for the constraints representing the non-NAC equalities and the NACs, respectively.

Rearranging the KKT matrix (5.13):

$$\begin{bmatrix} \mathbf{W}_0 & \mathbf{A}_0 & & & & & & & \widetilde{\mathbf{N}}_0 \\ \mathbf{A}_0^T & & & & & & & & 0 \\ & & \mathbf{W}_1 & \mathbf{A}_1 & & & & & \widetilde{\mathbf{N}}_1 \\ & & \mathbf{A}_1^T & & & & & & 0 \\ & & & & \ddots & & & & \vdots \\ & & & & & \ddots & & & \vdots \\ & & & & & & \mathbf{W}_{n_c} & \mathbf{A}_{n_c} & \widetilde{\mathbf{N}}_{n_c} \\ & & & & & & \mathbf{A}_{n_c}^T & & 0 \\ \widetilde{\mathbf{N}}_0^T & 0 & \widetilde{\mathbf{N}}_1^T & 0 & \dots & \dots & \widetilde{\mathbf{N}}_{n_c}^T & 0 & \end{bmatrix} \tag{5.15}$$

the linear system (5.1) can be rewritten in the following block-bordered-diagonal (BBD) form:

$$\begin{bmatrix} \mathbf{K}_0 & & & & \mathbf{N}_0 \\ & \mathbf{K}_1 & & & \mathbf{N}_1 \\ & & \ddots & & \vdots \\ & & & \mathbf{K}_{n_c} & \mathbf{N}_{n_c} \\ \mathbf{N}_0^T & \mathbf{N}_1^T & \dots & \mathbf{N}_{n_c}^T & \end{bmatrix} \begin{bmatrix} \Delta s_0 \\ \Delta s_1 \\ \vdots \\ \Delta s_{n_c} \\ \gamma \end{bmatrix} = - \begin{bmatrix} t_0 \\ t_1 \\ \vdots \\ t_{n_c} \\ 0 \end{bmatrix} \tag{5.16}$$

where $\mathbf{K}_c = \begin{bmatrix} \mathbf{W}_c & \mathbf{A}_c \\ \mathbf{A}_c^T & 0 \end{bmatrix}$, $\Delta s_c = \begin{bmatrix} \Delta x^c \\ \Delta \lambda^c \end{bmatrix}$, $t_c = \begin{bmatrix} \nabla_{x^c} \mathcal{L}(x^c, d^c) \\ c(x^c, d^c) \end{bmatrix}$. The primal variables associated with the scenario c in the multistage formulation (2.4) are $x^c = [\mathbf{z}_0^c, \mathbf{v}_0^c, \mathbf{z}_1^c, \mathbf{v}_1^c, \cdots, \mathbf{z}_{N-1}^c, \mathbf{v}_{N-1}^c, \mathbf{z}_N^c]^T$ and λ^c are the dual variables associated with scenario c.

In (5.16), $\mathbf{N}_c$ represents the NAC constraint that contains scenario c, where $\mathbf{N}_c = [\widetilde{\mathbf{N}}_c,\ 0]^T \in \mathbb{R}^{n+m_c} \times \mathbb{R}^{m_{NAC} \times n_u}$ and n and m_c are the number of primal variables and constraints, respectively, in each scenario, and m_{NAC} and n_u are the number of NAC constraints associated with each control variable, and the number of control variables, respectively. $\mathbf{N}_c$ and $\widetilde{\mathbf{N}}_c$ can be generated for robust scenarios of any length, and are sparse with nonzero elements of 1's and -1's that correspond only to control variables for the respective NAC. Additionally, γ in (5.16) is the multiplier associated with NAC (2.4d) with the dimensions $\gamma \in \mathbb{R}^{m_{NAC} \times n_u}$.

Solving the linear system (5.16) for the full multistage problem is computationally expensive for a large number of scenarios. We seek an approximate solution of (5.16) that is fast to compute. To be specific, we solve the standard NMPC formulation (2.2) for the nominal scenario to get the KKT matrix $\mathbf{K}_0$ of the nominal scenario, we apply the approximation $\mathbf{K}_c = \mathbf{K}_0, \; \forall c \in \mathbb{C}$ in (5.16).

Defining $\Delta \mathbf{d}$ as the maximum difference in the uncertainty vector between any two realizations of the uncertainty

$$|\Delta \mathbf{d}| = \max_{j,j' \in \mathbb{M}} |\mathbf{d}^j - \mathbf{d}^{j'}| \tag{5.17}$$

we expect an $O(|\Delta \mathbf{d}|)$ error to result from this approximation, as the sensitivities evaluated at the standard NMPC solution differ from the full multistage NMPC solution by $O(|\Delta \mathbf{d}|)$, i.e. we have $\mathbf{K}_c = \mathbf{K}_0 + O(|\Delta \mathbf{d}|)$.

Thus, assembling the full KKT matrix on the LHS of (5.16) only requires the solution of the standard NMPC problem, as the NAC matrices $\mathbf{N}_c \quad \forall c \in \mathbb{C}$ always stay the same. The solution of the approximated linear system (5.16) gives the sensitivity steps for all scenarios $\Delta \mathbf{s}_c, \; \forall c \in \mathbb{C}$.

Note that since we solve for the sensitivities of the standard NMPC problem to evaluate the critical scenarios anyway, we have access to the most updated $\mathbf{K}_0$ to be used in (5.16) at every time step. For especially large problems, an alternative may be to fix $\mathbf{K}_0$ corresponding to the solution of a steady state problem, and compute all the sensitivities offline to be reused at every time step. However, this ignores the model disturbance dynamics and may add another layer of inaccuracy.

The Schur complement decomposition can also be used to solve the linear system (5.16). For the KKT matrix in (5.16), the Schur complement is formed as:

$$S = \sum_{c \in \mathbb{C}} (\mathbf{N}_c^T \mathbf{K}_c^{-1} \mathbf{N}_c) \tag{5.18}$$

The NAC multipliers γ can then be obtained by solving:

$$S\,\gamma = -\sum_{c \in \mathbb{C}} (\mathbf{N}_c^T \mathbf{K}_c^{-1} t_c) \tag{5.19}$$

Finally the sensitivity steps for all scenarios $\Delta \mathbf{s}_c$ can be computed by solving:

$$\mathbf{K}_c \Delta \mathbf{s}_c = -(t_c + \mathbf{N}_c \gamma), \; \forall c \in \mathbb{C} \tag{5.20}$$

Note that for longer robust horizons, both the size and number of $\mathbf{N}_c$ matrices increases exponentially, and the summation of the matrix products in the RHS of (5.18) and (5.19) across all scenarios, significantly adds to the computational cost. To fully realize the advantage of using Schur complements, these matrix multiplications can be parallelized. The sensitivity steps in (5.20) can also be computed in parallel. Moreover, the approximation $\mathbf{K}_c = \mathbf{K}_0 \quad \forall c \in \mathbb{C}$ helps in speeding up the computations, since we can store the factorization of $\mathbf{K}_0$.

After identifying the critical and noncritical scenario sets as explained in Section 5.2.1, the sensitivity steps of the noncritical scenarios $c \in \bar{\mathbb{C}}$ are included in the objective function of the samNMPC formulation (5.6). Note that the NACs of the noncritical scenarios are still satisfied in the approximated linear system (5.16), and thus they are implicitly taken into account in the samNMPC formulation (5.6).

5.2.3 Overall samNMPC algorithm and implementation

The overall samNMPC strategy is summarized in Algorithm 1. Our procedure models the same category of feedback information and its impact on the controller as with ideal multistage NMPC. This is done by including the predicted state and control trajectories for the critical scenarios, i.e., worst-case uncertainty realizations, as well as sensitivity approximations for predicted state and control trajectories for all of the remaining scenarios. As a result, all of the scenarios considered in ideal multistage NMPC are also considered with samNMPC. Also note that the samNMPC solution differs from the ideal multistage solution by $O(|\Delta \mathbf{d}|)$ (see Chapter 6).

The samNMPC algorithm is implemented using the software `JuMP` (version 0.19.2) [62], which provides a convenient framework for mathematical optimization for NLPs. The `JuMP` tool works within the framework of the `Julia` (version 1.0.3) programming language [63]. The NLP solver used within this framework is `IPOPT` [53], which uses interior-point algorithms to solve NLPs. The `MA57` linear solver from the Harwell Subroutine Library [79] is used within `IPOPT`. All computational experiments are carried out with an Intel i7-7600 Quad Core CPU at 2.8 GHz and 16GB RAM.

`JuMP` allows for directly querying derivative information at the optimal solution, and thus can be effectively used to construct $\mathbf{K}_0$. The NAC-associated sparse matrices $\mathbf{N}_c, \ \forall c \in \mathbb{C}$ are generated automatically for the given number of control variables and the length of

Algorithm 1: Sensitivity-assisted multistage NMPC

```
Given: {max, nominal, min} of all uncertain parameters.
for $k = 1, 2, \dots$ do
    Get the current state of the plant $\mathbf{x}_k$.
    Solve the standard NMPC problem (2.2) for the nominal uncertainty $\mathbf{d}_k^0$, and
      get the KKT matrix $\mathbf{K}_0$ at the optimal solution.
    For critical scenarios: Extract $\nabla_{\mathbf{z}}\mathbf{c}$ and $\nabla_{\mathbf{d}}\mathbf{c}$ from $\mathbf{K}_0$, and solve (5.11) to form
      the critical scenario set $\widehat{\mathbb{C}}$.
    For noncritical scenarios: Solve the linear system (5.16) with the approximation
      $\mathbf{K}_c = \mathbf{K}_0 \quad \forall c \in \mathbb{C}$, and get the sensitivity steps for the noncritical scenarios
      $c \in \bar{\mathbb{C}}$.
    Solve samNMPC formulation (5.6), where constraints are imposed for critical
      scenarios and the noncritical scenarios are approximated with their sensitivity
      steps in the objective function.
    Set $\mathbf{u}_k = \mathbf{v}_0^c,\ c \in \widehat{\mathbb{C}} \cup \{0\}$ and inject into the plant.
end
```

the robust horizon. The resulting approximate linear system (5.16) is also solved using `MA57`.

It is worth mentioning in the context of this algorithm that the strict monotonicity assumption, which forms the basis for computing the critical scenarios, may not always hold for real applications. In such cases, one could potentially apply the control input $\mathbf{u}_k$ obtained by solving the samNMPC problem (5.6) to simulate all the noncritical scenarios in $\bar{\mathbb{C}}$, in order to check for constraint violations. If a noncritical scenario causes a constraint violation, it can be removed from $\bar{\mathbb{C}}$ and added to $\widehat{\mathbb{C}}$, and samNMPC can be solved again with the new critical scenario set. Although such a rigorous treatment of critical scenarios is not considered in this work, the results of the case studies in the next section indicate that there were no constraint violations for noncritical scenarios.

5.3 Case studies

5.3.1 CSTR example

We first consider the nonlinear benchmark CSTR problem [90], where the dynamics are described by the following equations:

$$\frac{dc_A}{dt} = F(c_{A0} - c_A) - k_1 c_A - k_3 c_A^2 \tag{5.21a}$$

$$\frac{dc_B}{dt} = -F c_B + k_1 c_A - k_2 c_B \tag{5.21b}$$

$$\frac{dT_R}{dt} = F(T_{in} - T_R) + \frac{k_W A}{\rho c_p V_R}(T_K - T_R) \quad - \frac{k_1 c_A \Delta H_{AB} + k_2 c_B \Delta H_{BC} + k_3 c_A^2 \Delta H_{AD}}{\rho c_p} \tag{5.21c}$$

$$\frac{dT_K}{dt} = \frac{1}{m_K c_{pK}}(\dot{Q}_K + k_W A(T_R - T_K)) \tag{5.21d}$$

where the reaction rate k_i follows the Arrhenius law, $k_i = k_{0,i} e^{\frac{-E_{A,i}}{R(T_R+273.15)}}$. The state vector $\mathbf{x} = [c_A, c_B, T_R, T_K]^T$, which are the concentration of A, concentration of B, the reactor temperature and jacket temperature, respectively. The control input vector $\mathbf{u} = [F, \dot{Q_K}]^T$, which are the inlet flow per reactor volume $F = V_{in}/V_R$ and the cooling rate $\dot{Q_K}$. Tables C.1 and C.2 show the model parameters and bounds, respectively.

The control objective is the setpoint tracking for desired product concentration c_B. An operation period of $0.2h$ is considered, with the setpoint for $t < 0.1h$ being $c_B^{ref} = 0.5\ mol/L$, and for $t \geq 0.1h$ it is $c_B^{ref} = 0.7\ mol/L$. The uncertain parameters in the system are $E_{A,3}$ and c_{A0}. The cost to be minimized is:

$$\varphi_l = (c_{Bl} - c_{Bl}^{ref})^2 + r_1 \Delta F_l^2 + r_2 \Delta \dot{Q_K} \tag{5.22}$$

where $\Delta F_l = F_l - F_{l-1}$ and $\Delta \dot{Q}_{Kl} = \dot{Q}_{Kl} - \dot{Q}_{Kl-1}$ are the difference between consecutive control actions for the flow rate and cooling rate, respectively, and $r_1 = 10^{-7}$ and $r_2 = 10^{-11}$ are the penalty parameters. The prediction horizon is chosen to be $N = 40$ with each step time being $0.005h$. The MPC algorithm is thus implemented in 40 runs for the entire operation period of $0.2h$.

The uncertainty range for $E_{A,3}$ and c_{A0} both is $\pm 10\%$ of the nominal value. In this case study, we consider that the uncertainty $\mathbf{d}_k$ is time-variant. To be precise, we consider that the true realizations of $E_{A,3}$ and c_{A0} take random values from their corresponding $\{\mathsf{max, nominal, min}\}$ values. We consider the robust horizons $N_r = 1, 2, 3$ to compare the standard, ideal multistage and samNMPC algorithms. The results for $N_r = 1, 2, 3$ are shown in Figures 5.2, 5.3 and 5.4, respectively.

For each of $E_{A,3}$ and c_{A0}, there are 3 possible realizations of the uncertainty - the number of branches per node in the scenario tree is thus 9. This corresponds to 9 scenarios in the full tree for $N_r = 1$. The tracking performance of the three NMPC schemes under uncertainty for $N_r = 1$ is shown in Figure 5.2. The ideal multistage and samNMPC schemes show similar performance in tracking the setpoint of c_B (shown in dashed blue line). Moreover, both schemes show robust constraint satisfaction, respecting the upper bound of T_R (shown in dashed black line). On the other hand, standard NMPC shows poor tracking performance due to the significant plant-model mismatch arising from the time-varying uncertainty. Standard NMPC is also not robust, violating the T_R constraint for a significant period of operation.

The tracking performance improves for both ideal multistage and samNMPC schemes for $N_r = 2$ (Figure 5.3), which corresponds to 81 scenarios in the full tree. This is because uncertainty in future time steps is also explicitly modeled in the scenario tree. Again, both ideal multistage and samNMPC are able to satisfy the state constraints. Increasing the robust horizon to $N_r = 3$ results in a full tree consisting of 729 scenarios, making ideal multistage NMPC computationally intractable due to large problem size. However, samNMPC is able to solve a much smaller problem as it identifies the critical scenarios, and is able to provide robustness in its tracking performance, as shown in Figure 5.4.

Note that here the uncertainty in $E_{A,3}$ and c_{A0} is $\pm 10\%$ each. For a smaller uncertainty range in these parameters, a much better tracking performance is achieved for all the NMPC schemes. However, due to the smaller plant-model mismatch, standard NMPC does not show any bound violations and ends up being robust.

Table 5.1 shows the CPU computations measured as the wall clock solution times for each of the three NMPC problems. These are averaged over 5 random sequences of uncertainty realizations. It can be clearly seen that samNMPC requires less computational effort than ideal multistage NMPC. Table 5.1 also gives a measure of the reduction in problem size with samNMPC, showing that the average number of critical scenarios scales only linearly with the robust horizon. The computationally intensive elements of the samNMPC algorithm are solving the linear system (5.16) and solving the approximate problem (5.6),

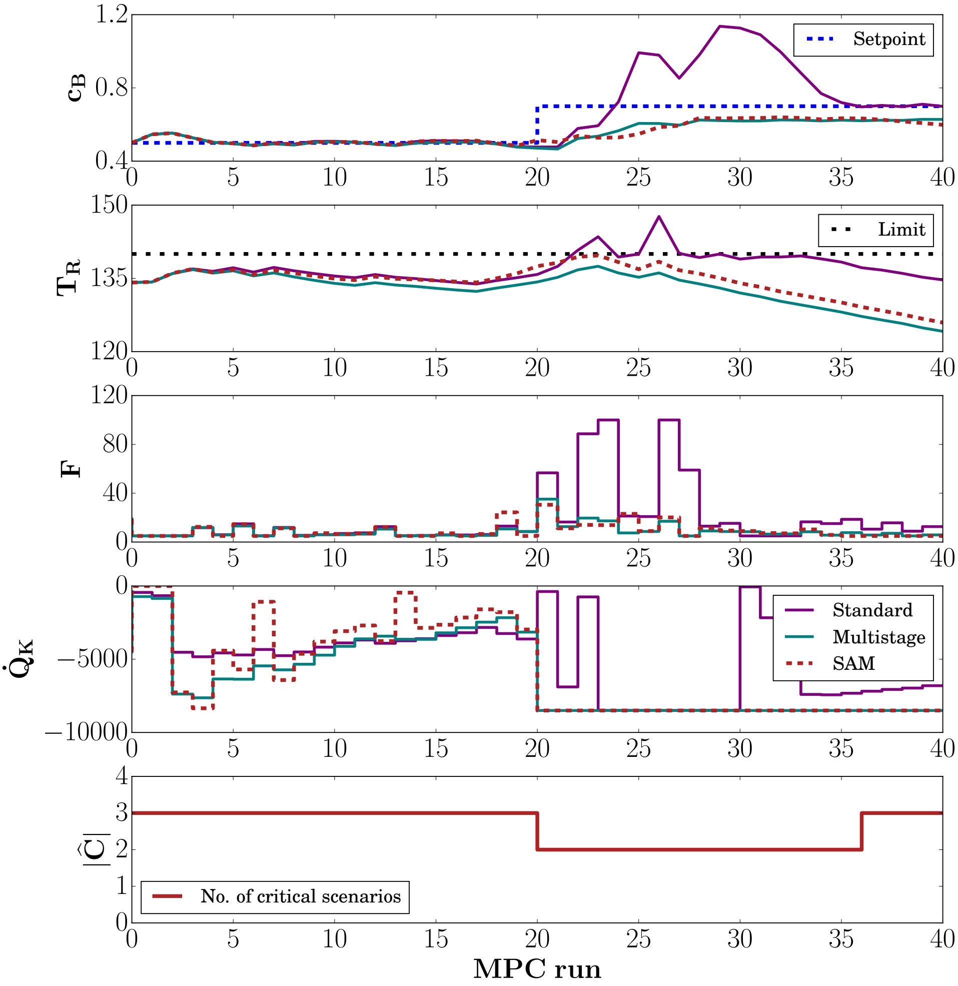


Figure 5.2: Setpoint tracking (c_B) and constraint satisfaction (T_R) for standard, ideal multistage and samNMPC with $N_r = 1$ (9 scenarios).

Table 5.1: CSTR - Average computational performance in CPUs. For samNMPC, the times for solving linear system (5.16) and the NLP (5.6) are reported separately.

N_r	$\lvert\mathbb{C}\rvert$	Standard	Multistage	SAM - solving (5.16) / (5.6)	Avg. $\lvert\widehat{\mathbb{C}}\rvert$
1	9	0.453 s	3.910 s	0.113 s / 1.589 s	2.54
2	81	0.453 s	46.327 s	1.819 s / 4.223 s	5.96
3	729	0.453 s	–	18.071 s / 9.367 s	9.24

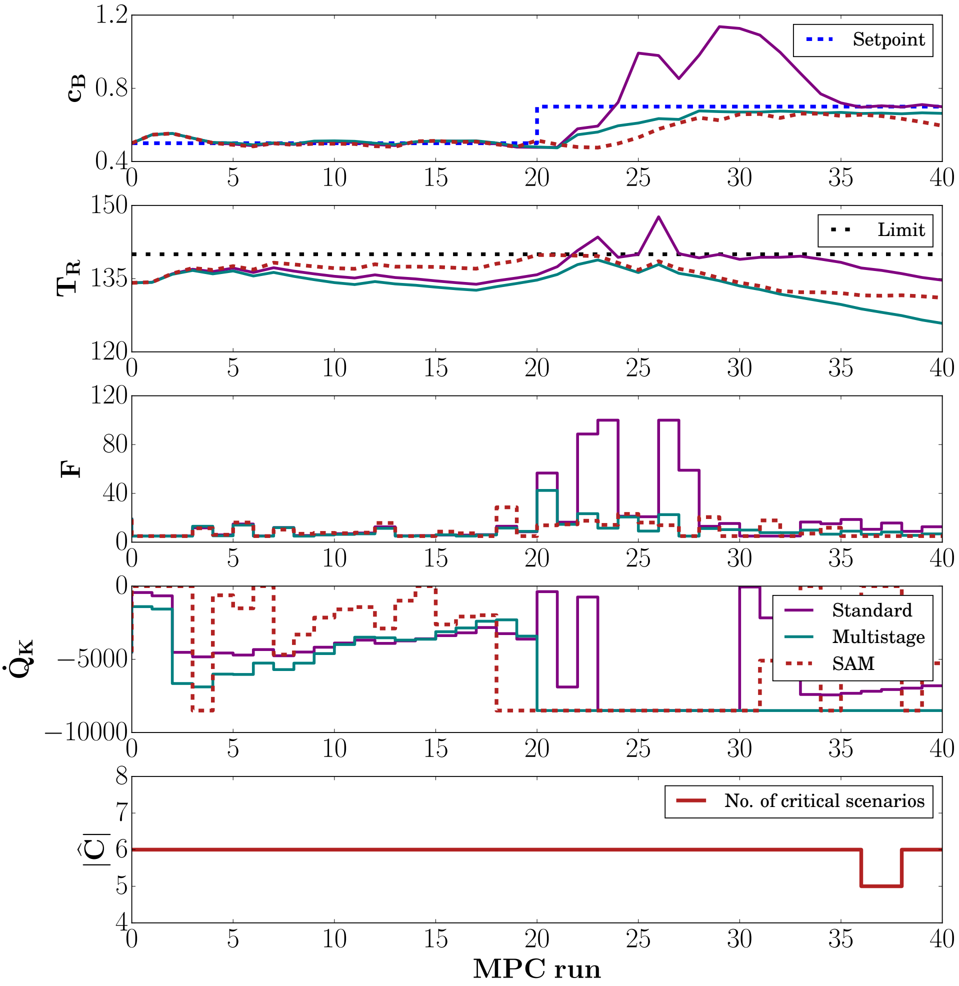


Figure 5.3: Setpoint tracking (c_B) and constraint satisfaction (T_R) for standard, ideal multistage and samNMPC with $N_r = 2$ (81 scenarios).

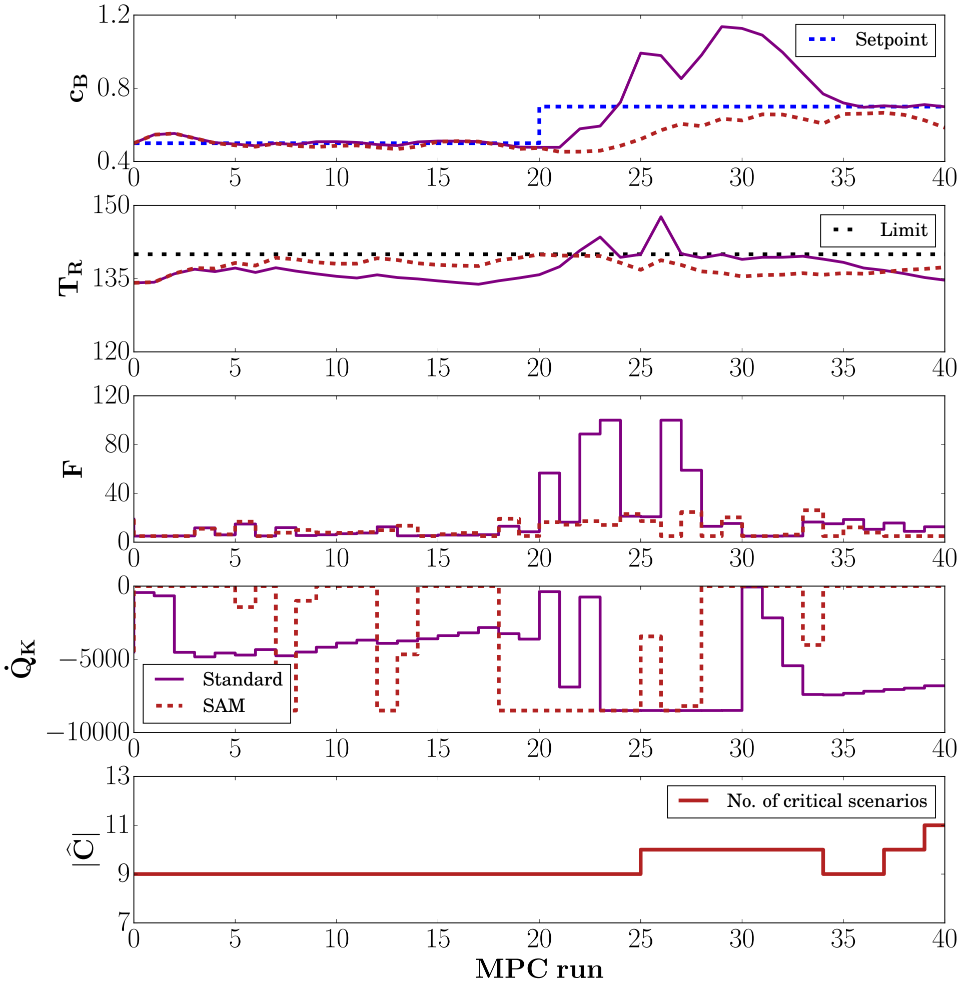


Figure 5.4: Setpoint tracking (c_B) and constraint satisfaction (T_R) for standard and samNMPC with $N_r = 3$ (729 scenarios).

Table 5.2: CSTR - Average computational performance in CPUs for solving (5.16) as large linear system and with Schur complement

N_r	$\|\mathbb{C}\|$	Solving (5.16) directly	Using Schur complement decomposition (best parallel estimate)
1	9	0.113 s	0.019 (0.002) s
2	81	1.819 s	1.326 (0.016) s
3	729	18.071 s	530.995 (0.728) s

which are shown separately in Table 5.1. The results show that the combined computational footprint of samNMPC is much smaller than that of the full-tree ideal multistage NMPC problem, which involves solving a full NLP.

The size of the KKT matrix in (5.16) grows exponentially with increasing N_r, and solving the linear system takes up the bulk of the time at longer robust horizons. Using the Schur complement strategy avoids solving the large linear system by decomposing it into smaller linear systems, which can then be solved in parallel to achieve better computational performance. However, as noted before, computing the Schur complement itself requires summation over the products of many matrices (5.18), and this can become very expensive without parallelization. This is especially the case for $N_r = 3$, where the Schur complement is computed by summing over 729 scenarios. On the other hand, this summation is trivially parallelizable and does not lead to communication latencies (see [91] for extensive analysis on this). As such, we note the wall times without parallelization for the Schur complement strategy, and divide them by the number of scenarios to report the "best estimate" CPUs with parallelization. Table 5.2 shows the comparison between solving (5.16) as a large linear system and with the use of Schur complement strategy, averaged over 5 random sequences of uncertainty realizations. Because the Schur complement computations for $N_r = 3$ are expensive without parallelization, we only report the wall time for solving one NMPC step in one random uncertainty sequence with the Schur complement (instead of the average wall time), for $N_r = 3$.

As the length of the robust horizon increases, using the Schur complement approach with parallelization can be up to two orders-of-magnitude faster than solving the large linear system (5.16).

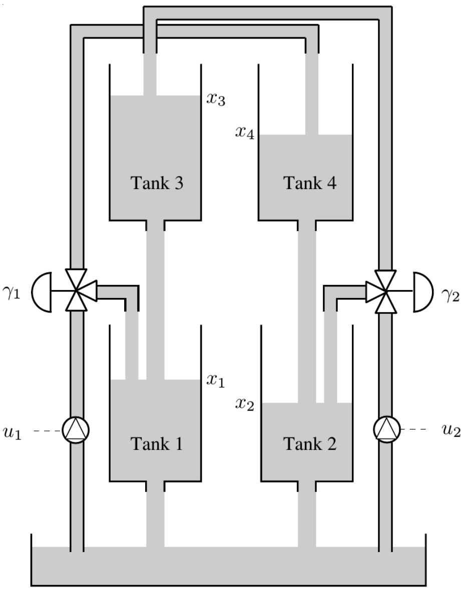


Figure 5.5: Quadtank schematic [92]

5.3.2 Quadtank example

In our second case study, we consider the Quadtank problem [92], with a configuration of four tanks as shown in Figure 5.5.

The water levels in the four tanks are governed by the following dynamics:

$$\frac{dx_1}{dt} = -\frac{a_1}{A_1}\sqrt{2gx_1} + \frac{a_3}{A_1}\sqrt{2gx_3} + \frac{\gamma_1}{A_1}u_1 \tag{5.23a}$$

$$\frac{dx_2}{dt} = -\frac{a_2}{A_2}\sqrt{2gx_2} + \frac{a_4}{A_2}\sqrt{2gx_4} + \frac{\gamma_2}{A_2}u_2 \tag{5.23b}$$

$$\frac{dx_3}{dt} = -\frac{a_3}{A_3}\sqrt{2gx_3} + \frac{1-\gamma_2}{A_3}u_2 \tag{5.23c}$$

$$\frac{dx_4}{dt} = -\frac{a_4}{A_4}\sqrt{2gx_4} + \frac{1-\gamma_1}{A_4}u_1 \tag{5.23d}$$

where $\mathbf{x}_i$ denotes the level in the tank i, and u_i represents the flow rate of pump i. The state vector is $\mathbf{x} = [x_1, x_2, x_3, x_4]$ and the control inputs are $\mathbf{u} = [u_1, u_2]$. A_i and a_i are the cross sectional areas of the tank i and its outlet port, respectively. The parameters γ_1 and γ_2 are the valve parameters, and are considered to the uncertainties in the system. The values of the various model parameters and bounds are shown in Tables C.3 and C.4.

Table 5.3: Quadtank - Average computational performance in CPUs. For samNMPC, the times for solving linear system (5.16) and the NLP (5.6) are reported separately.

N_r	$\vert\mathbb{C}\vert$	Standard	Multistage	SAM - solving (5.16) / (5.6)	Avg. $\vert\widehat{\mathbb{C}}\vert$
1	9	0.096 s	1.113 s	0.058 s / 0.293 s	3
2	81	0.096 s	12.906 s	0.703 s / 0.6 s	4.99
3	729	0.096 s	–	7.735 s / 1.850 s	6.97

The control goal is to track the water levels in the lower tanks (tank 1 and 2) at their setpoints $x_{1s} = 14cm$ and $x_{2s} = 14cm$. Further, we introduce predefined pulse changes in the state values at certain steps k to reinitialize the controller tracking, as shown in Table C.5. The prediction horizon is $N = 25$, with each time step being $10s$. The MPC simulation is run for 150 time steps.

The objective function is formulated as:

$$\varphi_l = (x_{1l} - x_{1s})^2 + (x_{2l} - x_{2s})^2 + r(\Delta u_1^2 + \Delta u_2^2) \tag{5.24}$$

where $\Delta u_{1l} = u_1 - u_{1l-1}$ and $\Delta u_2 = u_{2l} - u_{2l-1}$ are the difference between consecutive control actions for the pump flow rates, and the penalty parameter $r = 0.01$.

The uncertainty range for γ_1 and γ_2 is ± 0.15 of their nominal values of 0.4. As with the previous case study, we consider that the true realizations of γ_1 and γ_2 take random values from their corresponding $\{\mathsf{max}, \mathsf{nominal}, \mathsf{min}\}$ values. The resulting plots for an increasing robust horizon are shown in Figures 5.6, 5.7 and 5.8, respectively.

As is evident from the figures, all three NMPC schemes have very similar performance in tracking the setpoints of x_1 and x_2 (shown in dashed blue lines). Overall, they are able to maintain the water levels even in face of the pulse disturbances in the water levels. In terms of robustness, ideal multistage and samNMPC do not breach the specified water level limits for x_3 and x_4 (shown in dashed black lines), whereas there are frequent violations on part of standard NMPC. Moreover, the trajectories of ideal multistage and samNMPC overlap almost exactly for $N_r = 1$ (Figure 5.6), and are reasonably close for $N_r = 2$ (Figure 5.7). As with the CSTR case, the full tree problem becomes too large to solve for $N_r = 3$, but the samNMPC algorithm is able to handle it efficiently, as shown in Figure 5.8.

A comparison of the computational times, averaged over 5 random sequences of uncertainty realizations, is shown in Tables 5.3 and 5.4. Again, samNMPC significantly outper-

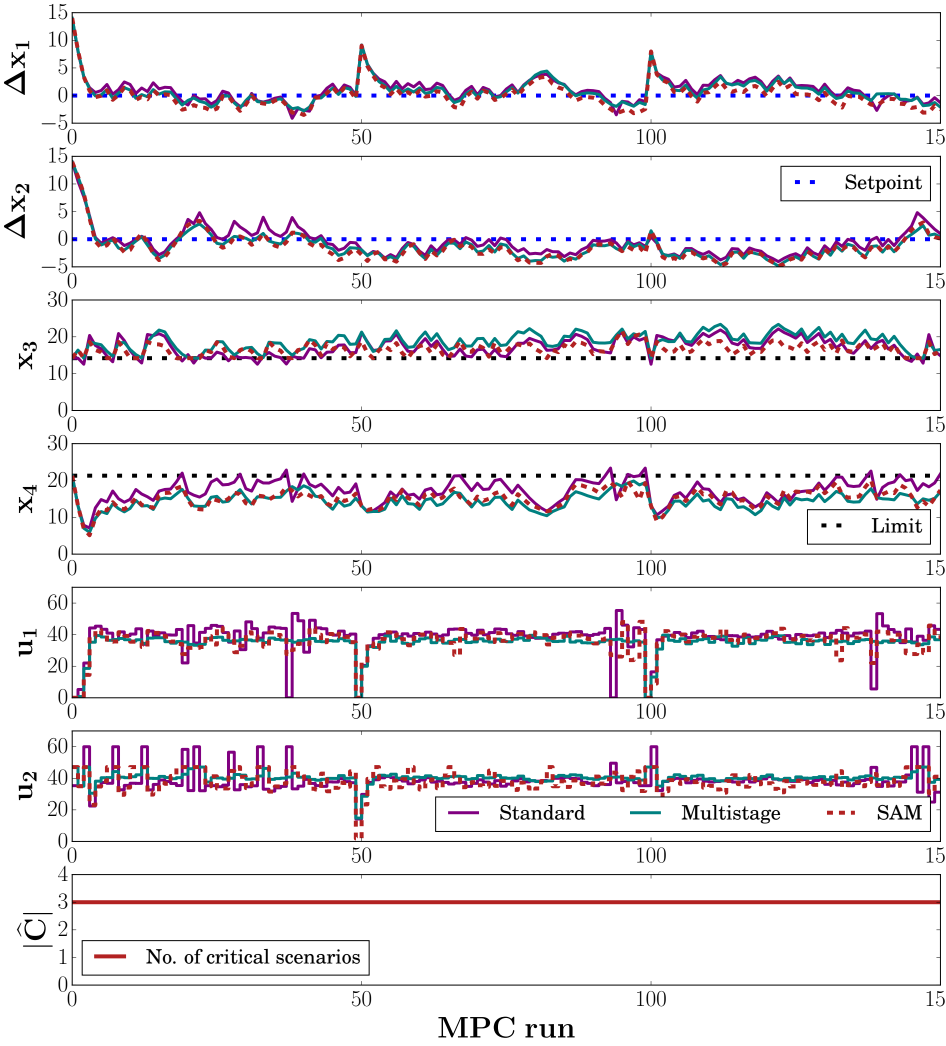


Figure 5.6: Setpoint tracking (x_1, x_2) and constraint satisfaction (x_3, x_4) for standard, ideal multi-stage and samNMPC with $N_r = 1$ (9 scenarios).

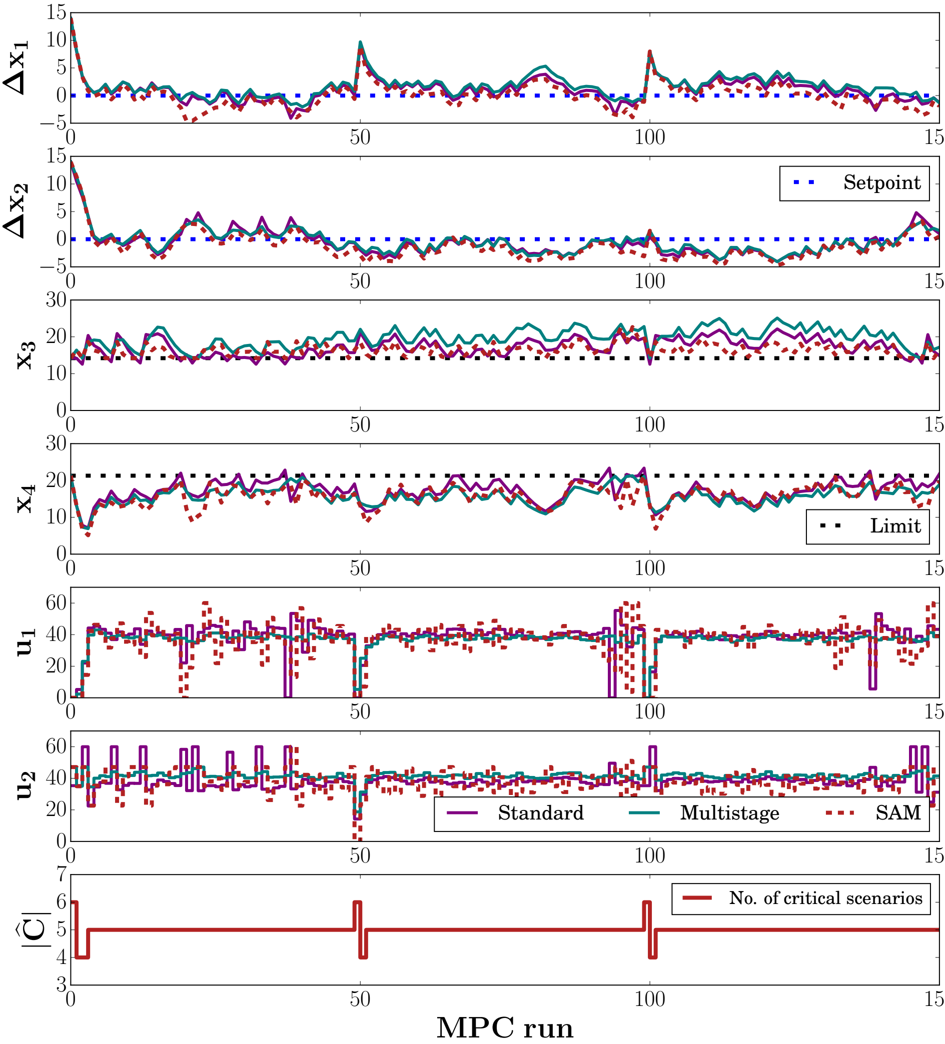


Figure 5.7: Setpoint tracking (x_1, x_2) and constraint satisfaction (x_3, x_4) for standard, ideal multistage and samNMPC with $N_r = 2$ (81 scenarios).

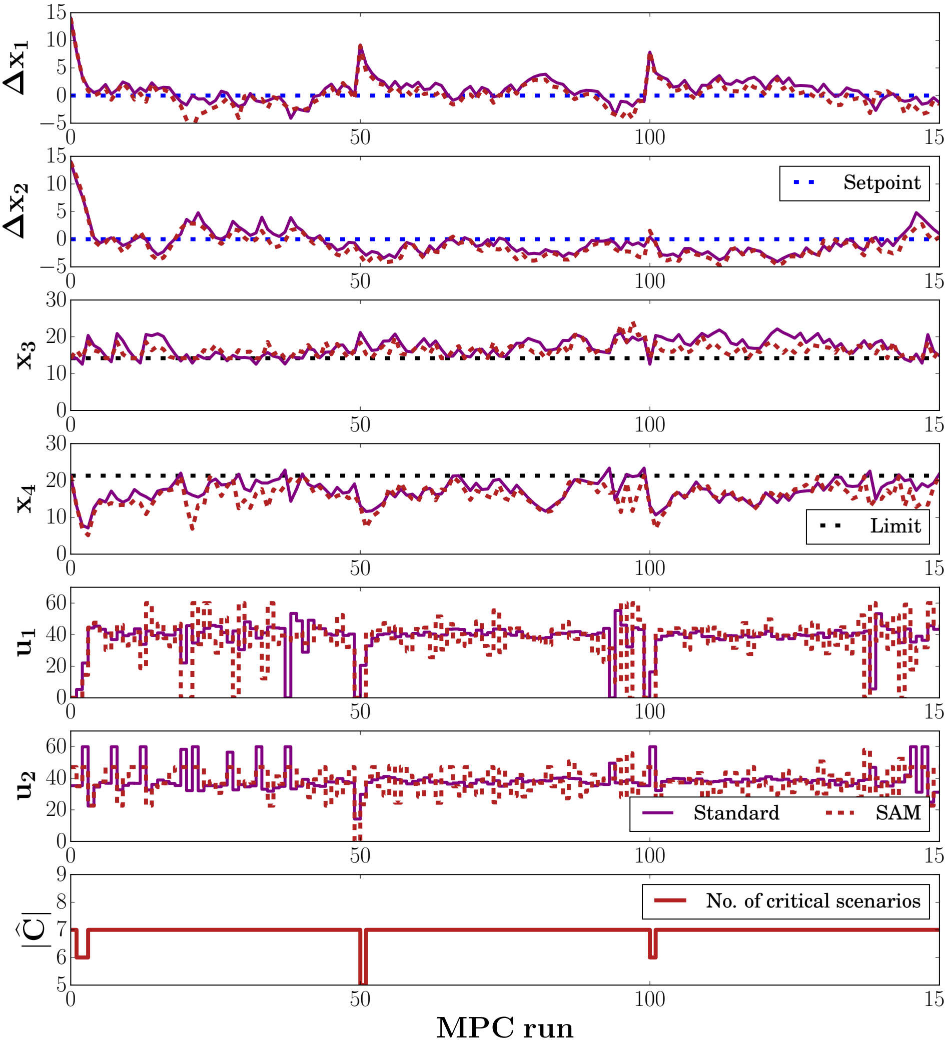


Figure 5.8: Setpoint tracking (x_1, x_2) and constraint satisfaction (x_3, x_4) for standard, ideal multi-stage and samNMPC with $N_r = 3$ (729 scenarios).

Table 5.4: Quadtank - Average computational performance in CPUs for solving (5.16) as large linear system and with Schur complement.

N_r	$\lvert\mathbb{C}\rvert$	Solving (5.16) directly	Using Schur complement decomposition (best parallel estimate)
1	9	0.058 s	0.0087 (0.0009) s
2	81	0.703 s	0.825 (0.01) s
3	729	7.735 s	389.26 (0.534) s

forms ideal multistage NMPC in terms of speed, without sacrificing robustness or setpoint tracking under uncertainty. Table 5.4 shows that, consistent with the previous case study, using the Schur complement method is expensive without parallelization. On the other hand, parallelization can significantly improve the computational speed of samNMPC.

5.4 Conclusion

This chapter presents an approximate, sensitivity-assisted multistage NMPC strategy to reduce the computational load of robust NMPC. The samNMPC approach optimizes over a set of critical scenarios that are most likely to cause constraint violations, and approximates the noncritical scenarios with their corresponding sensitivities in the objective function. These sensitivities are obtained by solving an approximate KKT system of the multistage NMPC problem. The strategy ensures robust constraint satisfaction under uncertainty, while performing similarly to the ideal multistage NMPC algorithm. In contrast with the exponential growth in ideal multistage NMPC, the problem size in samNMPC grows only linearly with the length of the robust horizon; and it is independent of the number of uncertain parameters and the number of discrete realizations of each uncertain parameter. It thus leads to much lower computational costs than its full-tree counterpart.

We apply the samNMPC approach to the CSTR and Quadtank case studies, and compare its performance with respect to standard NMPC and ideal multistage NMPC. The examples demonstrate that samNMPC achieves robustness and tracking performance similar to the ideal multistage NMPC, at a fraction of the computational footprint. It performs particularly well computationally for longer robust horizons where ideal multistage NMPC becomes too expensive. The use of parallel Schur complement decomposition can further speed up the solution time.

Chapter 6

Sensitivity-assisted Multistage Nonlinear Model Predictive Control: Stability Properties

In this chapter, we discuss the recursive feasibility and stability properties of both ideal multistage and samNMPC, using a novel analysis based on soft-constrained formulations. Previous studies have analyzed the recursive feasibility and stability properties of ideal multistage NMPC [93, 94]. In this chapter, we show using the soft-constrained formulations that recursive feasibility and stability of ideal multistage NMPC can be guaranteed even with the robust horizon assumption ($N_r \leq N$). These concepts are are also extended for samNMPC.

We first introduce fundamental concepts and assumptions of stability properties, then discuss the notion of relaxed inequality (soft) constraints, and finally proceed to discuss the recursive feasibility and input-to-state practical stability (ISpS) for both the ideal multistage and samNMPC. This chapter is based on the work done in [32].

6.1 Preliminaries

We start with some fundamental concepts needed for the stability analysis of ideal multi-stage and samNMPC.

Definition 7. *A continuous function* $\alpha(\cdot) : \mathbb{R} \to \mathbb{R}$ *is a* $\mathcal{K}$ *function if* $\alpha(0) = 0, \alpha(s) > 0, \forall s > 0$ *and it is strictly increasing. A continuous function* $\alpha(\cdot) : \mathbb{R} \to \mathbb{R}$ *is a* $\mathcal{K}_\infty$ *function if it is a* $\mathcal{K}$ *function and* $\lim_{s\to\infty} \alpha(s) = \infty$. *A continuous function* $\beta(\cdot,\cdot) : \mathbb{R} \times \mathbb{Z} \to \mathbb{R}$ *is a* $\mathcal{KL}$ *function if* $\beta(s,k)$ *is a* $\mathcal{K}$ *function in* s *for any* $k > 0$ *and for each* $s > 0, \beta(s,\cdot)$ *is decreasing and* $\beta(s,k) \to 0$ *as* $k \to \infty$.

Definition 8. *(Lyapunov function) A function* $V(\cdot)$ *is called a Lyapunov function for system* (2.1) *if there exist an invariant set* $\mathbb{X}$, *a feedback control law* $\mathbf{h}(\mathbf{x})$ *and* $\mathcal{K}_\infty$ *functions* α_1, α_2 *and* α_3 *such that,* $\forall \mathbf{x} \in \mathbb{X}$

$$V(\mathbf{x}) \geq \alpha_1(|\mathbf{x}|) \tag{6.1a}$$

$$V(\mathbf{x}) \leq \alpha_2(|\mathbf{x}|) \tag{6.1b}$$

$$\Delta V(\mathbf{x}) = V(\mathbf{f}(\mathbf{x},\mathbf{h}(\mathbf{x}))) - V(\mathbf{x}) \leq -\alpha_3(|\mathbf{x}|) \tag{6.1c}$$

Definition 9. *(Control invariant set, [1]) A set* $\mathbb{A}$ *is a control (or positive) invariant set for system* $\mathbf{x}^+ = \mathbf{f}(\mathbf{x},\mathbf{u})$ *if for all* $\mathbf{x} \in \mathbb{A}$, *there exists* $\mathbf{u} \in \mathbb{U}$ *such that* $\mathbf{f}(\mathbf{x},\mathbf{u}) \in \mathbb{A}$.

Definition 10. *(RPI) A set* $\mathcal{X}$ *is a robustly positive invariant (RPI) set for system* $\mathbf{x}^+ = \mathbf{f}(\mathbf{x},\mathbf{u},\mathbf{d})$ *if* $\mathbf{x}^+ \in \mathcal{X}$ *holds for* $\forall \mathbf{x} \in \mathcal{X}$, *and* $\forall \mathbf{d} \in \mathbb{D}$.

Definition 11. *(ISpS) The system* $\mathbf{x}^+ = \mathbf{f}(\mathbf{x},\mathbf{u},\mathbf{d})$ *is input-to-state practically stable (ISpS) in* $\mathcal{X}$ *if there exists a* $\mathcal{KL}$ *function* β, *a* $\mathcal{K}$ *function* γ *and* $c_d \geq 0$ *such that for all* $\mathbf{d} \in \mathbb{D}$,

$$|\mathbf{x}_k| \leq \gamma(|\Delta\mathbf{d}|) = \beta(|\mathbf{x}_0|,k) + \gamma(|\hat{\mathbf{d}}_k - \mathbf{d}^0|) + c_d, \ \forall k \geq 0, \ \forall \mathbf{x}_0 \in \mathcal{X} \tag{6.2}$$

where $\hat{\mathbf{d}}_k$ *is the realized value of* $\mathbf{d}$ *at time k, and* $\mathbf{d}^0$ *is the nominal disturbance.*

Definition 12. *(ISpS-Lyapunov function, [95]) A function* $V(\cdot)$ *is called an ISpS-Lyapunov function for system* $\mathbf{x}^+ = \mathbf{f}(\mathbf{x},\mathbf{u},\mathbf{d})$ *if there exist an RPI set* $\mathcal{X}$, $\mathcal{K}_\infty$ *functions* $\alpha_1, \alpha_2, \alpha_3$,

$\mathcal{K}$ function σ, and $c_0, c_1 \geq 0$ such that, $\forall \mathbf{x} \in \mathcal{X}$ and $\forall \mathbf{d} \in \mathbb{D}$,

$$\begin{aligned} V(\mathbf{x}) &\geq \alpha_1(|\mathbf{x}|) && \text{(6.3a)} \\ V(\mathbf{x}) &\leq \alpha_2(|\mathbf{x}|) + c_0 && \text{(6.3b)} \\ \Delta V(\mathbf{x}, \mathbf{d}) &= V(\mathbf{f}(\mathbf{x}, \mathbf{h}(\mathbf{x}), \mathbf{d})) - V(\mathbf{x}) \\ &\leq -\alpha_3(|\mathbf{x}|) + \sigma(|\hat{\mathbf{d}}_k - \mathbf{d}^0|) + c_1 && \text{(6.3c)} \end{aligned}$$

where $\mathbf{h}(\mathbf{x})$ is the feedback control law, $\hat{\mathbf{d}}_k$ is the realized value of $\mathbf{d}$ at time k, and $\mathbf{d}^0$ is the nominal disturbance.

6.2 Implication of soft constraints

To facilitate the stability discussion, we consider the notion of soft constraints for the NMPC problems proposed so far. First, we note that MFCQ and GSSOSC (Definitions 3 and 5) are the weakest conditions under which a perturbed solution of NLP (2.5) is locally unique [52], and Lipschitz continuity of $x^*(p)$ with respect to p can be guaranteed.

To put this in context, we now relax the inequalities in (2.5) with penalty variables r to form:

$$\begin{aligned} \min_{x} \quad & F(x;p) + M(r^T e) \\ \text{s.t.} \quad & c(x;p) = 0 \\ & x + r \geq 0 \\ & r \geq 0 \end{aligned} \tag{6.4}$$

where M is a penalty weight and $e^T = [1, \ldots, 1]$. Note that the inequality constraints are always feasible, and since the equality constraints in (6.4) are the discrete dynamic equations and NACs in ideal multistage NMPC (2.4) and samNMPC (5.6), these constraints have (forward) solutions for all inputs and initial conditions in their domains, and the gradients of the equality constraints therefore contain a nonsingular basis matrix and are linearly independent [96]. Now, with the relaxation of the inequality constraints, it is straightforward to show that the MFCQ always holds [96] at the solution of (6.4).

Finally, by adding the quadratic term

$$\|x - x^*\|_W^2 \tag{6.5}$$

to the objective in (6.4), where W is a positive definite weighting matrix, solutions of the KKT conditions of (6.4) are unchanged and (x^*, λ, ν) remains a KKT point. Moreover, by defining matrix Z as a basis of the nullspace of active constraint gradients (in Definition 4) and choosing W, with sufficiently large eigenvalues for $Z^T W Z$, then SSOSC and GSSOSC can always be satisfied at x^* (Note that if the IPOPT solver is used, it is not necessary to know *a priori* the optimal solution x^* for adding the quadratic term; in fact, an internal quadratic term is automatically added in IPOPT as part of its regularization strategy, such that the algorithm converges to the optimal solution [53]). As a result, the Lipschitz continuity property holds for all $x^*(p)$ and $F(x^*(p))$ with respect to all input parameters p, but feasibility of problems (2.4) and (5.6) holds only if $r = 0$ at the solution. We note that for all the case study solutions in Chapter 5, a penalty weight M could always be chosen large enough so that $r = 0$ held, and there were no infeasible solutions to the ideal multistage NMPC problem (2.4), or the samNMPC problem (5.6).

With these properties in mind, the standard NMPC problem (2.2) can be reformulated by relaxing the bounds on the state variables. Here we redefine $\mathbf{x} \in \mathbb{X} = \{\mathbf{x} \mid \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U\}$ and $\mathbf{x} \in \mathbb{X}_f = \{\mathbf{x} \mid \mathbf{x}_f^L \leq \mathbf{x} \leq \mathbf{x}_f^U\}$, and state the relaxed problem as:

$$
\begin{aligned}
J_{nom}(\mathbf{x}_k) \equiv \min_{\mathbf{z}_l, \mathbf{v}_l} \quad & \phi(\mathbf{z}_N, \mathbf{d}_{N-1}^0) + \sum_{l=0}^{N-1} \varphi(\mathbf{z}_l, \mathbf{v}_l, \mathbf{d}_l^0) + \\
& M_\phi \mathbf{e}^T \mathbf{r}_N + \sum_{l=0}^{N-1} M_\varphi \mathbf{e}^T \mathbf{r}_l && (6.6a) \\
\text{s.t.} \quad & \mathbf{z}_{l+1} = \mathbf{f}(\mathbf{z}_l, \mathbf{v}_l, \mathbf{d}_l^0) \qquad l = 0, \ldots, N-1 && (6.6b) \\
& \mathbf{z}_0 = \mathbf{x}_k && (6.6c) \\
& \mathbf{x}^L - \mathbf{r}_l \leq \mathbf{z}_l \leq \mathbf{x}^U + \mathbf{r}_l;\ \mathbf{r}_l \geq 0 && (6.6d) \\
& \mathbf{x}_f^L - \mathbf{r}_N \leq \mathbf{z}_N \leq \mathbf{x}_f^U + \mathbf{r}_N;\ \mathbf{r}_N \geq 0 && (6.6e) \\
& \mathbf{v}_l \in \mathbb{U} && (6.6f)
\end{aligned}
$$

where M_ϕ and M_φ are large weights for the penalty terms, and $\mathbf{e}^T = [1, \ldots, 1]$. The bound constraints on the states are softened with penalty variables $\mathbf{r}_l \in \mathbb{R}^{n_x}$ as shown in (6.6d) and (6.6e), and the corresponding penalty terms are added to the objective function J_{nom} in (6.6a).

For the ideal multistage NMPC problem, let $\mathbf{d}_{l=0,\ldots,N-1}^{c=1,\ldots,|\mathbb{C}|}$ denote the concatenated vector of all uncertainties in the scenario tree (across all scenarios and across all time steps in the prediction horizon). For brevity, we use the notation $\mathbf{d}^c$ to denote $\mathbf{d}_{l=0,\ldots,N-1}^{c=1,\ldots,|\mathbb{C}|}$ in the rest of this chapter, and make our subsequent problem formulations parametric in this $\mathbf{d}^c$. Relaxing the bounds on the state variables in the ideal multistage NMPC problem (2.4) leads to the following reformulation:

$$
\begin{aligned}
J_{ms}^N(\mathbf{x}_k,\mathbf{d}^c) \equiv \min_{\mathbf{z}_l^c,\mathbf{v}_l^c} \quad & \sum_{c\in\mathbb{C}} \omega^c \Big(\phi(\mathbf{z}_N^c,\mathbf{d}_{N-1}^c) + \sum_{l=0}^{N-1} \varphi(\mathbf{z}_l^c,\mathbf{v}_l^c,\mathbf{d}_l^c)\Big) + \\
& \sum_{c\in\mathbb{C}} \omega^c \Big(M_\phi \mathbf{e}^T \mathbf{r}_N^c + \sum_{l=0}^{N-1} M_\varphi \mathbf{e}^T \mathbf{r}_l^c\Big) && \text{(6.7a)}\\
\text{s.t.} \quad & \mathbf{z}_{l+1}^c = \mathbf{f}(\mathbf{z}_l^c,\mathbf{v}_l^c,\mathbf{d}_l^c) \quad l = 0,\ldots,N-1 && \text{(6.7b)}\\
& \mathbf{z}_0^c = \mathbf{x}_k && \text{(6.7c)}\\
& \mathbf{x}^L - \mathbf{r}_l^c \le \mathbf{z}_l^c \le \mathbf{x}^U + \mathbf{r}_l^c;\ \mathbf{r}_l^c \ge 0 && \text{(6.7d)}\\
& \mathbf{x}_f^L - \mathbf{r}_N^c \le \mathbf{z}_N^c \le \mathbf{x}_f^U + \mathbf{r}_N^c;\ \mathbf{r}_N^c \ge 0 && \text{(6.7e)}\\
& \mathbf{v}_l^c = \mathbf{v}_l^{c'} \quad \{(c,c') \mid \mathbf{z}_l^c = \mathbf{z}_l^{c'}\} && \text{(6.7f)}\\
& \mathbf{v}_l^c \in \mathbb{U},\ \mathbf{d}_l^c \in \mathbb{D} && \text{(6.7g)}\\
& \mathbf{d}_{l-1}^c = \mathbf{d}_l^c \quad l = N_r,\ldots,N-1 && \text{(6.7h)}\\
& \forall c, c' \in \mathbb{C}
\end{aligned}
$$

where we note that (6.7a) and its optimal objective function J_{ms}^N are parametric in $(\mathbf{x}_k,\mathbf{d}^c)$.

Similarly relaxing the bounds in the samNMPC problem (5.6) leads to the following formulation:

$$
\begin{aligned}
J_{sam}^{N}(\mathbf{x}_k, \mathbf{d}^c) \equiv \min_{\substack{\mathbf{z}_l^c, \mathbf{v}_l^c \\ c \in \widehat{\mathbb{C}} \cup \{0\}}} \quad & \sum_{c \in \widehat{\mathbb{C}} \cup \{0\}} \omega^c \Big(\phi(\mathbf{z}_N^c, \mathbf{d}_{N-1}^c) + \sum_{l=0}^{N-1} \varphi(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^c) \Big) + \\
& \sum_{c \in \widehat{\mathbb{C}} \cup \{0\}} \omega^c \Big(M_\phi \mathbf{e}^T \mathbf{r}_N^c + \sum_{l=0}^{N-1} M_\varphi \mathbf{e}^T \mathbf{r}_l^c \Big) + \\
& \sum_{c \in \bar{\mathbb{C}}} \omega^c \phi(\mathbf{z}_N^0 + \Delta \mathbf{z}_N^c, \mathbf{d}_{N-1}^c) + \\
& \sum_{c \in \bar{\mathbb{C}}} \omega^c \sum_{l=0}^{N-1} \varphi(\mathbf{z}_l^0 + \Delta \mathbf{z}_l^c, \mathbf{v}_l^0 + \Delta \mathbf{v}_l^c, \mathbf{d}_l^c) && \text{(6.8a)} \\
\text{s.t.} \quad & \mathbf{z}_{l+1}^c = \mathbf{f}(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^c) \quad l = 0, \ldots, N-1 && \text{(6.8b)} \\
& \mathbf{z}_0^c = \mathbf{x}_k && \text{(6.8c)} \\
& \mathbf{x}^L - \mathbf{r}_l^c \le \mathbf{z}_l^c \le \mathbf{x}^U + \mathbf{r}_l^c; \ \mathbf{r}_l^c \ge 0 && \text{(6.8d)} \\
& \mathbf{x}_f^L - \mathbf{r}_N^c \le \mathbf{z}_N^c \le \mathbf{x}_f^U + \mathbf{r}_N^c; \ \mathbf{r}_N^c \ge 0 && \text{(6.8e)} \\
& \mathbf{v}_l^c = \mathbf{v}_l^{c'} \quad \{(c, c') \mid \mathbf{z}_l^c = \mathbf{z}_l^{c'}\} && \text{(6.8f)} \\
& \mathbf{v}_l^c \in \mathbb{U}, \mathbf{d}_l^c \in \mathbb{D} && \text{(6.8g)} \\
& \mathbf{d}_{l-1}^c = \mathbf{d}_l^c \quad l = N_r, \ldots N-1 && \text{(6.8h)} \\
& \forall c, c' \in \widehat{\mathbb{C}} \cup \{0\}
\end{aligned}
$$

where (6.7a) and its optimal objective function J_{sam}^{N} are also parametric in $(\mathbf{x}_k, \mathbf{d}^c)$.

Further, we consider the *nominal case* of the ideal multistage NMPC problem (6.7) where we set $\mathbf{d}_l^c = \mathbf{d}_l^0 \;\; \forall c \in \mathbb{C}$:

$$\begin{aligned}
J_{ms}^N(\mathbf{x}_k, \mathbf{d}^0) \equiv \min_{\mathbf{z}_l^c, \mathbf{v}_l^c} \quad & \sum_{c \in \mathbb{C}} \omega^c \Big(\phi(\mathbf{z}_N^c, \mathbf{d}_{N-1}^0) + \sum_{l=0}^{N-1} \varphi(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^0) \Big) + \\
& \sum_{c \in \mathbb{C}} \omega^c \Big(M_\phi \mathbf{e}^T \mathbf{r}_N^c + \sum_{l=0}^{N-1} M_\varphi \mathbf{e}^T \mathbf{r}_l^c \Big) && \text{(6.9a)} \\
\text{s.t.} \quad & \mathbf{z}_{l+1}^c = \mathbf{f}(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^0) \quad l = 0, \ldots, N-1 && \text{(6.9b)} \\
& \mathbf{z}_0^c = \mathbf{x}_k && \text{(6.9c)} \\
& \mathbf{x}^L - \mathbf{r}_l^c \le \mathbf{z}_l^c \le \mathbf{x}^U + \mathbf{r}_l^c; \; \mathbf{r}_l^c \ge 0 && \text{(6.9d)} \\
& \mathbf{x}_f^L - \mathbf{r}_N^c \le \mathbf{z}_N^c \le \mathbf{x}_f^U + \mathbf{r}_N^c; \; \mathbf{r}_N^c \ge 0 && \text{(6.9e)} \\
& \mathbf{v}_l^c = \mathbf{v}_l^{c'} \quad \{(c, c') \mid \mathbf{z}_l^c = \mathbf{z}_l^{c'}\} && \text{(6.9f)} \\
& \mathbf{v}_l^c \in \mathbb{U}, \mathbf{d}_l^0 \in \mathbb{D} && \text{(6.9g)} \\
& \mathbf{d}_{l-1}^0 = \mathbf{d}_l^0 \quad l = N_r, \ldots N-1 && \text{(6.9h)} \\
& \forall c, c' \in \mathbb{C}
\end{aligned}$$

We note that if we similarly set $\mathbf{d}_l^c = \mathbf{d}_l^0, \; \forall c \in \mathbb{C}$ in the samNMPC problem (6.8), then (6.9) is also equivalent to this nominal case of (6.8), i.e. $J_{sam}^N(\mathbf{x}_k, \mathbf{d}^0) = J_{ms}^N(\mathbf{x}_k, \mathbf{d}^0)$.

6.3 Recursive feasibility for ideal multistage NMPC and samNMPC

In multistage NMPC, the true realization of the uncertainty at time step k can take the state to any of the branched scenarios at time step $k+1$ (Figure 2.2). For recursive feasibility, it is necessary to map the new scenario tree at time step $k+1$ to the one at time step k. The mapping has to be such that the probability of any particular scenario ω^c remains the same across the run time of the controller, i.e. $\omega_k^c = \omega_{k+1}^c$.

To this end, we define ω_l^j the probability of the state evolving from $\mathbf{z}_l$ to $\mathbf{z}_{l+1} = \mathbf{f}(\mathbf{z}_l, \mathbf{v}_l, \mathbf{d}_l^j)$, where $j \in \mathbb{M}$, and $l = 0, 1, \ldots, N-1$. Note that

$$\sum_{j \in \mathbb{M}} \omega_l^j = 1, \; l = \{0, \ldots, N_r - 1\} \tag{6.10}$$

The probability of each scenario $c \in \mathbb{C}$ is then calculated as:

$$\omega^c = \prod_{l=0}^{N_r-1} \omega_l^{j_l} \tag{6.11}$$

where the scenario c is represented as $c = \{j_0, j_1, \ldots, j_{N_r-1}\}$. We assume that $\omega_l^j = \omega_{l'}^j \quad \forall l, l' \in \{0, 1, \ldots, N_r - 1\}$, so that ω^c is the same for all time steps k. In this way we can handle the robust horizon with $N_r < N$. Note, however, that this does not necessarily imply $\omega^c = \omega^{c'} \quad \forall c, c' \in \mathbb{C}$.

Assumption 1.

- *For all* $\mathbf{x}$ *satisfying* (6.6d) - (6.6e) *and* $\mathbf{u} \in \mathbb{U}$ *the model dynamics and both stage and terminal costs are twice differentiable and Lipschitz continuous in all arguments, and they satisfy* $\forall j \in \mathbb{M},\ \mathbf{f}(0, 0, \mathbf{d}^j) = 0,\ \varphi(0, 0, \mathbf{d}^j) = 0,$ *and* $\phi(0, \mathbf{d}^j) = 0$ *where* $\mathbb{M}$ *is as defined in* (2.3). *We therefore assume that* (6.6d) - (6.6e) *defines an RPI set.*

- *The sets of state bounds* $\mathbb{X}$ *and the terminal region* $\mathbb{X}_f$ *are closed, and the control set* $\mathbb{U}$ *is compact. All sets contain the origin.*

- *For all* $j \in \mathbb{M}$, $\exists$ *a common* $\mathbb{X}_f$ *that is control invariant for* $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k^j)$ *and* $\mathbf{u}_k \in \mathbb{U},\ \forall x_k \in \mathbb{X}_f$*.*

Assumption 2.

- *For each parametric disturbance* $j \in \mathbb{M}$*, there exists a local control law* $\mathbf{u} = \mathbf{h}_f(\mathbf{x})$ *defined on* $\mathbb{X}_f^j$ *such that* $\mathbf{f}(\mathbf{x}, \mathbf{h}_f(\mathbf{x}), \mathbf{d}^j) \in \mathbb{X}_f^j, \forall\ \mathbf{x} \in \mathbb{X}_f^j$*, and* $\phi(\mathbf{f}(\mathbf{x}, \mathbf{h}_f(\mathbf{x}), \mathbf{d}^j), \mathbf{d}^j) - \phi(\mathbf{x}, \mathbf{d}^j) \leq -\varphi(\mathbf{x}, \mathbf{h}_f(\mathbf{x}), \mathbf{d}^j), \forall \mathbf{x} \in \mathbb{X}_f^j$*.*

- *For each parametric disturbance* $j \in \mathbb{M}$*, the stage cost* $\varphi(\mathbf{x}, \mathbf{u}, \mathbf{d}^j)$ *satisfies* $\alpha_p(|\mathbf{x}|) \leq \varphi(\mathbf{x}, \mathbf{u}, \mathbf{d}^j) \leq \alpha_q(|\mathbf{x}|) + \sigma_q(|\Delta\mathbf{d}|)$ *where* $\alpha_p(\cdot)$*,* $\alpha_q(\cdot)$ *are* $\mathcal{K}_\infty$ *functions, and* $\sigma_q(\cdot)$ *is a* $\mathcal{K}$ *function.*

- *For the relaxed nominal NMPC problem* (6.6)*, equivalent to problem* (6.9)*, the horizon* N *and the weighted terminal cost are chosen sufficiently large so that the solution of* (6.9) *satisfies* $\mathbf{z}_N \in \mathbb{X}_f$ *with* $\mathbf{r}_N = 0\ \forall\ \mathbf{x}_k$ *in the relaxation of* $\mathbb{X}$ *defined by* (6.6d)*. Existence of these solutions was shown in [97].*

From these assumptions we can state the following result:

Theorem 2. *(Recursive feasibility of ideal multistage NMPC and samNMPC) Suppose Assumptions 1 and 2 hold, then the soft-constrained problems* (6.7) *and* (6.8) *with a robust horizon* $(N_r \leq N)$ *are recursively feasible.*

Proof. Since MFCQ and GSSOSC hold for the relaxed problems (6.7) and (6.8), Lipschitz continuity holds at the optimal solutions of these problems for all values of $\mathbf{x}_k$ and $\mathbf{d}_l^c$ (see Chapter 6.2).

Comparing the ideal multistage NMPC problem (6.7) with its nominal case (6.9), it follows from NLP sensitivity that the difference of their respective solutions are bounded by:

$$\left|\Big(\mathbf{z}^*(\mathbf{d}^c), \mathbf{v}^*(\mathbf{d}^c), \mathbf{r}^*(\mathbf{d}^c)\Big) - \Big(\mathbf{z}^*(\mathbf{d}^0), \mathbf{v}^*(\mathbf{d}^0), \mathbf{r}^*(\mathbf{d}^0)\Big)\right| \leq L_S|\Delta\mathbf{d}| \tag{6.12}$$

where L_S is a positive Lipschitz constant, and $|\Delta\mathbf{d}|$ is as defined in (5.17). Similarly, we note that the same relations hold when we compare the samNMPC problem (6.8) with its nominal case (6.9) (remember that (6.9) represents the nominal case of *both* the ideal multistage and samNMPC problems).

Finally, since the system state evolves from $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \hat{\mathbf{d}}_k)$ where $\hat{\mathbf{d}}_k$ is the realization of $\mathbf{d}_k$ at time k, the difference from the nominal state $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k^0)$ is bounded by $O(|\Delta\mathbf{d}|)$. Moreover, the difference between optimal values $(\mathbf{z}_l, \mathbf{v}_l, \mathbf{r}_l)$ from (6.7) and their nominal solutions from (6.9), is also bounded by $O(|\Delta\mathbf{d}|)$. Similarly, the difference between optimal values $(\mathbf{z}_l, \mathbf{v}_l, \mathbf{r}_l)$ from (6.8) and their nominal solutions from (6.9) is bounded by $O(|\Delta\mathbf{d}|)$.

Thus, irrespective of the evolution of the scenario tree (and even with $N_r \leq N$), a nonnegative, bounded $\mathbf{r}_k$ can always be found that satisfies (6.7d) – (6.7e) for ideal multistage NMPC, and all of the (relaxed) constraints remain satisfied. In the case of samNMPC, similar arguments can be made for (6.8d) – (6.8e).

Thus, the ideal multistage NMPC problem (6.7) and the samNMPC problem (6.8) are recursively feasible. ■

6.4 Input-to-State practical stability for ideal multistage NMPC

We now consider ideal multistage NMPC and show that it is ISpS-stable.

Theorem 3. *(Robust stability of ideal multistage NMPC) Under Assumptions 1 and 2, the cost function J_{ms} obtained from the solution of (6.7) is an ISpS-Lyapunov function, and the resulting closed-loop system is ISpS stable.*

Proof. Consider the nominal case of the ideal multistage NMPC formulation as shown in (6.9). Here, since every scenario in ideal multistage NMPC becomes the nominal scenario, this problem has nominal (asymptotic) stability [1]. Let the optimal control sequence obtained from (6.9) for the nominal scenario be $\{\mathbf{v}_0^c, \mathbf{v}_1^c, \ldots, \mathbf{v}_{N-1}^c, \mathbf{h}_f(\mathbf{z}_N)\}$. Based on Theorem 2, recursive feasibility is guaranteed for (6.9) and is valid for $N_r \leq N$; and so this control sequence is also feasible for the same problem with an extended prediction horizon $N+1$. The objective function for the extended problem is given by:

$$\begin{aligned}\widetilde{J}_{ms}^{N+1}(\mathbf{x}_k, \mathbf{d}^0) &= \sum_{c\in\mathbb{C}} \omega^c \Big(\phi(\mathbf{z}_{N+1}^c, \mathbf{d}_{N+1}^0) + \sum_{l=0}^{N} \varphi(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^0)\Big) \\ &\quad + \sum_{c\in\mathbb{C}} \omega^c \Big(M_\phi \mathbf{e}^T \mathbf{r}_{N+1}^c + \sum_{l=0}^{N} M_\varphi \mathbf{e}^T \mathbf{r}_l^c\Big) \qquad (6.13)\end{aligned}$$

and the following is valid from Assumption 2:

$$\begin{aligned}\widetilde{J}_{ms}^{N+1}(\mathbf{x}_k, \mathbf{d}^0) - J_{ms}^{N}(\mathbf{x}_k, \mathbf{d}^0) &= \sum_{c\in\mathbb{C}} \omega^c \Big(\phi(\mathbf{z}_{N+1}^c, \mathbf{d}_{N+1}^0) - \phi(\mathbf{z}_N^c, \mathbf{d}_N^0)\Big) + \\ &\quad \sum_{c\in\mathbb{C}} \omega^c \varphi(\mathbf{z}_N^c, \mathbf{v}_N^c, \mathbf{d}_N^0) \\ &\leq 0 \qquad (6.14)\end{aligned}$$

Moreover, from (6.10) and (6.14) we have:

$$\begin{aligned}J_{ms}^{N}(\mathbf{x}_k, \mathbf{d}^0) &\geq \widetilde{J}_{ms}^{N+1}(\mathbf{x}_k, \mathbf{d}^0) \\ &= \varphi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_l^0) + \sum_{j\in\mathbb{M}} \omega^j J_{ms}^{N}\Big(f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_l^0), \mathbf{d}^0\Big) \\ &= \varphi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_l^0) + J_{ms}^{N}\Big(f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_l^0), \mathbf{d}^0\Big) \qquad (6.15)\end{aligned}$$

which leads to the descent property in the absence of disturbances:

$$J_{ms}^{N}\Big(f(\mathbf{x}_k,\mathbf{u}_k,\mathbf{d}_l^0),\mathbf{d}^0\Big) - J_{ms}^{N}(\mathbf{x}_k,\mathbf{d}^0) \leq -\varphi(\mathbf{x}_k,\mathbf{u}_k,\mathbf{d}_l^0) \tag{6.16}$$

With actual uncertainties the system state evolves as $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k,\mathbf{u}_k,\hat{\mathbf{d}}_k)$, where $\hat{\mathbf{d}}_k$ is the realization of $\mathbf{d}$ at time step k. In this case we have

$$J_{ms}^{N}(\mathbf{x}_{k+1},\mathbf{d}^0) - J_{ms}^{N}(\mathbf{x}_k,\mathbf{d}^0) \;\leq\; -\varphi(\mathbf{x}_k,\mathbf{u}_k,\mathbf{d}_l^0) + L_J|\hat{\mathbf{d}}_k - \mathbf{d}_k^0| \tag{6.17}$$

Finally, from (6.7) and (6.9) we have

$$\begin{aligned} J_{ms}^{N}(\mathbf{x}_k,\mathbf{d}^c) &= J_{ms}^{N}(\mathbf{x}_k,\mathbf{d}^0) + O(|\Delta\mathbf{d}|) && (6.18)\\ J_{ms}^{N}(\mathbf{x}_{k+1},\mathbf{d}^c) &= J_{ms}^{N}(\mathbf{x}_{k+1},\mathbf{d}^0) + O(|\Delta\mathbf{d}|) && (6.19)\end{aligned}$$

which can be combined with (6.17) to form:

$$\begin{aligned} J_{ms}^{N}(\mathbf{x}_{k+1},\mathbf{d}^c) - J_{ms}^{N}(\mathbf{x}_k,\mathbf{d}^c) &\leq -\varphi(\mathbf{x}_k,\mathbf{u}_k,\mathbf{d}_l^0) + L_J|\hat{\mathbf{d}}_k - \mathbf{d}_k^0| + \\ &\quad \Big(J_{ms}^{N}(\mathbf{x}_{k+1},\mathbf{d}^c) - J_{ms}^{N}(\mathbf{x}_{k+1},\mathbf{d}^0)\Big) + \\ &\quad \Big(J_{ms}^{N}(\mathbf{x}_k,\mathbf{d}^0) - J_{ms}^{N}(\mathbf{x}_k,\mathbf{d}^c)\Big) \\ &\leq -\varphi(\mathbf{x}_k,\mathbf{u}_k,\mathbf{d}_l^0) + L_J|\hat{\mathbf{d}}_k - \mathbf{d}_k^0| + 2L_K|\Delta\mathbf{d}| \\ &= -\alpha(\mathbf{x}_k) + \sigma(|\hat{\mathbf{d}}_k - \mathbf{d}_k^0|) + c_d. && (6.20)\end{aligned}$$

where $c_d \geq 0$. This result proves that ideal multistage NMPC is ISpS-stable. ■

6.5 Input-to-State practical stability for samNMPC

To extend the stability discussions to samNMPC, we compare the formulations of the relaxed standard NMPC (6.6) and relaxed samNMPC (6.8). Since MFCQ and GSSOSC are satisfied for (6.6) and (6.8), Lipschitz continuity holds for $J_{nom}(\mathbf{x}_k)$ and $J_{sam}^{N}(\mathbf{x}_k,\mathbf{d}^c)$.

Theorem 4. *(Robust Stability of samNMPC) Under Assumptions 1 and 2, the cost function J_{sam} obtained from the solution of relaxed samNMPC problem* (6.8) *is an ISpS-Lyapunov function, and the resulting closed-loop system is ISpS stable.*

Proof. From Theorem 2, recursive feasibility properties hold for samNMPC and are valid for $N_r \leq N$. To compare (6.6) and (6.8) we replicate (6.6) $|\mathbb{C}|$ times and we partition the objective function into its critical, noncritical and nominal components, $J_{sam}(\mathbf{x}_k, \mathbf{d}^c) = \widehat{J}(\mathbf{x}_k, \mathbf{d}^c) + \bar{J}(\mathbf{x}_k, \mathbf{d}^c) + \omega^0 J_{nom}(\mathbf{x}_k)$ using $\mathbb{C} = \widehat{\mathbb{C}} \cup \bar{\mathbb{C}} \cup \{0\}$ and the construction of (6.8).

For $\widehat{\mathbb{C}}$ and the solutions of (6.6) and (6.8) we note that:

$$\begin{aligned}
\widehat{J}(\mathbf{x}_k, \mathbf{d}^c) - \sum_{c \in \widehat{\mathbb{C}}} \omega^c \, J_{nom}(\mathbf{x}_k) \;&=\; \sum_{c \in \widehat{C}} \omega^c \Big((\phi(\mathbf{z}_N^c, \mathbf{d}_N^c) - \phi(\mathbf{z}_N^0, \mathbf{d}_N^0) \Big) + \\
&\quad \sum_{c \in \widehat{C}} \omega^c M_\phi e^T (\mathbf{r}_N^c - \mathbf{r}_N) + \\
&\quad \sum_{c \in \widehat{C}} \omega^c \sum_{l=0}^{N-1} \Big(\varphi(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^c) - \varphi(\mathbf{z}_l^0, \mathbf{v}_l^0, \mathbf{d}_l^0)) \Big) + \\
&\quad \sum_{c \in \widehat{C}} \omega^c \sum_{l=0}^{N-1} M_\varphi e^T (\mathbf{r}_l^c - \mathbf{r}_l) \\
&= \; O(|\Delta \mathbf{d}|)
\end{aligned} \tag{6.21}$$

For $\bar{\mathbb{C}}$ we note from the solution of (6.6) and its KKT sensitivity that

$$\begin{aligned}
\bar{J}(\mathbf{x}_k, \mathbf{d}^c) - \sum_{c \in \bar{\mathbb{C}}} \omega^c \, J_{nom}(\mathbf{x}_k) \;&=\; \sum_{c \in \bar{C}} \omega^c \Big(\phi(\mathbf{z}_N^0 + \Delta \mathbf{z}_N^c, \mathbf{d}_N^c) - \phi(\mathbf{z}_N^0, \mathbf{d}_N^0) \Big) + \\
&\quad \sum_{c \in \bar{C}} \omega^c M e^T (\mathbf{r}_N^c - \mathbf{r}_N) + \\
&\quad \sum_{c \in \bar{C}} \omega^c \sum_{l=0}^{N-1} \Big(\varphi(\mathbf{z}_l^0 + \Delta \mathbf{z}_l^c, \mathbf{v}_l^0 + \Delta \mathbf{v}_l^c, \mathbf{d}_l^c) - \varphi(\mathbf{z}_l^0, \mathbf{v}_l^0, \mathbf{d}_l^0) \Big) + \\
&\quad \sum_{c \in \bar{C}} \omega^c \sum_{l=0}^{N-1} M e^T (\mathbf{r}_l^c - \mathbf{r}_l) \\
&= \; O(|\Delta \mathbf{d}|).
\end{aligned} \tag{6.22}$$

From (6.8), (6.9), (6.21) and (6.22) we have

$$\begin{aligned}
J_{sam}^N(\mathbf{x}_k, \mathbf{d}_l^c) &= J_{sam}^N(\mathbf{x}_k, \mathbf{d}_l^0) + O(|\Delta \mathbf{d}|) \\
J_{sam}^N(\mathbf{x}_{k+1}, \mathbf{d}_l^c) &= J_{sam}^N(\mathbf{x}_{k+1}, \mathbf{d}_l^0) + O(|\Delta \mathbf{d}|)
\end{aligned} \tag{6.23}$$

which can be combined with (6.17) to form:

$$\begin{aligned}
J^N_{sam}(\mathbf{x}_{k+1}, \mathbf{d}^c_l) - J^N_{sam}(\mathbf{x}_k, \mathbf{d}^c_l) \quad &\leq \quad -\varphi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}^0_l) + L_J|\hat{\mathbf{d}}_k - \mathbf{d}^0_k|) + \\
&\Big(J^N_{sam}(\mathbf{x}_{k+1}, \mathbf{d}^c_l) - \sum_{c\in\mathbb{C}} \omega^c \, J_{nom}(\mathbf{x}_{k+1})\Big) + \\
&\Big(\sum_{c\in\mathbb{C}} \omega^c \, J_{nom}(\mathbf{x}_k) - J^N_{sam}(\mathbf{x}_k, \mathbf{d}^c_l)\Big) \\
&\leq \quad -\varphi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}^0_l) + L_J|\hat{\mathbf{d}}_k - \mathbf{d}^0_k|) + 2L_K|\Delta\mathbf{d}| \\
&= \quad -\alpha(\mathbf{x}_k) + \sigma(|\hat{\mathbf{d}}_k - \mathbf{d}^0_k|) + c_d. \qquad (6.24)
\end{aligned}$$

where $c_d \geq 0$. This result shows that samNMPC is ISpS-stable. ■

It is worth mentioning here that, in Theorems 3 and 4, the constant c_d scales with $|\Delta\mathbf{d}|$ as shown in (6.20) and (6.24). Equation (5.17) shows that $|\Delta\mathbf{d}|$ is the maximum difference in the uncertainty vectors between any scenarios. In context of ISpS, this allows for potentially large values of the constant c_d if the uncertainty range is large. Although ISpS still stands, a large uncertainty range would somewhat weaken the above results.

6.6 Conclusion

By relaxing the bound constraints on the states, it is shown that both ideal multistage and samNMPC are recursively feasible under the robust horizon formulation. In addition, the ideal multistage and samNMPC formulations are compared with their nominal cases, and the corresponding differences in their value functions are shown to be bounded. Thus it is shown that ideal multistage and samNMPC are ISpS-stable, even under the robust horizon formulation.

Chapter 7

Sensitivity-assisted Multistage Nonlinear Model Predictive Control with Path-Following

In Chapter 5, we looked at an approximate multistage NMPC strategy with a pruned scenario tree consisting of critical scenarios, while the noncritical scenarios were approximated with one-step linear sensitivity steps in the objective function. The sensitivity steps were calculated according to the linear system (5.1), which comes from the application of Theorem 1 to the optimality conditions of (2.5). A key assumption of Theorem 1 is strict complementarity (SC), which implies that the active set does not change for any p in the neighborhood of p_0. The SC assumption usually holds for small perturbations, but for a large $\Delta p = p - p_0$, there may very well be changes in the active set. Neglecting such changes in the active set typically will affect the accuracy of the obtained solution.

A more general approach allows for changes in the active set and several sensitivity updates along a path from $p = p_0$ to $p = p$. Such multi-step path-following algorithms have been proposed to improve the NLP sensitivity predictions and accuracy. For instance, the advanced-step NMPC framework (proposed by [83]), was implemented for economic NMPC with a predictor path-following method in [98]. In this method, active set changes are rigorously handled by tracking the optimal solution path from $p = p_0$ to $p = p$ with a sequence of pure-predictor quadratic programs (QPs). This was further improved in [99] where the path-following algorithm employed a sequence of predictor-corrector QPs. In context of ideal multistage NMPC, this latter algorithm was used in [100] for facilitating a scenario decomposition approach to speed up online computations. A more rigorous treatment of path-following algorithms that also allow for non-unique Lagrange multipliers can be found in [98, 101, 102, 103].

In this chapter, we extend the samNMPC framework with a path-following algorithm based on predictor-corrector QPs. This approach, which we call sampfNMPC, has subtle but significant differences with the samNMPC approach. Here, we formulate an approximate multistage NMPC problem with a reduced scenario tree consisting of the critical and nominal scenarios, but unlike in samNMPC, the noncritical scenario costs are first *equated* to the nominal scenario cost in the objective function. From the solution of this approximate problem, we path-follow from the nominal uncertainty representing the noncritical scenarios to the actual noncritical uncertainty representing the noncritical scenarios. In essence, the reduced NLP first handles the critical (and nominal) scenarios, and the noncritical scenarios are subsequently accounted for via path-following.

The following sections expand upon the path-following algorithm with predictor-corrector QPs and its application in the sampfNMPC approach. We compare the results of ideal multistage NMPC and sampfNMPC in terms of robustness, for the Quadtank case study. The contents of the this chapter are part of the preliminary draft [33].

7.1 Sensitivity-based path-following

7.1.1 Predictor-corrector QP

For the parametric NLP (2.5), recall that the active set of bounds at x^* is defined as $J = \{j \mid x_j^* = 0\}$. We define the strongly active set as $J_+ = \{j \mid \nu_j > 0\}$ and the union of the weakly active and inactive sets as $J_0 = \{j \mid \nu_j = 0\}$. It can be shown that the strongly active set does not change in the small neighborhood of p_0. Thus, we can formulate a QP related to the sensitivity system (5.1) such that we stay on the strongly active constraints in this small neighborhood. In fact, the primal-dual solution of this QP is the directional derivative of the primal-dual solution $s^*(p)$ of the parametric NLP (2.5).

Theorem 5. *(NLP Sensitivity) If $F(\cdot)$ and $c(\cdot)$ of the parametric NLP problem* (2.5) *are* *twice continuously differentiable in a neighborhood of the nominal primal and dual solution $s^*(p_0)$ and this solution satisfies the LICQ and SSOSC, then,*

- *$s^*(p)$ is Lipschitz continuous in a neighborhood of $s^*(p_0)$.*

- *$s^*(p)$ is directionally differentiable.*

- *The directional derivative uniquely solves the following QP:*

$$\begin{aligned}
\min_{\Delta x} \quad & \frac{1}{2}\Delta x^T \nabla_{xx}\mathcal{L}(s^*(p_0))\Delta x + \Delta x^T \nabla_{xp}\mathcal{L}(s^*(p_0))\Delta p && \text{(7.1a)}\\
\textit{s.t.} \quad & \nabla_x c(s^*(p_0))^T \Delta x + \nabla_p c(s^*(p_0))^T \Delta p = 0 && \text{(7.1b)}\\
& \Delta x_j = 0 \quad j \in J_+ && \text{(7.1c)}\\
& \Delta x_j \geq 0 \quad j \in J_0 && \text{(7.1d)}
\end{aligned}$$

Proof: See [104] (Section 5.2), and [105] (Proposition 3.4.1). ■

As opposed to Theorem 1, the SC condition is not needed for the above result. For the small Δp, the theorem above gives the approximate solution of (2.5) at $p_0 + \Delta p$. The solution of the QP (7.1) is a first-order estimate of the change in the solution of (2.5) from $p = p_0$ to $p = p_0 + \Delta p$ [101]. This QP is often referred to as a pure-predictor QP. A key difference between calculating the sensitivity step using the QP (7.1) and the linear system (5.1) is that with the QP, we account for the correct active set by staying on the strongly active constraints as shown in (7.1c). Note that in the absence of weakly active constraints (i.e. if SC holds), the sensitivity step obtained from (7.1) is equivalent to that obtained from (5.1).

A pure-predictor QP, however, may not lead to a sufficiently accurate approximation. A common approach to improve accuracy is to add a corrector step separately (see for instance [101, 102, 103]) to push the estimate towards the true solution. In this chapter, we consider a corrector term that is directly incorporated into the QP formulation as shown in [99]. This predictor-corrector QP approximation of the parametric NLP (2.5) is formed by including a corrector term in the objective function of the QP. The QP constraints are formed by linearizing the NLP constraints with respect to x and p, and again enforcing the strongly active inequalities as equalities. With the technical assumption that p enters linearly in the constraints, we formulate the following predictor-corrector QP:

$$\min_{\Delta x} \quad \frac{1}{2}\Delta x^T \nabla_{xx}\mathcal{L}(s^*(p_0);p)\Delta x + \nabla_x F(s^*(p_0);p)^T \Delta x \tag{7.2a}$$

$$\text{s.t.} \quad c(s^*(p_0);p) + \nabla_x c(s^*(p_0);p)^T \Delta x = 0 \tag{7.2b}$$

$$x_j + \Delta x_j = 0 \quad j \in J_+ \tag{7.2c}$$

$$x_j + \Delta x_j \geq 0 \quad j \in J_0 \tag{7.2d}$$

The predictor component in the QP (7.2) estimates a first-order approximation of the perturbed NLP, and this estimate is improved with the corrector component that makes it more closely aligned with satisfying the KKT conditions at the new point [101]. Note that in (7.2), the gradients, Jacobians and Hessians are computed at the updated parameter value $p = p_0 + \Delta p$. The solution of (7.2) thus leads to good approximations of the NLP solution for a small Δp. However, in the context of sampfNMPC, we want to approximate the noncritical scenario solutions from the nominal scenario solution, and here the Δp (which is the difference between the nominal and noncritical uncertainty) may not be small. This necessitates a multi-step path-following approach that solves a sequence of predictor-corrector QPs with small Δp's, as explained in the following section.

7.1.2 Path-following algorithm

The optimal solution path from p_0 to p_f can be parameterized with the path-following parameter t_k such that:

$$p(t_k) = (1 - t_k)p_0 + t_k p_f \tag{7.3}$$

The predictor-corrector QP (7.2) is solved at each $p(t_k)$ with the path sequence $t_1, \ldots, t_K$, where $0 < t_1 < \ldots < t_k < \ldots < t_K = 1$. The primal-dual solution is updated along the path according to:

$$x(t_{k+1}) = x(t_k) + \Delta x \tag{7.4a}$$

$$\lambda(t_{k+1}) = \Delta\lambda \tag{7.4b}$$

$$\nu(t_{k+1}) = \Delta\nu \tag{7.4c}$$

where Δx is obtained from the solution of (7.2), and $\Delta\lambda$ and $\Delta\nu$ are the corresponding multipliers of (7.2). This approach is similar to the Euler integration for solving ODEs on a discretized time interval.

As explained in [99], if a constraint changes from strongly active to inactive at some point along the path, the corresponding multiplier will become $\nu_j = 0$, and it will be added to J_0 for the next QP. The next QP solution can then move away from this constraint as it is imposed as an inequality. Similarly, if constraint becomes active, the corresponding multiplier will be added to J_+ for the next QP, forcing the subsequent solution to be tracked further along the equality. The path-following algorithm is shown in Algorithm 2.

Algorithm 2: Path-following algorithm (adapted from [99])

```
Given:
  • Initial parameter $p_0$, and final parameter $p_f$
  • initial number of points along the path $K$
  • constant $0 < \alpha < 1$
Input: Primal-dual solution of NLP (2.5) at $p_0$, $s^*(p_0)$
Output: Approximate primal-dual solution of NLP (2.5) at $p_f$, $\tilde{s}(p_f)$
set $k = 1$, $t_0 = 0$, $\Delta t = 1/K$
while TRUE do
    Update $t_k = t_{k-1} + \Delta t$
    Compute $p(t_k) = (1 - t_k)p_0 + t_k p_f$
    Solve predictor-corrector QP (7.2) at $p(t_k)$    /* to obtain $\Delta x$, $\Delta\lambda$, $\Delta\nu$ */
    if QP is feasible then
        Update primal-dual variables according to (7.4)
        Update $k \leftarrow k + 1$
    else
        Reduce QP stepsize $\Delta t \leftarrow \alpha \Delta t$
    end
    if $t_k \geq 1$ then
        Return updated primal-dual variables    /* $x$, $\lambda$, $\nu$ */
        break
    end
end
```

7.2 Sensitivity-assisted multistage NMPC scheme with path-following

The sampfNMPC approach, in a nutshell, can be summed up in a two-step process:

1. explicitly handle the uncertainties of the critical and nominal scenarios through a reduced scenario tree formulation, and then

2. path-follow our way to the uncertainties of the noncritical scenarios to arrive at the (approximate) solution of the ideal multistage NMPC problem.

To this end, we first formulate the sets of critical and noncritical scenarios as explained in Chapter 5.2.1.

7.2.1 Reduced NLP for critical and nominal scenarios

Recall that the critical and noncritical scenario sets are denoted by $\widehat{\mathbb{C}}$ and $\bar{\mathbb{C}}$. The NLP formulation in the reduced space of critical and nominal scenarios is as follows:

$$\begin{aligned}\min_{\substack{\mathbf{z}_l^c,\mathbf{v}_l^c\\ c\in\widehat{\mathbb{C}}\cup\{0\}}} \quad & \sum_{c\in\widehat{\mathbb{C}}\cup\{0\}} \omega_c\Big(\phi(\mathbf{z}_N^c,\mathbf{d}_{N-1}^c)+\sum_{l=0}^{N-1}\varphi(\mathbf{z}_l^c,\mathbf{v}_l^c,\mathbf{d}_l^c)\Big)+\\ & \sum_{c\in\bar{\mathbb{C}}}\omega_c\Big(\phi(\mathbf{z}_N^0,\mathbf{d}_{N-1}^0)+\sum_{l=0}^{N-1}\varphi(\mathbf{z}_l^0,\mathbf{v}_l^0,\mathbf{d}_l^0)\Big) \qquad (7.5a)\end{aligned}$$

$$\begin{aligned}\text{s.t.} \quad & \mathbf{z}_{l+1}^c=\mathbf{f}(\mathbf{z}_l^c,\mathbf{v}_l^c,\mathbf{d}_l^c) \quad l=0,\dots,N-1 && (7.5b)\\ & \mathbf{z}_0^c=\mathbf{x}_k && (7.5c)\\ & \mathbf{v}_l^c=\mathbf{v}_l^{c'} \quad \{(c,c')\mid \mathbf{z}_l^c=\mathbf{z}_l^{c'}\} && (7.5d)\\ & \mathbf{z}_l^c\in\mathbb{X},\mathbf{v}_l^c\in\mathbb{U},\mathbf{z}_N^c\in\mathbb{X}_f,\mathbf{d}_l^c\in\mathbb{D} && (7.5e)\\ & \mathbf{d}_{l-1}^c=\mathbf{d}_l^c \quad l=N_r,\dots N-1 && (7.5f)\\ & \forall c,c'\in\widehat{\mathbb{C}}\cup\{0\}\end{aligned}$$

Importantly, the noncritical scenario costs are approximated by the nominal scenario costs (with corresponding weights) in the objective function (7.5a). We denote the concatenated vector of all uncertainties in the reduced problem (with the subscripts $_{l=0,\dots,N-1}$ omitted for brevity) as:

$$\mathbf{d}_{reduced} := \begin{bmatrix} \mathbf{d}^0 \\ \hline \mathbf{d}^c \\ \vdots \\ {\scriptstyle (\forall c \in \widehat{\mathbb{C}})} \end{bmatrix} \tag{7.6}$$

We further denote the primal-dual solution vector of (7.5) as:

$$s_{reduced} := \begin{bmatrix} s^0 \\ \hline s^c \\ \vdots \\ {\scriptstyle (\forall c \in \widehat{\mathbb{C}})} \\ \hline \widehat{\gamma} \end{bmatrix} \tag{7.7}$$

where each s^c denotes the primal-dual solution vector for the corresponding scenario variables, and $\widehat{\gamma}$ denotes the vector of the multipliers for the nonanticipativity constraints (7.5d).

The formulation (7.5) can be thought of as a "partial" representation of the ideal multistage NMPC problem (2.4), albeit in the reduced space, where the noncritical scenario uncertainty is equated to the nominal scenario uncertainty. This is because the nominal cost replaces all the noncritical scenario costs in the objective function, and all the noncritical scenario constraints (including the corresponding nonanticipativity constraints) are represented all at once by the nominal scenario constraints (7.5b)–(7.5f) for $\mathbf{d}_l^0$.

7.2.2 Path-following to noncritical scenarios

The aim here is to path-follow from the "partial" solution of the ideal multistage problem (2.4) obtained from (7.5), to the "full" solution of (2.4) where the noncritical scenario uncertainties are represented by their true values. Note that this path-following from the partial solution to the full solution of (2.4) thus needs to happen in the full space of all scenarios, i.e. the predictor-corrector QPs (7.2) along the path should represent the full NLP (2.4). Since we obtain the solution vector (7.7) of the reduced NLP, we can extrapolate it to represent the partial solution of (2.4) as follows:

$$s_{reduced} := \left[\begin{array}{c} s^0 \\ \hline s^c \\ \vdots \\ {\scriptstyle (\forall c \in \widehat{\mathbb{C}})} \\ \hline \widehat{\gamma} \end{array}\right] \Longrightarrow \left[\begin{array}{c} s^0 \\ \hline s^c \\ \vdots \\ {\scriptstyle (\forall c \in \widehat{\mathbb{C}})} \\ \hline s^0 \\ \vdots \\ {\scriptstyle (\forall c \in \bar{\mathbb{C}})} \\ \hline \widehat{\gamma} \\ \bar{\gamma} \end{array}\right] =: s_{ms,partial} \tag{7.8}$$

In this way, the noncritical scenario solutions are represented by the nominal scenario solution in the extrapolated vector. Note that the NAC multipliers associated with noncritical scenarios $\bar{\gamma}$ do not come from $s_{reduced}$, and are chosen arbitrarily. Thus, the solution vector $s_{ms,partal}$ represents the solution of the ideal multistage NMPC problem (2.4) with the partial uncertainty vector:

$$\mathbf{d}_{ms,partial} := \left[\begin{array}{c} \mathbf{d}^0 \\ \hline \mathbf{d}^c \\ \vdots \\ {\scriptstyle (\forall c \in \widehat{\mathbb{C}})} \\ \hline \mathbf{d}^0 \\ \vdots \\ {\scriptstyle (\forall c \in \bar{\mathbb{C}})} \end{array}\right] \tag{7.9}$$

The derivative information (gradients, Jacobians, Hessians) from the ideal multistage NMPC problem (2.4) is used to formulate the predictor-corrector QP (7.2). We can now solve a sequence of these predictor-corrector QPs, path-following from $p_0 = \mathbf{d}_{ms,partial}$ to $p_f = \mathbf{d}_{ms,full}$:

$$\mathbf{d}_{ms,partial} := \left[\begin{array}{c} \mathbf{d}^0 \\ \hline \mathbf{d}^c \\ \vdots \\ {\scriptstyle (\forall c \in \widehat{\mathbb{C}})} \\ \hline \mathbf{d}^0 \\ \vdots \\ {\scriptstyle (\forall c \in \bar{\mathbb{C}})} \end{array}\right] \to \cdots \to \cdots \to \left[\begin{array}{c} \mathbf{d}^0 \\ \hline \mathbf{d}^c \\ \vdots \\ {\scriptstyle (\forall c \in \widehat{\mathbb{C}})} \\ \hline \mathbf{d}^c \\ \vdots \\ {\scriptstyle (\forall c \in \bar{\mathbb{C}})} \end{array}\right] =: \mathbf{d}_{ms,full} \tag{7.10}$$

Consequently, we reach the (approximate) full solution vector $\tilde{s}_{ms,full}$ of (2.4) from its partial solution vector:

$$s_{ms,partial} := \left[\begin{array}{c} s^0 \\ \hline s^c \\ \vdots \\ {\scriptstyle (\forall c \in \widehat{\mathbb{C}})} \\ \hline s^0 \\ \vdots \\ {\scriptstyle (\forall c \in \bar{\mathbb{C}})} \\ \hline \widehat{\gamma} \\ \bar{\gamma} \end{array}\right] \rightarrow \cdots \rightarrow \cdots \rightarrow \left[\begin{array}{c} \tilde{s}^0 \\ \hline \tilde{s}^c \\ \vdots \\ {\scriptstyle (\forall c \in \widehat{\mathbb{C}})} \\ \hline \tilde{s}^c \\ \vdots \\ {\scriptstyle (\forall c \in \bar{\mathbb{C}})} \\ \hline \tilde{\gamma} \end{array}\right] =: \tilde{s}_{ms,full} \tag{7.11}$$

where $\tilde{\gamma}$ are the multipliers of all NACs (2.4d). The NACs (2.4d) are linear equality constraints, and so these are automatically imposed in the corresponding QPs also as equalities. This ensures that NACs are satisfied at the full solution $\tilde{s}_{ms,full}$, despite choosing the initial $\bar{\gamma}$ arbitrarily in (7.8).

Note that we only ever solve the predictor-corrector QP (7.2) associated with the ideal multistage NMPC (2.4), and never the full NLP (2.4) itself. Also note that, in theory, if we path-follow along an infinite number of points along the path (i.e. with Δp infinitesimally small at each segment of the path), we will reach the exact solution the ideal multistage NMPC problem (2.4).

7.2.3 Overall sampfNMPC algorithm and implementation

The overall sampfNMPC strategy is summarized in Algorithm 3.

Algorithm 3 is implemented using `JuMP` (version 0.19.2) [62], a tool works within the framework of the `Julia` (version 1.0.3) programming language [63]. The NLP solver used within this framework is `IPOPT` [53], and the `MA57` linear solver from the HSL libraries [79] is used within `IPOPT`. We also use `IPOPT` to solve the QPs, with the "mehrotra_algorithm" option enabled [106]. All computational experiments are carried out with an Intel i7-7600 Quad Core CPU at 2.8 GHz and 16GB RAM.

`JuMP` allows for querying the derivative information directly from a created optimization model at a given point, without having to explicitly solve the model. Thus we create (but not solve) a model for the ideal multistage NMPC problem (2.4) in `JuMP`, and initialize its derivative information, so that the necessary gradients, Jacobians and Hessians that

Algorithm 3: Sensitivity-assisted multistage NMPC with path-following

```
Given: {max, nominal, min} of all uncertain parameters.
for $k = 1, 2, \ldots$ do
    Get the current state of the plant $\mathbf{x}_k$.
    Solve the standard NMPC problem (2.2) for the nominal uncertainty $\mathbf{d}_k^0$, and
      get the KKT matrix $\mathbf{K}_0$ at the optimal solution.
    Critical scenarios: Extract $\nabla_{\mathbf{z}}\mathbf{c}$ and $\nabla_{\mathbf{d}}\mathbf{c}$ from $\mathbf{K}_0$, and solve (5.11) to form the
      critical scenario set $\widehat{\mathbb{C}}$.
    Reduced NLP: Solve the reduced NLP (7.5) to get $s_{reduced}$, and extrapolate this
      to get $s_{ms,partial}$ according to (7.8).
    Formulate QP: Use derivative information from ideal multistage NMPC
      formulation (2.4) to formulate the associated predictor-corrector QP (7.2).
    Path-follow: Apply Algorithm 2 to path-follow from $s_{ms,partial}$ to $\tilde{s}_{ms,full}$ as
      the uncertainty changes from $\mathbf{d}_{ms,partial}$ to $\mathbf{d}_{ms,full}$.
    Set $\mathbf{u}_k = \mathbf{v}_0^c, \; c \in \mathbb{C}$ and inject into the plant.
end
```

go into the QP formulation (7.2) can be calculated at the appropriate points along the path.

An interesting point to discuss relating to this algorithm is how the path-following affects the critical scenario set. Recall that the computation of critical scenarios in sampfNMPC is done in the same way as sampfNMPC, under the strong assumption of strict monotonicity. Now with path-following, the changes in the active set can reveal more information about which scenarios are actually likely to violate constraints in the absence of strict monotonicity. The scenarios corresponding to the inequality constraints that change from inactive to active along the path can then be added to the critical scenario set $\widehat{\mathbb{C}}$, and the sampfNMPC algorithm can then be reinitiated with the new $\widehat{\mathbb{C}}$. However, this rigorous treatment of critical scenarios has not been considered in this thesis, and is an avenue for further research on the topic.

7.3 Case study

We consider the Quadtank model from Chapter 5.3.2 for setpoint tracking in a system of four interconnected tanks. The model dynamics are given in (5.23), and supplementary information relevant to the model is given in Tables C.3, C.4 and C.5.

The task is to track the water levels in the lower tanks at 14 cm, while respecting level bounds in the upper tanks. Note that, as before, we introduce predefined pulses at time steps 50 and 100 to reinitialize the controller tracking. The objective function is also the same as in (5.24).

We consider that γ_1 and γ_2 are uncertain with the discrete realizations $\{$0.25, 0.4, 0.55$\}$. Thus we have 9 scenarios for a robust horizon of $N_r = 1$. The true realizations of γ_1 and γ_2 are chosen randomly for each time step from $\{$0.25, 0.4, 0.55$\}$. A prediction horizon of $N = 10$ time steps is chosen with each time step of length $10s$. The MPC simulation is executed for 150 time steps. Further, we consider an initial $K = 4$ points for the path-following Algorithm 2.

The results, shown in Figure 7.1, show overlapping tracking performance for ideal multistage NMPC and sampfNMPC (see plots for Δx_1 and Δx_2). In terms of robustness as well, the performances of both approaches are almost identical. Importantly, neither approach results in constraint violations (see plots for x_3 and x_4). Moreover, both approaches have very similar input usage (see plots for u_1 and u_2).

A comparison of the computation times between the ideal multistage and the sampfNMPC approaches, averaged across all the time steps shown in the Figure 7.1 simulation, is as follows. The ideal multistage NMPC approach takes, on average, 0.376 CPUs to solve (2.4). On the other hand, the reduced NLP of the sampfNMPC approach takes only 0.172 CPUs to solve (7.5), on average, owing to its smaller size. However, with the current software implementation, the path-following QPs add significantly to the sampfNMPC computation time, taking on average 0.526 CPUs to solve (7.2) multiple times along the path. The main reason for this is that we do not use specialized QP solvers, and solving the relatively small NLP (2.4) directly is efficient.

We expect sampfNMPC to be significantly faster than ideal multistage NMPC for larger problems with longer robust horizons, and especially with specialized QP solvers. However, at the time of writing this thesis, there are certain software-related computational challenges with our implementation of the sampfNMPC algorithm, that do not allow us to consider longer robust horizons. Specifically, the computational bottleneck pertains to building the large QP models in our software. Although solving these QP models was found to be relatively fast under limited tests, generating these models is prohibitively expensive in the current implementation.

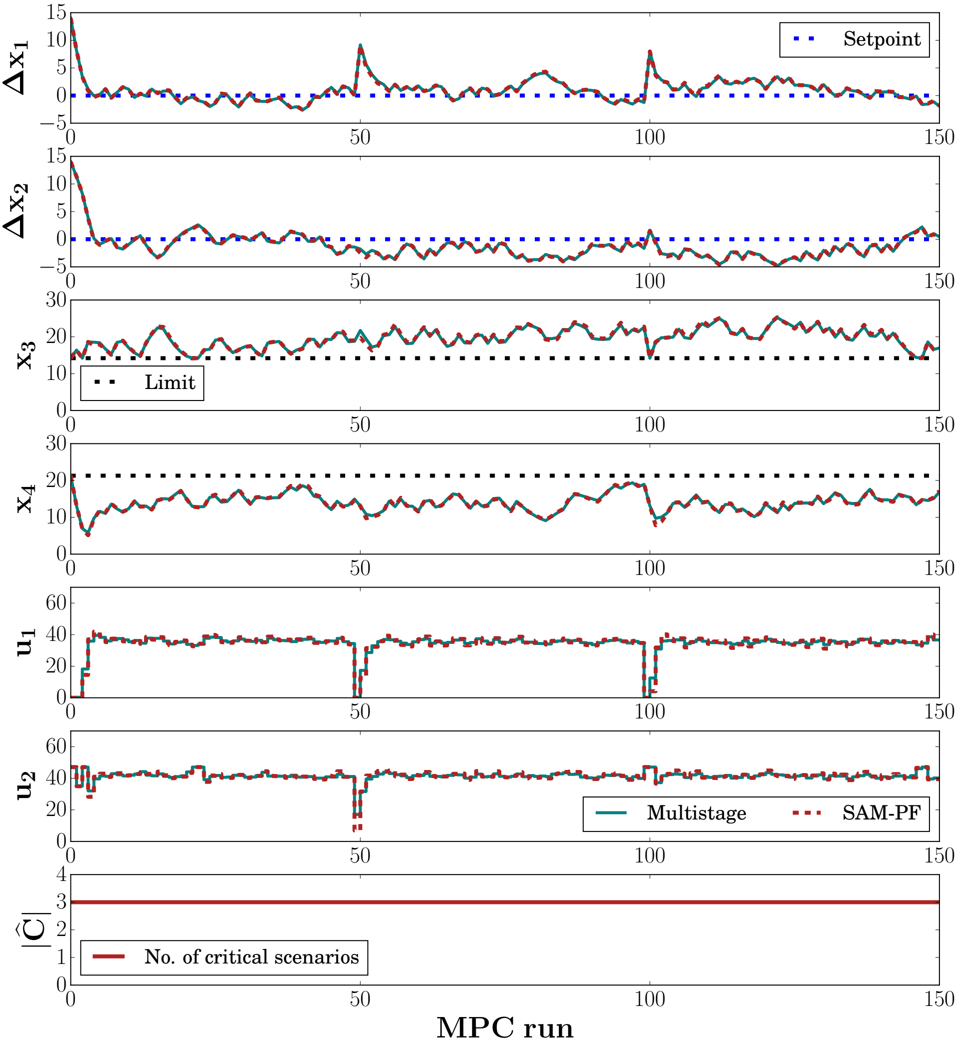


Figure 7.1: Setpoint tracking (x_1, x_2) and constraint satisfaction (x_3, x_4) for ideal multistage and sampfNMPC ($N_r = 1$ i.e. 9 scenarios).

7.4 Conclusion

This chapter extends upon the samNMPC approach presented in Chapter 5, with a path-following algorithm. In sampfNMPC, instead of one-step linear sensitivity-based approximations for the noncritical scenarios, we use a multi-step path-following approach based on predictor-corrector QPs. The benefit here is that sensitivity-based approximate solutions can be sought for larger changes in parameters. Another key advantage is that such an approach accounts for the correct active constraint set at all points along the path.

In the sampfNMPC approach, a reduced NLP is solved to get the partial solution of the ideal multistage NMPC, where noncritical uncertainties are approximated as the nominal uncertainty. This partial solution serves a starting point to path-follow to the full solution of ideal multistage NMPC, where all uncertainties are accounted for. We demonstrate the performance of the proposed approach on the Quadtank case study, and show that it performs as well as the ideal multistage NMPC in terms of robustness and setpoint tracking. Currently, there are significant software-related challenges in terms of computational efficiency of the proposed approach, which we discuss further in the concluding remarks of this thesis.

Conclusion and Future Work

Conclusion and Future Work

Summary and contributions

This thesis has examined the robust multistage NMPC problem and proposed new methods and algorithms to further its practical applicability. These novel approaches have sought to retain the key advantages of the multistage NMPC formulation, while overcoming some of its limitations. In particular, the focus has been on improving the important aspects of scenario selection and computational efficiency. To this end, the first part of the thesis has shown that employing multivariate data-analysis is prudent when tailoring the scenario tree to the given application, so that interdependencies within the uncertainty data are fully leveraged. The second part of the thesis has demonstrated, through the use of nonlinear optimization theory and sensitivity analysis, a framework that drastically shrinks the size of the scenario tree (especially for longer robust horizons), offers robustness in terms of performance and stability, and is computationally fast.

Chapter 1 gives an overview of the salient features of the multistage NMPC approach in the context of robust NMPC, and identifies some pertinent issues that motivate the topics of this thesis. Chapter 2 discusses the relevant background for multistage NMPC, in terms of the some historical context as well as its practical optimization formulation, and lays the foundation for discussing the novel approaches in this thesis.

Introducing Part I of the thesis, Chapter 3 motivates the need for data-driven scenario selection to get a better representation of the uncertainty. Using PCA, the variability in large correlated data sets in explained in fewer dimensions. This method has a twopronged advantage: the solutions are less conservative because the scenario tree is more compact, and the problem size is smaller because of the dimensionality reduction. Further, time varying uncertainty sets are considered, and an online adjustment strategy for the scenario tree is shown. The online updates are in the form of PCA-driven scenario selection and a changing length of the robust horizon, depending on availability of new data. The main contributions of Chapter 3 are as follows:

- Proposed a scenario selection strategy based on PCA.
- Proposed a dynamic update strategy for the scenario tree in face of time-varying uncertainty.
- Compared the conventional scenario selection approach with PCA-based scenario selection for a simple thermal energy storage model.
- Compared the performance of the approach with respect to a constant and a dynamic robust horizon, for the simple thermal energy storage model.

Chapter 4 demonstrates the practical applicability of the PCA-based scenario selection strategy for a detailed case study motivated by industrial data. A brief background of thermal energy storage systems is given in the context of optimal operation. Uncertainty in heat supply and demand is considered, with the objective of minimizing the use of peak heating sources, and satisfying temperature constraints for the participating plants. Thorough analysis of the industrial data set reveals correlations in the heat supply and demand, and allows for the use of PCA-based multistage NMPC. The approach is shown to provide robustness in terms of keeping the relevant temperatures within bounds even when the actual heat supply and demand profiles are taken from a completely different data set. The main contributions are listed below:

- Developed a detailed heat exchanger model where heat transfer from suppliers to consumers is facilitated through thermal energy storage.
- Performed rigorous preprocessing of the industrial data to identify where diurnal thermal energy storage is most relevant, and where PCA-based scenarios selection can be used.
- Employed PCA for outlier detection to get rid of data samples that are not representative of the uncertainty.
- Compared the PCA-based multistage NMPC approach with a standard NMPC approach to demonstrate robust constraint handling.

Chapter 5 introduces Part II of the thesis, presenting the sensitivity-assisted multistage NMPC (samNMPC) approach. Here, a critical scenario set is formed by going through the inequality constraints and ascertaining which uncertainty realizations are most likely to cause the inequality to be violated. One of the main highlights of this approach is that the critical scenarios only increase linearly with the robust horizon length, allowing for consideration of problems with longer robust horizons. This allows for solving an approximate multistage NMPC problem with these few scenarios. Moreover, the rest of the scenarios are approximated based on their NLP sensitivities in the objective function, and thus do not increase the problem size. Computational results are shown for two examples from literature. The work done for this chapter includes several contributions as shown below:

- Developed the samNMPC framework as an approximation strategy for multistage NMPC
- Generalized the computation of critical scenario set for any length of the robust horizon, based on the KKT matrix of a parameterized standard NMPC problem.
- Performed the sensitivity calculations for noncritical scenarios using a quick buildup of an approximate multistage KKT matrix, further exploiting its structure using the Schur complement decomposition.
- Applied the proposed framework to track setpoints in benchmark CSTR and Quad-tank examples for up to a robust horizon length of 3 (729 scenarios), showing robust constraint handling and significant reduction in computational cost compared to conventional multistage NMPC, especially for longer robust horizons.
- Compared the sensitivity computation times with and without Schur complement decomposition, in the context of parallelization.
- In terms of software, wrote a piece of code in `Julia` that auto-generates the NACs and the Jacobian of NACs for any given number of uncertain parameters and any given length of the robust horizon.
- In terms of software, wrote a piece of code in `Julia` that computes NLP sensitivity steps for any given parametric NLP.

In Chapter 6, the recursive feasibility and stability properties of ideal multistage NMPC and samNMPC are investigated. The different NMPC problems are reformulated with soft inequality constraints using slack variables, and it is shown that Lipschitz continuity and

NLP sensitivity properties hold for the relaxed problems. Consequently, the bounded nature of the slack variables lends itself to the proof of recursive feasibility for both formulations. Comparing the two formulations with their nominal case, ISpS is shown to hold. The main contributions of this chapter are as follows:

- Proved recursive feasibility of ideal multistage NMPC under the robust horizon assumption.
- Proved recursive feasibility of samNMPC under the robust horizon assumption.
- Proved ISpS for ideal multistage NMPC under the robust horizon assumption.
- Proved ISpS for samNMPC under the robust horizon assumption.

In Chapter 7, a path-following sensitivity update strategy is used to improve the accuracy of samNMPC. Solving a reduced problem over critical and nominal scenarios, the sampfNMPC approach first seeks a the partial solution of the multistage NMPC problem. Having obtained the partial solution, it path-follows along all the noncritical scenarios to reach the full multistage solution. The path-following algorithm employs predictor-corrector QPs that keep track of changes in the active set along the path. Although some computational challenges remain in terms of the software implementation, sampfNMPC is shown to perform extremely well in terms of robust constraint handling and tracking for an illustrative example. Listed below are the main contributions from this chapter:

- Proposed the sampfNMPC framework.
- Proposed an extrapolation strategy to get the partial solution of multistage NMPC from the reduced NLP solution.
- In terms of software, wrote a piece of code in `Julia` that computes NLP sensitivity updates using the path-following algorithm for any given parametric NLP.

Recommendations for future work

A few avenues for further work in the topics of this thesis are presented here. Some of these are broad recommendations, while some are specific to the proposed algorithms.

Data-driven scenario selection

The use of PCA for selection of scenarios, as was done in Part I of this thesis, has some obvious limitations. Since PCA is a linear transformation technique, it can only reveal correlations that are linear in nature. Although this was the case in the case studies demonstrated in Part I, process data can exhibit interdependencies that are inherently nonlinear. To exploit such data sets, data classification techniques such as the nonlinear support vector machines can be employed. These techniques use so-called kernel functions to project the data into a higher-dimensional space, where it can then be linearly separated [107].

A robust NMPC technique that combines multistage NMPC with tube-based NMPC is considered in [108]. The idea is to use multistage NMPC as a primary controller to compute different state trajectories for the "important" uncertainties, and to use an ancillary controller to track these state trajectories. The proposed framework can be thought of in the context of PCA-based scenario selection. Scenarios along the dominant principal components can be incorporated in the primary controller, whereas an ancillary control law is sought in the direction of the "insignificant" principal components, which were discarded in scenario selection procedure in this work.

Scenarios chosen along the dominant and insignificant principal components can also be seen as the critical and noncritical scenarios of the samNMPC or sampfNMPC frameworks discussed in Part II of the thesis. Since variability is low along the insignificant principal components, the corresponding noncritical scenarios will be well represented by their NLP sensitivity updates.

Thus there are potential synergies to be explored between the following approaches:

1. the combined multistage and tube-based NMPC as proposed in [108],
2. PCA-based multistage NMPC proposed in Part I, and
3. samNMPC / sampfNMPC proposed in Part II.

samNMPC and sampfNMPC implementation

In the current implementation of samNMPC, although Schur complement decomposition is used, it is not parallelized (remember that only *theoretical* parallel estimates are reported in Chapter 5). Extending the framework with parallel computing will show significant bump in computational performance. The samNMPC is an approximation strategy, but since it retains its multistage structure, the computational effort can also be aided by employing other NLP decomposition strategies. Another option to speed up computations is to integrate samNMPC with the advanced-step multistage NMPC proposed in [28]. The approach also needs to be tested on large-scale industrial case studies with many uncertain parameters, where it becomes necessary to branch the scenario tree further in time in order to better represent the uncertainty.

The sampfNMPC approach is shown to work well and compares favorably with ideal multistage NMPC from the robustness perspective. However, at the time of writing this thesis, there are two challenges with the software implementation of the sampfNMPC approach. First, the QP model generation time is a bottleneck with the current implementation. This is because there are thousands of QP constraints even for small problems, and most of these constraints each involve computing the dot product of large vectors (see the constraints (7.2b) for example). Although the QPs are solved fast, they take a long while to be generated at each point along the path, limiting the analysis in Chapter 7 to only $N_r = 1$. It also does not allow a fair comparison to be made between the computation times of sampfNMPC and ideal multistage NMPC. A way to resolve this issue is to reuse QP models from the previous point in the path and only update those constraints that represent a change in the active set. In this way, the entire QP model need not be regenerated at every point in the path. Second, the current implementation does not use specialized QP solvers. Since the QPs represent a multistage problem, the corresponding KKT matrices have a block-bordered-diagonal structure that can be exploited using Schur complement decomposition. Solving the QPs based on a parallelized Schur complement implementation would significantly improve the computational performance of sampfNMPC.

Finally, it must be noted that a software package for computing NLP sensitivities, in a similar vein as sIPOPT [109], is currently in the works for `Julia`. The package is also meant to provide functionality for computing sensitivity updates using path-following algorithms.

Bibliography

[1] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2017.

[2] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control* (A. Garulli and A. Tesi, eds.), (London), pp. 207–226, Springer London, 1999.

[3] S. Lucia, *Robust Multi-stage Nonlinear Model Predictive Control*. PhD thesis, Technical University of Dortmund, 2014.

[4] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*. Springer Publishing Company, Incorporated, 2nd ed., 2011.

[5] S. Lucia, T. Finkler, D. Basak, and S. Engell, "A new Robust NMPC Scheme and its Application to a Semi-batch Reactor Example," *IFAC Proceedings Volumes*, vol. 45, no. 15, pp. 69 – 74, 2012. 8th IFAC Symposium on Advanced Control of Chemical Processes.

[6] S. Lucia, T. Finkler, and S. Engell, "Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty," *Journal of Process Control*, vol. 23, no. 9, pp. 1306 – 1319, 2013.

[7] S. Lucia, J. A. Andersson, H. Brandt, M. Diehl, and S. Engell, "Handling uncertainty in economic nonlinear model predictive control: A comparative case study,"

Journal of Process Control, vol. 24, no. 8, pp. 1247 – 1259, 2014. Economic nonlinear model predictive control.

[8] H. Jang, J. H. Lee, and L. T. Biegler, "A robust NMPC scheme for semi-batch polymerization reactors," *IFAC-PapersOnLine*, vol. 49, no. 7, pp. 37 – 42, 2016.

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

[10] R. Martí, S. Lucia, D. Sarabia, R. Paulen, S. Engell, and C. de Prada, "Improving Scenario Decomposition Algorithms for Robust Nonlinear Model Predictive Control," *Computers & Chemical Engineering*, vol. 79, pp. 30–45, 2015.

[11] D. Krishnamoorthy, B. Foss, and S. Skogestad, "Real time optimization under uncertainty - applied to gas lifted wells," *Processes*, vol. 4, no. 4, 2016.

[12] D. Hakerl, C. Lindscheid, S. Subramanian, P. Diewald, A. Tatulea-Codrean, and S. Engell, "Economics optimizing control of a multi-product reactive distillation process under model uncertainty," *Computers & Chemical Engineering*, vol. 118, pp. 25 – 48, 2018.

[13] J. Puschke and A. Mitsos, "Robust feasible control based on multi-stage eNMPC considering worst-case scenarios," *Journal of Process Control*, vol. 69, pp. 8 – 15, 2018.

[14] R. Findeisen and F. Allgöwer, "Computational delay in nonlinear model predictive control," *IFAC Proceedings Volumes*, vol. 37, no. 1, pp. 427 – 432, 2004.

[15] D. L. Ma and R. D. Braatz, "Worst-case analysis of finite-time control policies," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 5, pp. 766–774, 2001.

[16] Z. K. Nagy and R. D. Braatz, "Robust nonlinear model predictive control of batch processes," *AIChE Journal*, vol. 49, no. 7, pp. 1776–1786, 2003.

[17] M. Diehl, H. G. Bock, and E. A. Kostina, "An approximation technique for robust nonlinear optimization," *Math. Program.*, vol. 107, no. 1-2, pp. 213–230, 2006.

[18] Z. Nagy and R. Braatz, “Distributional uncertainty analysis using power series and polynomial chaos expansions,” *Journal of Process Control*, vol. 17, no. 3, pp. 229 – 240, 2007. Special Issue ADCHEM 2006 Symposium.

[19] A. Mesbah, S. Streif, R. Findeisen, and R. D. Braatz, “Stochastic nonlinear model predictive control with probabilistic constraints,” in *American Control Conference, ACC 2014, Portland, OR, USA, June 4-6, 2014*, pp. 2413–2419, IEEE, 2014.

[20] E. Bradford and L. Imsland, “Output feedback stochastic nonlinear model predictive control for batch processes,” *Computers & Chemical Engineering*, vol. 126, pp. 434 – 450, 2019.

[21] S. Lucia and R. Paulen, “Robust Nonlinear Model Predictive Control with Reduction of Uncertainty via Robust Optimal Experiment Design,” *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 1904 – 1909, 2014. 19th IFAC World Congress.

[22] D. Krishnamoorthy, S. Skogestad, and J. Jäschke, “Multistage model predictive control with online scenario tree update using recursive bayesian weighting,” in *2019 18th European Control Conference (ECC)*, pp. 1443–1448, 2019.

[23] D. Krishnamoorthy, B. Foss, and S. Skogestad, “A Primal decomposition algorithm for distributed multistage scenario model predictive control,” *Journal of Process Control*, vol. 81, pp. 162 – 171, 2019.

[24] S. Lucia, S. Subramanian, and S. Engell, “Non-conservative Robust Nonlinear Model Predictive Control via Scenario Decomposition,” in *Control Applications (CCA), 2013 IEEE International Conference on*, pp. 586–591, IEEE, 2013.

[25] C. Leidereiter, A. Potschka, and H. G. Bock, “Dual decomposition for QPs in scenario tree NMPC,” in *2015 European Control Conference (ECC)*, pp. 1608–1613, July 2015.

[26] W. Daosud, P. Kittisupakorn, M. Fikar, S. Lucia, and R. Paulen, “Efficient robust nonlinear model predictive control via approximate multi-stage programming: A neural networks based approach,” *Computer Aided Chemical Engineering*, vol. 46, pp. 1261 – 1266, 2019.

[27] F. Holtorf, A. Mitsos, and L. Biegler, “Multistage NMPC with on-line generated scenario trees: Application to a semi-batch polymerization process,” *Journal of Process Control*, vol. 80, pp. 167–179, 2019.

[28] Z. J. Yu and L. T. Biegler, “Advanced-step multistage nonlinear model predictive control: Robustness and stability,” *Journal of Process Control*, vol. 84, pp. 192 – 206, 2019.

[29] M. Thombre, D. Krishnamoorthy, and J. Jäschke, “Data-driven online adaptation of the scenario-tree in multistage model predictive control,” *IFAC-PapersOnLine*, vol. 52, no. 1, pp. 461 – 467, 2019.

[30] M. Thombre, Z. Mdoe, and J. Jäschke, “Data-Driven Robust Optimal Operation of Thermal Energy Storage in Industrial Clusters,” *Processes*, vol. 8, p. 194, Feb 2020.

[31] M. Thombre, S. Prakash, B. R. Knudsen, and J. Jäschke, “Optimizing the capacity of thermal energy storage in industrial clusters,” *Computer Aided Chemical Engineering*, vol. 48, pp. 1459 – 1464, 2020.

[32] M. Thombre, Z. J. Yu, J. Jäschke, and L. T. Biegler, “Sensitivity-Assisted Multistage Nonlinear Model Predictive Control: Robustness, Stability and Computational Efficiency.” Submitted to *Computers & Chemical Engineering*.

[33] M. Thombre, L. T. Biegler, and J. Jäschke, “Sensitivity-Assisted Multistage Nonlinear Model Predictive Control with Path-following.” In preparation.

[34] U. M. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. USA: Society for Industrial and Applied Mathematics, 1st ed., 1998.

[35] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko, *The mathematical theory of optimal processes*. New York, NY: Wiley, 1962.

[36] H. Bock and K. Plitt, “A multiple shooting algorithm for direct solution of optimal control problems,” *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 1603 – 1608, 1984.

[37] J. E. Cuthrell and L. T. Biegler, “On the optimization of differential-algebraic process systems,” *AIChE Journal*, vol. 33, no. 8, pp. 1257–1270, 1987.

[38] L. T. Biegler, *Nonlinear programming: concepts, algorithms, and applications to chemical processes*, vol. 10. SIAM, 2010.

[39] G. de Nicolao, L. Magni, and R. Scattolini, "On the robustness of receding-horizon control with terminal constraints," *IEEE Transactions on Automatic Control*, vol. 41, pp. 451–453, March 1996.

[40] S. Yu, M. Reble, H. Chen, and F. Allgöwer, "Inherent Robustness Properties of Quasi-infinite Horizon Nonlinear Model Predictive Control," *Automatica*, vol. 50, no. 9, pp. 2269 – 2280, 2014.

[41] D. A. Allan, C. N. Bates, M. J. Risbeck, and J. B. Rawlings, "On the inherent robustness of optimal and suboptimal nonlinear mpc," *Systems & Control Letters*, vol. 106, pp. 68 – 78, 2017.

[42] D. Mayne, M. Seron, and S. Rakovi, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219 – 224, 2005.

[43] D. Q. Mayne and E. C. Kerrigan, "Tube-based robust nonlinear model predicitive control," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 36 – 41, 2007. 7th IFAC Symposium on Nonlinear Control Systems.

[44] P. J. Campo and M. Morari, "Robust model predictive control," in *1987 American Control Conference*, pp. 1021–1026, June 1987.

[45] P. Scokaert and D. Q. Mayne, "Min-max Feedback Model Predictive Control for Constrained Linear Systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 8, pp. 1136–1142, 1998.

[46] M. Guay, V. Adetola, and D. DeHaan, *Robust and Adaptive Model Predictive Control of Nonlinear Systems*. Institution of Engineering and Technology, 2015.

[47] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967 – 2986, 2014.

[48] D. Q. Mayne, "Robust and stochastic MPC: Are we going in the right direction?," *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 1–8, 2015.

[49] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York, NY, USA: Springer, second ed., 2006.

[50] J. Gauvin, “A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming,” *Math. Program.*, vol. 12, p. 136138, December 1977.

[51] S. M. Robinson, “Strongly Regular Generalized Equations,” *Mathematics of Operations Research*, vol. 5, no. 1, pp. 43–62, 1980.

[52] M. Kojima, “Strongly stable stationary solutions in nonlinear programs,” in *Analysis and Computation of Fixed Points* (S. M. Robinson, ed.), pp. 93 – 138, Academic Press, 1980.

[53] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[54] D. Krishnamoorthy, M. Thombre, S. Skogestad, and J. Jäschke, “Data-driven scenario selection for multistage robust model predictive control,” *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 462 – 468, 2018.

[55] W. Wiesemann, D. Kuhn, and M. Sim, “Distributionally robust convex optimization,” *Operations Research*, vol. 62, no. 6, pp. 1358–1376, 2014.

[56] B. P. G. V. Parys, D. Kuhn, P. J. Goulart, and M. Morari, “Distributionally robust control of constrained stochastic systems,” *IEEE Transactions on Automatic Control*, vol. 61, pp. 430–442, Feb 2016.

[57] P. R. Maurath, A. J. Laub, D. E. Seborg, and D. A. Mellichamp, “Predictive controller design by principal components analysis,” *Industrial & engineering chemistry research*, vol. 27, no. 7, pp. 1204–1212, 1988.

[58] F. Loquasto and D. E. Seborg, “Model predictive controller monitoring based on pattern classification and pca,” in *American Control Conference, 2003. Proceedings of the 2003*, vol. 3, pp. 1968–1973, IEEE, 2003.

[59] A. AlGhazzawi and B. Lennox, “Model predictive control monitoring using multivariate statistics,” *Journal of Process Control*, vol. 19, no. 2, pp. 314–327, 2009.

[60] I. T. Jolliffe and J. Cadima, “Principal component analysis: a review and recent developments,” *Philosophical transactions Series A, Mathematical, physical, and engineering sciences*, 2016.

[61] L. Grüne and M. Stieler, “Asymptotic stability and transient optimality of economic mpc without terminal conditions,” *Journal of Process Control*, vol. 24, no. 8, pp. 1187 – 1196, 2014. Economic nonlinear model predictive control.

[62] I. Dunning, J. Huchette, and M. Lubin, “Jump: A modeling language for mathematical optimization,” *SIAM Review*, vol. 59, no. 2, pp. 295–320, 2017.

[63] J. Bezanson, A. Edelman, S. Karpinski, and V. Shah, “Julia: A fresh approach to numerical computing,” *SIAM Review*, vol. 59, no. 1, pp. 65–98, 2017.

[64] S. Brückner, L. Miró, L. F. Cabeza, M. Pehnt, and E. Lävemann, “Methods to estimate the industrial waste heat potential of regions a categorization and literature review,” *Renewable and Sustainable Energy Reviews*, vol. 38, pp. 164 – 171, 2014.

[65] H. Fang, J. Xia, and Y. Jiang, “Key issues and solutions in a district heating system using low-grade industrial waste heat,” *Energy*, vol. 86, pp. 589 – 602, 2015.

[66] A. Anastasovski, P. Rakovi, and Z. Guzovi, “Design and analysis of heat recovery system in bioprocess plant,” *Energy Conversion and Management*, vol. 104, pp. 32 – 43, 2015. Special Issue on Sustainable development of energy, water and environment systems.

[67] S. Brückner, S. Liu, L. Miró, M. Radspieler, L. F. Cabeza, and E. Lävemann, “Industrial waste heat recovery technologies: An economic analysis of heat transformation technologies,” *Applied Energy*, vol. 151, pp. 157 – 167, 2015.

[68] V. de Oliveira, J. Jäschke, and S. Skogestad, “Optimal operation of energy storage in buildings: Use of the hot water system,” *Journal of Energy Storage*, vol. 5, pp. 102 – 112, 2016.

[69] Y. Ma, A. Kelman, A. Daly, and F. Borrelli, “Predictive control for energy efficient buildings with thermal storage: Modeling, stimulation, and experiments,” *IEEE Control Systems Magazine*, vol. 32, pp. 44–64, Feb 2012.

[70] R. Tang and S. Wang, "Model predictive control for thermal energy storage and thermal comfort optimization of building demand response in smart grids," *Applied Energy*, vol. 242, pp. 873 – 882, 2019.

[71] M. Labidi, J. Eynard, O. Faugeroux, and S. Grieu, "Predictive control and optimal design of thermal storage systems for multi-energy district boilers," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 10305 – 10310, 2014. 19th IFAC World Congress.

[72] K. M. Powell and T. F. Edgar, "Control of a large scale solar thermal energy storage system," in *Proceedings of the 2011 American Control Conference*, pp. 1530–1535, June 2011.

[73] B. R. Knudsen, H. Kauko, and T. Andresen, "An optimal-control scheme for coordinated surplus-heat exchange in industry clusters," *Energies*, vol. 12, no. 10, 2019.

[74] L. Miró, J. Gasia, and L. F. Cabeza, "Thermal energy storage (tes) for industrial waste heat (iwh) recovery: A review," *Applied Energy*, vol. 179, pp. 284 – 301, 2016.

[75] I. Sarbu and C. Sebarchievici, "A comprehensive review of thermal energy storage," *Sustainability*, vol. 10, no. 1, 2018.

[76] W. Cole, K. Powell, and T. Edgar, "Optimization and advanced control of thermal energy storage systems," *Reviews in Chemical Engineering*, vol. 28, pp. 81–99, 7 2012.

[77] F. Huang, J. Zheng, J. Baleynaud, and J. Lu, "Heat recovery potentials and technologies in industrial zones," *Journal of the Energy Institute*, vol. 90, no. 6, pp. 951 – 961, 2017.

[78] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted l1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, pp. 877–905, Dec 2008.

[79] STFC Rutherford Appleton Laboratory, "HSL. A collection of Fortran codes for large scale scientific computation." `http://www.hsl.rl.ac.uk/`, 2019. Accessed: June 06, 2020.

[80] L. Biegler, X. Yang, and G. Fischer, “Advances in sensitivity-based nonlinear model predictive control and dynamic real-time optimization,” *Journal of Process Control*, vol. 30, pp. 104 – 116, 2015. CAB/DYCOPS 2013.

[81] I. J. Wolf and W. Marquardt, “Fast NMPC schemes for regulatory and economic NMPC – A review,” *Journal of Process Control*, vol. 44, pp. 162 – 183, 2016.

[82] D. W. Griffith, L. T. Biegler, and S. C. Patwardhan, “Robustly stable adaptive horizon nonlinear model predictive control,” *Journal of Process Control*, vol. 70, pp. 109 – 122, 2018.

[83] V. M. Zavala and L. T. Biegler, “The advanced-step NMPC controller: Optimality, stability and robustness,” *Automatica*, vol. 45, no. 1, pp. 86 – 93, 2009.

[84] Z. Yu, *Advances in Decision-making Under Uncertainty with Nonlinear Model Predictive Control*. PhD thesis, Carnegie Mellon University, Jan 2020.

[85] Z. J. Yu and L. T. Biegler, “Sensitivity-assisted Robust Nonlinear Model Predictive Control with Scenario Generation,” *IFAC-PapersOnLine, to appear*, 2020. presented at 1st Virtual IFAC World Congress, Berlin.

[86] A. V. Fiacco, “Sensitivity analysis for nonlinear programming using penalty methods,” *Mathematical Programming*, vol. 10, pp. 287 – 311, 1976.

[87] A. V. Fiacco, *Introduction to sensitivity and stability analysis in nonlinear programming*. Mathematics in Science and Engineering, Burlington, MA: Elsevier, 1983.

[88] B. Srinivasan, D. Bonvin, E. Visser, and S. Palanki, “Dynamic optimization of batch processes: Ii. role of measurements in handling uncertainty,” *Computers & chemical engineering*, vol. 27, no. 1, pp. 27–44, 2003.

[89] L. T. Biegler, I. E. Grossmann, and A. W. Westerberg, *Systematic methods for chemical process design*, ch. Process Flexibility. Prentice Hall PTR, 12 1997.

[90] K.-U. Klatt and S. Engell, “Gain-scheduling trajectory control of a continuous stirred tank reactor,” *Computers & Chemical Engineering*, vol. 22, no. 4, pp. 491 – 502, 1998.

[91] J. Kang, Y. Cao, D. P. Word, and C. Laird, "An interior-point method for efficient solution of block-structured nlp problems using an implicit schur-complement decomposition," *Computers & Chemical Engineering*, vol. 71, pp. 563 – 573, 2014.

[92] T. Raff, S. Huber, Z. K. Nagy, and F. Allgower, "Nonlinear model predictive control of a four tank system: An experimental stability study," in *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, pp. 237–242, 2006.

[93] S. Lucia, S. Subramanian, D. Limon, and S. Engell, "Stability properties of multi-stage nonlinear model predictive control," *Systems & Control Letters*, vol. 143, p. 104743, 2020.

[94] M. Maiworm, T. Bäthge, and R. Findeisen, "Scenario-based Model Predictive Control: Recursive Feasibility and Stability," *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 50–56, 2015.

[95] D. M. Raimondo, D. Limon, M. Lazar, L. Magni, and E. F. Camacho, "Min-max Model Predictive Control of Nonlinear Systems: A Unifying Overview on Stability," *European Journal of Control*, vol. 15, no. 1, pp. 5–21, 2009.

[96] J. Jäschke, X. Yang, and L. T. Biegler, "Fast Economic Model Predictive Control Based on NLP-Sensitivities," *Journal of Process Control*, vol. 24, pp. 1260–1272, 2014.

[97] G. Pannocchia, J. Rawlings, and S. Wright, "Conditions under which supboptimal nonlinear MPC is inherently robust," *Systems & Control Letters*, vol. 60, pp. 747–755, 2011.

[98] J. Jäschke, X. Yang, and L. T. Biegler, "Fast economic model predictive control based on NLP-sensitivities," *Journal of Process Control*, vol. 24, no. 8, pp. 1260 – 1272, 2014. Economic nonlinear model predictive control.

[99] E. Suwartadi, V. Kungurtsev, and J. Jäschke, "Sensitivity-based economic nmpc with a path-following approach," *Processes*, vol. 5, p. 8, Feb 2017.

[100] D. Krishnamoorthy, E. Suwartadi, B. Foss, S. Skogestad, and J. Jäschke, "Improving scenario decomposition for multistage mpc using a sensitivity-based path-following algorithm," *IEEE Control Systems Letters*, vol. 2, no. 4, 2018.

[101] V. Kungurtsev and M. Diehl, "Sequential quadratic programming methods for parametric nonlinear optimization," *Comput. Optim. Appl.*, vol. 59, no. 3, pp. 475–509, 2014.

[102] V. Kungurtsev and J. Jäschke, "A predictor-corrector path-following algorithm for dual-degenerate parametric optimization problems," *SIAM J. Optim.*, vol. 27, no. 1, pp. 538–564, 2017.

[103] E. Suwartadi, V. Kungurtsev, and J. Jäschke, "Fast sensitivity-based economic model predictive control for degenerate systems," *Journal of Process Control*, vol. 88, pp. 54 – 62, 2020.

[104] A. B. Levy, "Solution Sensitivity from General Principles," *SIAM Journal on Control and Optimization*, vol. 40, no. 1, pp. 1–38, 2001.

[105] J. F. Bonnans and A. Shapiro, "Optimization problems with perturbations: A guided tour," *SIAM Rev.*, vol. 40, p. 228264, June 1998.

[106] COIN-OR. `https://coin-or.github.io/Ipopt/`, 2020. Accessed: December 15, 2020.

[107] S. Yue, P. Li, and P.-Y. Hao, "Svm classification:its contents and challenges," *Applied Mathematics-A Journal of Chinese Universities*, vol. 18, pp. 332–342, 2003.

[108] S. Subramanian, S. Lucia, and S. Engell, "A synergistic approach to robust output feedback control: Tube-based multi-stage nmpc," in *10th IFAC International Symposium on Advanced Control of Chemical Processes*, pp. 494–499, IFAC, 2018.

[109] H. Pirnay, R. López-Negrete, and L. T. Biegler, "Optimal sensitivity based on ipopt," *Mathematical Programming Computation*, vol. 4, pp. 307–331, 2012.

[110] L. Eriksson, T. Byrne, E. Johansson, J. Trygg, and C. Vikström, *Multi-and megavariate data analysis basic principles and applications*, vol. 1. Umetrics Academy, 2013.

[111] F. Harrou, F. Kadri, S. Chaabane, C. Tahon, and Y. Sun, “Improved principal component analysis for anomaly detection: Application to an emergency department,” *Computers & Industrial Engineering*, vol. 88, pp. 63–77, 2015.

[112] H. Hotelling, “Analysis of a complex of statistical variables into principal components.,” *Journal of Educational Psychology*, vol. 24, no. 6, p. 417, 1933.

Appendix A

Supporting information: Chapter 3

Simple TES model

Figure A.1 shows the schematic with the different states, inputs and disturbances in the TES model. The heat exchangers are modeled as devices with two chambers representing the hot side and the cold side. Both chambers of the heat exchanger, as well as the TES tank itself, are considered to have the same temperature throughout their volumes. Thus the temperatures exiting these volumes are considered to be same as those inside the volumes. The dynamics of the process involve energy balances over the hot and cold sides of the heat exchangers, and over the TES tank.

The TES tank has a volume V^{TES}, and experiences heat loss with a conductance $(UA)^{TES}$. Each side of the heat exchangers has a volume V^{hex} and heat transfer conduc-

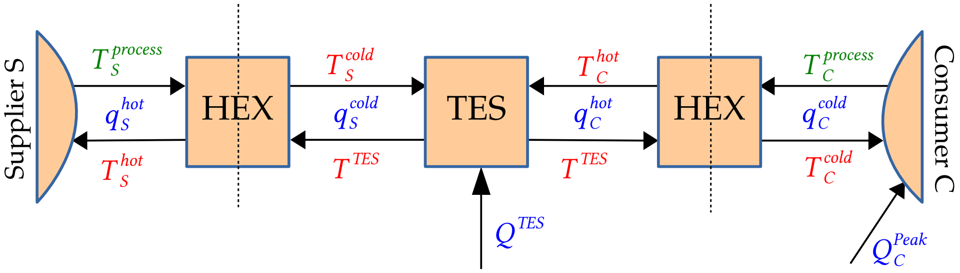


Figure A.1: Schematic of the model. The states, inputs and uncertainties are shown in red,blue, and green respectively.

tance $(UA)^{hex}$. The heat exchanger fluid is water, with density and specific heat capacity denoted by ρ and C_p respectively.

The energy balances on the supplier side can then be formulated as:

$$\rho C_p V^{hex} \frac{dT_S^{hot}}{dt} = \rho C_p q_S^{hot} (T_S^{process} - T_S^{hot}) - (UA)^{hex} (T_S^{hot} - T_S^{cold}) \tag{A.1a}$$

$$\rho C_p V^{hex} \frac{dT_S^{cold}}{dt} = \rho C_p q_S^{cold} (T^{TES} - T_S^{cold}) + (UA)^{hex} (T_S^{hot} - T_S^{cold}) \tag{A.1b}$$

Similarly on the consumer side:

$$\rho C_p V^{hex} \frac{dT_C^{hot}}{dt} = \rho C_p q_C^{hot} (T^{TES} - T_C^{hot}) - (UA)^{hex} (T_C^{hot} - T_C^{cold}) \tag{A.2a}$$

$$\rho C_p V^{hex} \frac{dT_C^{cold}}{dt} = \rho C_p q_C^{cold} (T_C^{process} - T_C^{cold}) + (UA)^{hex} (T_C^{hot} - T_C^{cold}) \tag{A.2b}$$

The energy balance over the TES tank yields:

$$\begin{aligned} \rho C_p V^{TES} \frac{dT^{TES}}{dt} &= \rho C_p q_S^{cold} (T_S^{cold} - T^{TES}) + \rho C_p q_C^{hot} (T_C^{hot} - T^{TES}) \\ &\quad - (UA)^{TES} (T^{TES} - T^{amb}) + Q^{TES} \end{aligned} \tag{A.3}$$

Finally, the demand satisfaction constraint on the consumer side can be written as:

$$\rho C_p q_C^{cold} (T_C^{cold} - T_C^{process}) + Q_C^{peak} \geq Q_C^{demand} \tag{A.4}$$

For the multistage NMPC formulation (2.4), we consider an economic objective function related to minimizing the cost of energy purchase for the TES system:

$$\psi_l = P^{TES} (Q^{TES})_l + P^{peak} (Q_C^{peak})_l \tag{A.5}$$

The model parameters are shown in Table A.1, and the bounds on states and inputs are given in Table A.2.

Table A.1: Simple TES - Model parameters.

Parameter	Value	Unit
ρ	1000	kg/m^3
C_p	4.18	$kJ/kg.K$
V^{TES}	100	m^3
V^{hex}	0.5	m^3
$(UA)^{TES}$	50	kW/K
$(UA)^{hex}$	150	kW/K
T^{amb}	20	$^{\circ}C$
Q_C^{demand} (normal)	5	MW
Q_C^{demand} (peak)	10	MW
P^{TES}	5	price units
P^{peak}	1000	price units

Table A.2: Simple TES - Bounds on states and inputs.

Variable	Minimum	Maximum	Unit
T_S^{hot}	0	100	$^{\circ}C$
T_S^{cold}	0	100	$^{\circ}C$
T_C^{hot}	0	100	$^{\circ}C$
T_C^{cold}	0	100	$^{\circ}C$
T^{TES}	30	100	$^{\circ}C$
q_S^{hot}	0	1	m^3/s
q_S^{cold}	0	1	m^3/s
q_C^{hot}	0	1	m^3/s
q_C^{cold}	0	1	m^3/s
Q^{TES}	0	5	MW
Q_C^{peak}	0	-	MW

Appendix B

Supporting information: Chapter 4

Detailed TES model

A full schematic of the process used for model derivation is shown in Figure B.1.

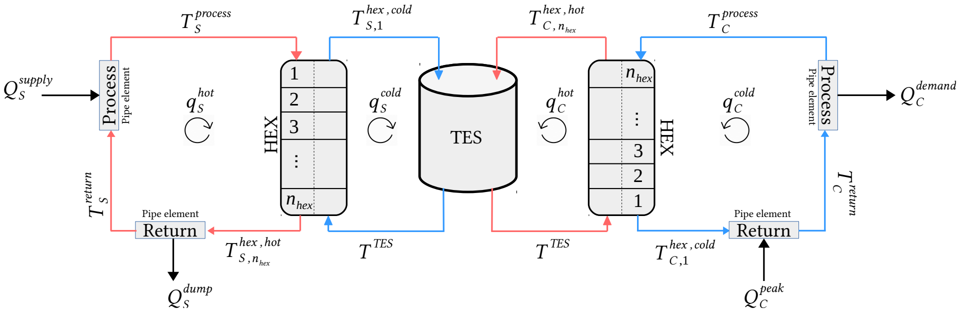


Figure B.1: Detailed model illustration for the heat exchangers in the TES. The red and blue lines represent the hot and cold streams respectively.

The TES tank has a volume V^{TES}, and experiences heat loss with a conductance $(UA)^{TES}$. Each cell of the heat exchangers has a volume V_{part}^{hex} and heat transfer conductance $(UA)^{hex}$. The heat exchanger fluid is water, with density and specific heat capacity denoted by ρ and C_p respectively.

We apply enthalpy balances over the components shown in Figure B.1, namely the process and return pipe elements, as well as the hot and cold sides of the heat exchanger. Note that we consider countercurrent flow in the heat exchangers.

The enthalpy balance over the process pipe element yields:

$$\rho C_p V_{part}^{hex} \frac{dT_S^{process}}{dt} = Q_S^{supply} - \rho C_p q_S^{hot}(T_S^{process} - T_S^{return}) \tag{B.1}$$

Similarly, the enthalpy balance over the return pipe element gives:

$$\rho C_p V_{part}^{hex} \frac{dT_S^{return}}{dt} = \rho C_p q_S^{hot}(T_{S,n_{hex}}^{hex,hot} - T_S^{return}) - Q_S^{dump} \tag{B.2}$$

The balances over the hot and cold sides of the supplier-side heat exchanger lead to the following equations:

$$\rho C_p V_{part}^{hex} \frac{dT_{S,k}^{hot}}{dt} = \rho C_p q_S^{hot}(T_{S,k-1}^{hot} - T_{S,k}^{hot}) - UA^{hex}(T_{S,k}^{hot} - T_{S,k}^{cold}), \tag{B.3a}$$

$$T_{S,0}^{hot} = T_S^{process}, \ \forall k \in \{1, 2, \ldots, n_{hex}\} \tag{B.3b}$$

$$\rho C_p V_{part}^{hex} \frac{dT_{S,k}^{cold}}{dt} = \rho C_p q_S^{cold}(T_{S,k+1}^{cold} - T_{S,k}^{cold}) + UA^{hex}(T_{S,k}^{hot} - T_{S,k}^{cold}), \tag{B.4a}$$

$$T_{S,n_{hex}+1}^{cold} = T^{TES}, \ \forall k \in \{1, 2, \ldots, n_{hex}\} \tag{B.4b}$$

The enthalpy balances for the consumer side are analogous to the supplier side. The enthalpy balance for the process pipe element leads to:

$$\rho C_p V_{part}^{hex} \frac{dT_C^{process}}{dt} = \rho C_p q_C^{hot}(T_C^{return} - T_C^{process}) - Q_C^{demand} \tag{B.5}$$

Similarly, the balance over the consumer-side return pipe element yields:

$$\rho C_p V_{part}^{hex} \frac{dT_C^{return}}{dt} = Q_C^{peak} - \rho C_p q_C^{hot}(T_C^{return} - T_{C,1}^{hex,cold}) \tag{B.6}$$

Finally, the enthalpy balances over the hot and cold sides on the consumer-side heat exchanger result in the following equations:

$$\rho C_p V_{part}^{hex} \frac{dT_{C,k}^{hot}}{dt} = \rho C_p q_C^{hot}(T_{C,k-1}^{hot} - T_{C,k}^{hot}) - UA^{hex}(T_{C,k}^{hot} - T_{C,k}^{cold}), \tag{B.7a}$$

$$T_{C,0}^{hot} = T^{TES}, \ \forall k \in \{1, 2, \ldots, n_{hex}\} \tag{B.7b}$$

$$\rho C_p V_{part}^{hex} \frac{dT_{C,k}^{cold}}{dt} = \rho C_p q_C^{cold}(T_{C,k+1}^{cold} - T_{C,k}^{cold}) + UA^{hex}(T_{C,k}^{hot} - T_{C,k}^{cold}), \tag{B.8a}$$

$$T_{C,n_{hex}+1}^{cold} = T_C^{process}, \ \forall k \in \{1, 2, \ldots, n_{hex}\} \tag{B.8b}$$

$Q_S^{TES,in}$ is the heat supplied to the TES, and can be calculated as the heat transfer from the hot to cold side of the supplier-side heat exchanger (since heat loss in the pipes is negligible).

$$Q_S^{TES,in} = \sum_{k=1}^{n_{hex}} (UA)_k^{hex} (T_{S,k}^{hex,hot} - T_{S,k}^{hex,cold}) \tag{B.9}$$

where $(UA)^{hex}$ is the heat-transfer conductance of the heat exchangers. Similarly, $Q_C^{TES,out}$ is the heat extracted from the TES.

$$Q_C^{TES,out} = \sum_{k=1}^{n_{hex}} (UA)_k^{hex} (T_{C,k}^{hex,hot} - T_{C,k}^{hex,cold}) \tag{B.10}$$

The resulting TES energy balance is:

$$\rho c_p V^{TES} \frac{dT^{TES}}{dt} = Q_S^{TES,in} - Q_C^{TES,out} - Q^{TES,loss} \tag{B.11}$$

where $Q^{TES,loss}$ is the heat lost by the tank to the surroundings.

$$Q^{TES,loss} = (UA)^{TES} \big(T^{TES} - T^{amb}\big) \tag{B.12}$$

It may be noted from Figure B.1 that the volumetric flow rates of water going in and out of the TES are equal. The volume of the water inside the TES is thus always constant.

For the NMPC formulations such as (2.2) and (2.4), we consider an economic objective cost related to minimizing the use of peak heating sources in the industrial cluster:

$$\psi_l = \left(Q_C^{peak}\right)_l \tag{B.13}$$

The model parameters are shown in Table B.1, and the bounds on states and inputs are given in Table B.2.

Outlier detection using PCA

The data set is limited to winter months of January–March and December when thermal energy storage is applicable. This resulting supply and demand data sets each have 121 sample days with 24 hourly measurements in each day.

PCA is performed separately on the supply and demand data matrices each of dimensions 121 × 24, resulting into two models. Since we aim at finding a linear correlation

Table B.1: Detailed TES - Model parameters.

Parameter	Value	Unit
V^{TES}	1000	m^3/s
$(UA)^{TES}$	0.1	kW/K
V^{hex}	1	m^3
$(UA)^{hex}$	3000	kW/K
n_{hex}	5	-
T^{amb}	-5	$^{\circ}C$
ρ	1000	kg/m^3
C_p	4.18	$kJ/kg.K$
q_S^{hot},q_S^{cold}	0.3	m^3/s

Table B.2: Detailed TES - Bounds on states and inputs.

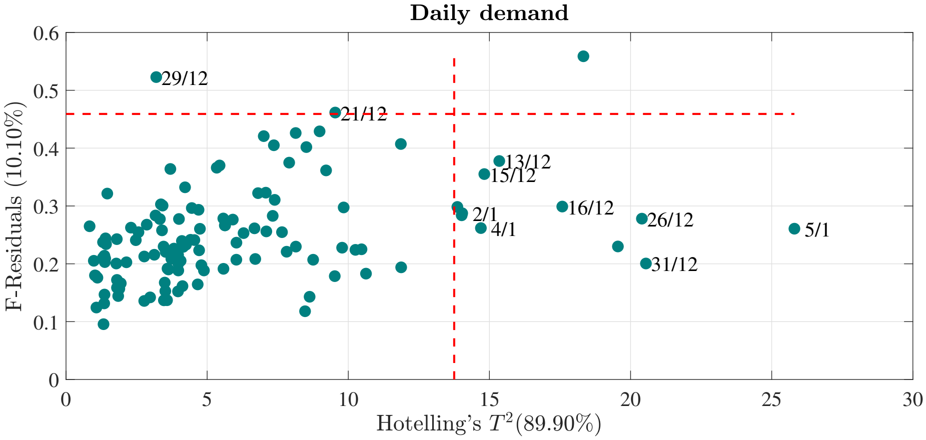
Variable	Minimum	Maximum	Unit
$T_S^{process}$	50	100	$^{\circ}C$
T_S^{return}	50	85	$^{\circ}C$
$T_C^{process}$	50	100	$^{\circ}C$
T_C^{return}	60	100	$^{\circ}C$
T^{TES}	50	100	$^{\circ}C$
q_C^{hot}	0.001	0.3	m^3/s
q_C^{cold}	0.001	0.3	m^3/s
Q_S^{dump}	0	-	kW
Q_C^{peak}	0	-	kW

between supply and demand, we detect outlier samples appearing in either of the two PCA models. The supply data PCA model covered a total variance of about 89.90% using 7 principal components. On the other hand, demand data PCA model captured more variation with fewer components and achieved a total explained variance of 95.58% using 6 principal components.

Determination of an outlier is done by calculation of sample residuals and leverage in each PCA model. Leverage is the distance of the sample point from the origin of the PCA model measured along the new model subspace. A sample with high leverage implies over-influence on the PCA model, making it not a general representation of the system. Samples with significantly higher residuals and leverage compared to the others are considered outliers. This is a statistical anomaly detection which can be applied to processes as explained in [110, 111].

The sample residuals are estimated to be F-distributed and the X-residuals are converted to F-residuals. A multivariate t-test on the PCA scores (not the original variables) is called the Hotelling's T^2 statistic [112]. The Hotelling's T^2 statistic is directly related to the leverage of the sample on the PCA model. It is also approximated as being F-distributed. The critical limits for Hotelling's T^2 and F-residuals are estimated such that the confidence interval is 95%. A plot of F-residual against Hotelling's T^2 shown in Figure B.2, called the influence plot, represents the outlier analysis for supply and demand data sets.

The samples that are above the critical limits (red-dashed lines) in either F-residual and Hotelling's T^2 statistic are considered outliers and are discarded from further analysis. In the Figure B.2, the outliers corresponding to the months of January and December are numbered with their corresponding date.



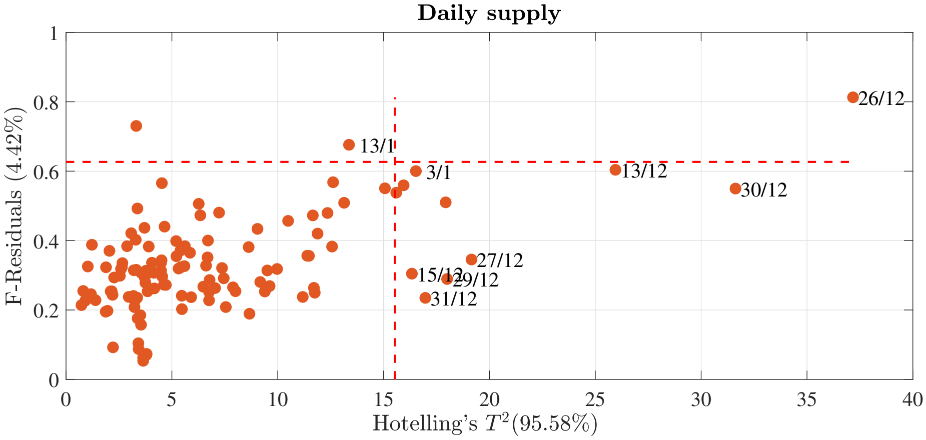


Figure B.2: Influence plots for demand and supply data in the winter months (January–March and December) of 2017. The confidence interval is set to 95% and the limits are shown as red-dashed lines. Samples with either high Hotelling's T^2 or F-Residuals (F-distributed) are considered outliers. The outliers are labelled with their corresponding date.

Appendix C

Supporting information: Chapter 5

CSTR model

Table C.1: CSTR - Model parameters.

Parameter	Value	Unit
$k_{0,1}$	1.287×10^{12}	$1/h$
$k_{0,2}$	1.287×10^{12}	$1/h$
$k_{0,3}$	9.043×10^{9}	$L/mol.h$
$E_{A,1}/R$	9758.3	K
$E_{A,2}/R$	9758.3	K
$E_{A,3}/R$	8560.0	K
ΔH_{AB}	4.2	kJ/mol
ΔH_{BC}	-11.0	kJ/mol
ΔH_{AD}	-41.85	kJ/mol
c_p	3.01	$kJ/kg.K$
c_{pK}	2.0	$kJ/kg.K$
ρ	0.9342	kg/L
A	0.215	m^2
V_R	10.0	L
T_{in}	130	kg
k_W	4032	$kJ/h.m^2.K$
m_K	5.0	kg
c_{A0}	5.1	mol/L

Table C.2: CSTR - Initial conditions and 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	5.0	mol/L
T_R	134.14	50	140	$^\circ C$
T_K	134.0	50	180	$^\circ C$
F	18.83	5	100	$1/h$
$\dot{Q}_K$	-4495.7	-8500	0	kJ/h

Quadtank model

Table C.3: Quadtank - 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	$-$	γ_2	0.4	$-$

Table C.4: Quadtank - 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	mol/L
u_2	0.0	60.0	mol/L

Table C.5: Predefined pulse changes to state variables in Quadtank case study.

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

Appendix D

Optimal Sizing in a Thermal Energy Storage System

In this chapter, we investigate design optimization formulations for finding the optimum size of the storage tank and heat exchangers, that offer robustness against heat supply and demand uncertainty. Two stochastic programming formulations, one based on a single-level and another based on a bilevel approach are shown. For the bilevel formulation, the design decisions are the upper level variables and the operating decision constitute the lower level variables. Since we consider linear formulations, the bilevel problem is reformulated as a mathematical program with complementarity constraints (MPCC) by replacing the lower level problem with its optimality conditions. The MPCC is further cast as a mixed-integer linear program (MILP) using the Big M approach, which is then solved using Gurobi, a specialized MILP solver.

Acknowledgment

We thank Mo Fjernvarme AS for providing industrial data on heat supply and demand used in this chapter.

PROCEEDINGS OF THE 30th European Symposium on Computer Aided Process Engineering
(ESCAPE30), May 24-27, 2020, Milano, Italy

Optimizing the Capacity of Thermal Energy Storage in Industrial Clusters

Mandar Thombre,[a] Sandeep Prakash,[a] Brage Rugstad Knudsen,[b] Johannes Jäschke[a*]

[a] Norwegian University of Science and Technology, Trondheim 7491, Norway
[b]SINTEF Energy Research, Trondheim 7491, Norway
johannes.jaschke@ntnu.no

Abstract

A key factor for energy-efficient industrial clusters is the recovery of waste heat. To this end, thermal energy storage (TES) is an appealing technology that facilitates dynamic heat integration between supplier and consumer plants. A long-term strategy for energy savings must involve adequate consideration for the optimal design of the TES. From an industrial perspective, finding the capacity of the TES unit is often based on heuristic rules which may lead to suboptimal design. This approach does not account for the short-term variability in operation of the TES system. Scenario-based stochastic programming approaches, where the operational uncertainty is described in form of discrete scenarios, can be used to find the best design for the TES system. We present two problem formulations for finding the optimal capacity of the TES unit. The first is a single-level formulation where the design and operating constraints are combined for all scenarios, with the objective of minimizing the combined cost of design and operation. The second is a bilevel formulation where the design decisions are taken on the upper level to minimize overall system cost, whereas the lower level problems (one per scenario) represent the optimal operation for the chosen design variables, each minimizing the operating cost for their respective scenarios. We compare the results of the two approaches with an illustrative case study of an industrial cluster with one supplier plant and one consumer plant exchanging heat via a TES unit.

Keywords: thermal energy storage, bilevel programming, industrial cluster

1. Introduction

Storage and reuse of industrial waste heat is vital for improving energy efficiency of many energy-intensive processes. When multiple industrial plants operate in close proximity of each other, waste heat can be recovered from one plant and supplied to another plant in need of it. Thermal energy storage (TES) can mitigate the issue of asynchronous heat supply and demand by storing energy during off-peak periods and discharging it during peak demands, leading to savings in operating costs. The capital investment costs for installing a TES system are proportional to the capacity of the TES, and may become significantly high. In order to find a trade-off between high capital costs (large capacity) and high operating costs (small capacity), it is worth investigating methods for optimally sizing the TES. A well designed TES system has to contend with operational uncertainty, for example the daily/weekly fluctuations in heat supply and demand. By incorporating this uncertainty information in the design phase itself, it is possible to size a TES system that is robust against this uncertainty. Solving a single deterministic optimization problem that spans across the entire operation horizon of the TES (typically multiple years), and accounts for all the heat profile fluctuations therein, is computationally intractable. To overcome this, stochastic programming approaches can be used to optimize the design decisions over a set of representative scenarios.

Our aim is to find a measure for the optimal sizing of the TES equipment - the volume of a TES unit and the areas of the HEX delivering/extracting heat from the TES unit - by rigorously accounting for the uncertain heat supply and demand in operation phase. For the TES system, the decisions can be divided into two stages - design and operation. In the extensive form of stochastic programming (Birge and Louveaux, 2011), the design variables are "here-and-now", whereas the "wait-and-see" operation variables are assigned to each scenario. This results in a single-level optimization problem, where the objective function represents the overall system cost. The design constraints and the operating constraints for each scenario are all imposed together in this formulation.

Another stochastic approach is the bilevel formulation, based on a Stackelberg leader-follower hypothesis. The upper level problem (leader) identifies the optimal design decisions that minimize the overall cost over a set of scenarios. On the other hand, the lower level problems (followers), representing different scenarios, aim to minimize their corresponding operating cost (see Xu et al. (2017), for example). Bilevel problems are typically nonconvex and NP-hard. However, for cases where the lower level problems are convex and follow some constraint qualifications, the lower level problems can be replaced with their Karush-Kuhn-Tucker (KKT) optimality conditions (Dempe and Franke (2019)). The KKT reformulation turns the bilevel problem into a single-level mathematical program with complementarity constraints (MPEC). The complementarity constraints can be further linearized using disjunctive programming (Fortuny-Amat and McCarl, 1981), rendering the problem a mixed-integer program.

In this paper, we develop a linear model for the TES system and present the two formulations for optimizing the TES design. The results are compared with the help of a case study that is motivated from an industrial district heating network in northern Norway. We compare the results of the two approaches in terms of design parameters for the TES - its volume, the HEX area and the associated capital investment.

2. Methodology

Topology of a TES system with one supplier and one consumer is shown in Figure 1. We employ a simplified linear model in terms of heat duties (MW) to represent the TES system. The heat supplier needs to reject $Q_{supply}(t)$ amount of duty, whereas the consumer has a heat demand $Q_{demand}(t)$ to be met. If the TES cannot meet the total demand of the consumer, the excess energy $Q_{peak}(t)$ is imported from an external peak heating source. Similarly, if all of supplied heat cannot be extracted from the supplier, the excess energy $Q_{dump}(t)$ is rejected into a cooling water system. The resulting heat flows in and out of the tank are denoted by $Q_{tes}^{in}(t)$ and $Q_{tes}^{out}(t)$. The energy in the TES unit is denoted by $E_{tes}(t)$ (MWh). Heat losses from the TES unit to the surroundings are denoted by $Q_{loss}(t)$, which proportional to its energy content. The peak heating and heat dumping duties, along with the energy in the TES unit represent the operating variables in the system. $x_{opr} := \{Q_{peak}(t), Q_{dump}(t), E_{tes}(t)\}$ The associated costs (NOK/MWh) of importing and dumping heat are $C_{peak}(t)$ and $C_{dump}(t)$ respectively. Considering an operating period from t_0 to t_f, the total operating cost can be shown as

$$C_{opr} := \int_{t_0}^{t_f} \left(C_{peak}(t) Q_{peak}(t) + C_{dump}(t) Q_{dump}(t) \right) dt \tag{1}$$

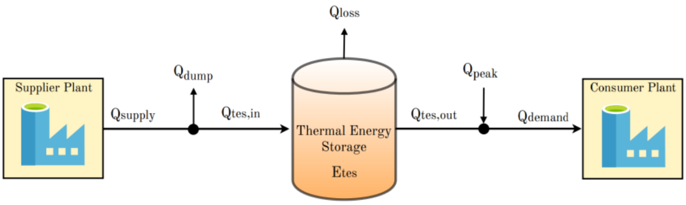


Figure 1: Industrial cluster with one supplier and one consumer exchanging heat through TES.

In context of the system design, the required total energy capacity of the TES unit (MWh) is denoted by CAP_{tes}, whereas the required maximum power rating for heat exchange with the TES unit (MW) is POW_{tes}. For taking design decisions, the former gives the basis for choosing the total volume of the tank. Similarly, the maximum power rating, often serving as the heat exchanger (HEX) design point is related to the HEX area required to deliver and withdraw heat from the TES unit. Thus, CAP_{tes} and POW_{tes} represent the design variables in the system. $x_{des} := \{CAP_{tes}, POW_{tes}\}$. The prices associated with these variables are C_{cap}(NOK/MWh) and C_{pow}(NOK/MW) respectively. The total design cost is: $C_{des} := C_{cap}CAP_{tes} + C_{pow}POW_{tes}$.

Our aim is to identify optimal design parameters for TES system under some information about the operational uncertainty. Uncertainty is modeled in terms of N scenarios, each representing a discrete combination of the heat supply and demand profiles $Q_{supply}(t)$ and $Q_{demand}(t)$ across the operating period. We consider two different formulations of the design optimization problem, a single-level formulation and a bilevel formulation.

2.1. Single-Level formulation

Considering N scenarios of operation, the single-level problem is formulated as (2).

$$\min_{x_{des},x_{opr}} \quad C_{des} + \sum_{n=1}^{N} \omega_n \, C_{opr,n} \tag{2a}$$

$$s.t. \quad CAP_{tes} \geq 0 \tag{2b}$$

$$POW_{tes} \geq 0 \tag{2c}$$

$$\dot{E}_{tes,n}(t) = Q_{tes,n}^{in}(t) - Q_{tes,n}^{out}(t) - Q_{loss,n}(t) \qquad n = 1,\ldots,N \tag{2d}$$

$$0 \leq Q_{peak,n}(t) \leq Q_{peak,max} \qquad n = 1,\ldots,N \tag{2e}$$

$$0 \leq Q_{dump,n}(t) \leq Q_{dump,max} \qquad n = 1,\ldots,N \tag{2f}$$

$$0 \leq Q_{tes,n}^{in}(t) \leq POW_{tes} \qquad n = 1,\ldots,N \tag{2g}$$

$$0 \leq Q_{tes,n}^{out}(t) \leq POW_{tes} \qquad n = 1,\ldots,N \tag{2h}$$

$$0 \leq E_{tes,n}(t) \leq CAP_{tes} \qquad n = 1,\ldots,N \tag{2i}$$

Here, the subscript n represents the nth scenario of operation. In the objective (2a), ω_n is the probability associated with the nth scenario. Equation (2d) is the energy balance equation for the TES, where $\dot{E}_{tes,n}(t)$ is the derivative of the energy in the TES unit. The heat flows in and out of the TES unit are upper bounded by the POW_{tes}, and energy in TES unit is upper bounded by its capacity CAP_{tes}. For implementation, we discretize all the continuous variables in (2) using constant time steps. The integral in the objective (2a) is thus replaced by summation over all the discretized time steps.

Moreover, we employ a forward Euler scheme to discretize the energy balance equation (2d). This transforms (2) into an LP, solvable by MILP solvers like Gurobi and CPEX.

2.2. Bilevel formulation

In the bilevel formulation (3), the lower level operating variables are constrained to be the optimal solutions of the lower level problems (3d), corresponding to their respective scenarios of operation. The upper level objective function is the overall cost (same as (2)), whereas the objective function of each lower level problem is the operating cost for the corresponding scenario.

$$\min_{x_{des}, x_{opr}} \quad C_{des} + \sum_{n=1}^{N} \omega_n \, C_{opr,n} \tag{3a}$$

$$s.t. \quad CAP_{tes} \geq 0 \tag{3b}$$

$$POW_{tes} \geq 0 \tag{3c}$$

$$\begin{aligned} x_{opr,n} \in \arg\min_{x_{opr,n}} \quad & C_{opr,n} \qquad\qquad n = 1, \ldots, N \\ s.t. \quad & \dot{E}_{tes,n}(t) = Q^{in}_{tes,n}(t) - Q^{out}_{tes,n}(t) - Q_{loss,n}(t) \\ & 0 \leq Q_{peak,n}(t) \leq Q_{peak,max} \\ & 0 \leq Q_{dump,n}(t) \leq Q_{dump,max} \\ & 0 \leq Q^{in}_{tes,n}(t) \leq POW_{tes} \\ & 0 \leq Q^{out}_{tes,n}(t) \leq POW_{tes} \\ & 0 \leq E_{tes,n}(t) \leq CAP_{tes} \end{aligned} \tag{3d}$$

The notation used for various variables is the same as in (2). Note that the lower level constraints involve the upper level variables. We also use the same discretization scheme as (2) to convert (3) into a linear bilevel program. This linear bilevel program is still nonconvex is nature owing to the constraints (3d). However, since the lower level problems (3d) are LPs after discretization, the bilevel problem (3) can be reformulated as an MILP as explained in Section 1, and can thus be solved by solvers like Gurobi and CPLEX.

The formulation (3) has tighter feasible set than (2). Problem (3) is thus expected to result in more conservative solutions than (2). However, the bilevel formulation is more representative of the design problem since in practice the problem has a hierarchical nature with the operator's decisions following those of the designer's, with both trying to optimize their respective objectives.

3. Case study – design basis

An industrial TES system with one supplier and one consumer of heat is studied. To formulate the design problem, 5 years of operation is assumed for the TES unit. The overall objective function (Equations (2a) and (3a)) in the formulation then consists the design cost and the operating cost for 5 years of operation. On the operation level, we consider only hourly variation in the heat duties and, to maintain computational tractability, an operating horizon of one week (168 hours). On the design level, we approximate the total 5-year operating cost by extrapolating the weekly operating cost from the operation level over 5 years of operation. The scenarios for weekly operation are taken from the 2017 winter data for heat supply/demand provided by Mo Fjernvarme, a district heating company in northern Norway. Further, all scenarios are considered equally likely in the formulations (2) and (3). The prices for peak heating, C_{peak} are taken to be the corresponding hourly 2017 electricity prices in northern Norway. The prices for heat dumping, C_{dump}, are assumed to be 1/10th of the peak heating prices. The maximum peak heating and heat dumping rates $Q_{peak,max}$ and $Q_{dump,max}$ are set to be 50 MW each. The design basis for calculating the TES volume and HEX area is as follows. The maximum energy storage capacity CAP_{tes} is related to the TES volume and depends on the total enthalpy change of the TES fluid in the tank between the fully charged and fully discharged state, $CAP_{tes} = \rho C_p V_{tes} \Delta T$. Assuming water as the storage medium and an operating window of $20^\circ C$ for the storage tank, the following relation is obtained:

$$V_{tes}\ (\text{m}^3) = 43.06\, CAP_{tes}\ (\text{MWh}) \tag{4}$$

The maximum power rating POW_{tes} corresponds to the maximum duty transferred across the HEX to and from the TES unit, given by $Q = UA(\Delta T)_{LMTD} = mC_p\Delta T$. Charging to a nearly fully charged TES or discharging from a nearly discharged TES unit would give the maximum area requirement of the HEX. We assume that the TES unit is large enough to have a nearly flat profile across the HEX and use a $10^\circ C$ approach temperature in the HEX. Using the fluid properties of water, we estimate the lowest $(\Delta T)_{LMTD}$ to be $19.5^\circ C$, and get the relation between POW_{tes} and an upper bound for the area required for the HEX as:

$$A_{hex}\ (\text{m}^2) = 60.24\, POW_{tes}\ (\text{MW}) \tag{5}$$

Finally, we use a linearized approximation of the total purchased equipment cost as provided by Sinnott and Towler (2009), to estimate our design costs C_{cap} and C_{pow}. Following the factorial method to convert the purchase costs to total design costs, we get the following approximate relations:

$$C_{cap}(\text{mil. NOK 2017}) = 0.7 + 0.11\, CAP_{tes}\ (\text{MWh}) \tag{6}$$

$$C_{pow}(\text{mil. NOK 2017}) = 0.095 + 0.3\, POW_{tes}\ (\text{MW}) \tag{7}$$

4. Results and discussions

We compare the results between the two formulations while considering 5, 10 and 20 weekly scenarios, chosen from the 2017 winter data. Also, when considering real data, care has to be taken to avoid any outliers that may skew the results of the design optimization. Figure 2 shows that the single-level formulation results in higher TES capacities, whereas the bilevel formulation emphasizes higher HEX areas for efficient heat transfer. This implies that, at higher TES capacities, optimal lower level solutions result in a higher design cost for the bilevel formulation. The bilevel formulation prioritizes minimizing the design objective at the expense of operation objective. Although this results in a higher operation cost for the bilevel formulation (Figure 3), it ensures that the chosen design parameters lead to optimal operation on the lower level. Also interesting to note is that the design costs from the bilevel formulation remain unchanged when scenarios are increased from 10 to 20. The single-level formulation leads to lower overall costs, but optimal operation is not guaranteed explicitly for any of the chosen scenarios. Including more scenarios seems to reduce the design cost in single-level formulation. Availability of more data would allow us to check if this cost converges to a particular value.

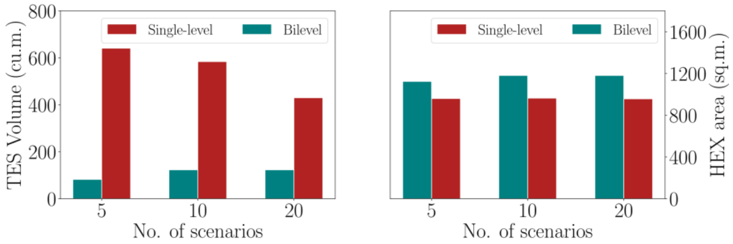


Figure 2 : TES volume and HEX area for the two formulations

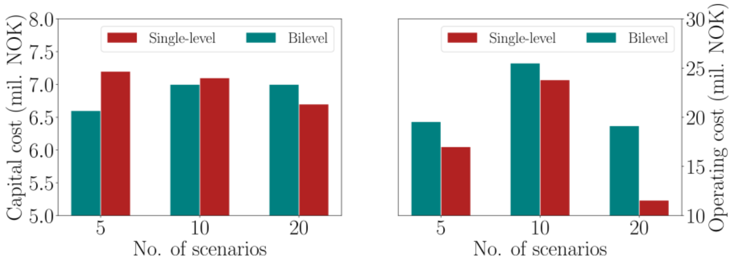


Figure 3 : Design and operating cost for the two formulations

5. Conclusion

In this paper, we compared two stochastic formulations for design optimization of TES systems. The results show that the bilevel formulation prioritizes minimizing the design cost, leading to higher operating costs. On the other hand, the single-level formulation minimizes the overall cost, but does not explicitly account for optimal operation.

Acknowledgements

This publication has been funded by HighEFF - Centre for an Energy Efficient and Competitive Industry for the Future, an 8-year Research Centre under the FME-scheme (Centre for Environment-friendly Energy Research, 257632). The authors gratefully acknowledge the financial support from the Research Council of Norway and user partners of HighEFF. The authors thank Mo Fjernvarme AS in particular for providing industrial data on heat supply and demand.

References

J. R. Birge, F. Louveaux, 2011. Introduction to Stochastic Programming, 2nd Edition. Springer Publishing Company, Incorporated.

S. Dempe, S. Franke, 2019. Solution of bilevel optimization problems using the KKT approach. Optimization 68 (8), 1471–1489

J. Fortuny-Amat, B. McCarl, 1981. A representation and economic interpretation of a two-level programming problem. Journal of the Operational Research Society 32 (9), 783–792

R. Sinnott, G. Towler, 2009. Chemical Engineering Design. Butterworth-Heinemann/IChemE series. ButterworthHeinemann.

B. Xu, Y. Wang, Y. Dvorkin, R. Fernández-Blanco, C. A. Silva-Monroy, J. Watson, D. S. Kirschen, 2017. Scalable planning for energy storage in energy and reserve markets. IEEE Transactions on Power Systems 32 (6), 4515–4527.