Sensitivity-Assisted Multistage Nonlinear Model Predictive Control: Robustness, Stability and Computational Efficiency

Mandar Thombre[a], Zhou (Joyce) Yu[b], Johannes Jäschke[a], Lorenz T. Biegler[b]

[a] Department of Chemical Engineering, Norwegian University of Science and Technology, Trondheim 7491, Norway

[b] Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Emails: mandar.thombre@ntnu.no, zhouy1@andrew.cmu.edu, johannes.jaschke@ntnu.no, lb01@andrew.cmu.edu

Corresponding author: Lorenz T. Biegler, lb01@andrew.mu.edu

Abstract

Key requirements for robust nonlinear model predictive control (NMPC) are stability, efficient performance under uncertainty, constraint satisfaction and computational efficiency. Multistage NMPC, based on a scenario tree formulation for the uncertainty, has been shown to satisfy the first three objectives under plant-model mismatch. However, a limiting factor in multistage NMPC, is the exponential scaling of the scenarios with respect to uncertain parameters and the length of the robust horizon. To address this issue, we present an approximate sensitivity-assisted multistage NMPC (samNMPC) scheme that reduces the problem size by dividing the scenario set into 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 nonlinear programming (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 robust horizon. This allows for faster computations with longer robust horizons that rigorously account for future uncertainty. In this paper, we explore the samNMPC approach and discuss its robust stability properties in context of the robust horizon. We demonstrate the applicability of the approach for the continuous stirred tank reactor (CSTR) and the quadtank case studies for tracking setpoints, and show that samNMPC compares favorably in performance and robustness to ideal multistage NMPC, but with a significant reduction in computational cost.

Keywords: *Robust Nonlinear Model Predictive Control, Dynamic Optimization, Stochastic Programming, Sensitivity*

1. Introduction

Model predictive control (MPC) is a powerful tool that has been widely used for control and optimization in the chemical process industry, 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 [30]. Plant dynamics are often highly nonlinear, and hence the nonlinear counterpart of MPC (NMPC) has received attention.

The presence of plant-model mismatch or noise can easily cause the system to violate constraints or become suboptimal. As such, robust NMPC approaches that rigorously handle the uncertainty have been studied in the past few decades. Although standard NMPC provides some inherent robustness against uncertainty, this is not enough when the uncertainty is pronounced. Different approaches for handling uncertainty have been proposed in the literature, going back to the min-max MPC presented in [3], 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 (i.e., *feedback*) actions that may counteract the uncertainty. Feedback min-max

MPC was proposed in [32], where closed-loop optimization is sought over different sequences of control inputs for different realizations of the uncertainty. Extending this approach, multistage NMPC proposed in [19] offers robustness in terms of constraint satisfaction without being overly conservative. Here, the uncertainty propagates through time in the form of a scenario tree, with each scenario representing a discrete realization of the uncertainty.

The major challenge in multistage NMPC, however, is that the computational size of the problem grows 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, and 3) the length of the prediction horizon. The problem quickly becomes computationally intractable, and this poses a challenge for real-time implementation of NMPC. To expedite computations, various approximation strategies have been proposed. Most implementations of multistage NMPC apply so-called robust horizon [19], 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. A parallelizable advanced-step multistage NMPC algorithm proposed in [39] precomputes a set of solutions offline, and makes a quick sensitivity-based correction online to reduce computational effort. A data-driven method of selecting scenarios is shown in [35], where the uncertainty information is captured with fewer scenarios using multivariate data analysis. Alternatively, cost-to-go functions of different scenarios were approximated by neural networks in [4] and applied to a semi-batch reactor. To exploit the inherent structure of the multistage NMPC problem, algorithms based on primal decomposition [16] and dual decomposition [20, 23, 17] have been proposed. An online scenario generation approach that approximates the multistage NMPC with a much smaller scenario tree is proposed in [11]. This method is based on finding the worst-case realizations of uncertainty with respect to constraint feasibilities.

In this study, we extend the sensitivity-based robust NMPC scheme recently proposed in [38, 40]. We develop an efficient approximation to the ideal multistage NMPC problem by dividing the scenarios into two sets: a (small) set of so-called *critical* scenarios, and a (larger) set of *non-critical* scenarios. The critical scenarios are those most likely to violate inequality constraints, and are composed of the corresponding uncertainty realizations. 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 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.

The recursive feasibility and stability properties of the proposed method are also discussed in this paper. By relaxing the bound constraints on the states, recursive feasibility of samNMPC (and in fact, of ideal multistage NMPC as well) can be guaranteed, even with the robust horizon assumption. Further, under suitable assumptions, samNMPC demonstrates input-to-state practical stability (ISpS) properties in the presence of bounded uncertainty.

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 ideal multistage NMPC formulations, in terms of robustness and computational efficiency.

The remainder of this paper is structured as follows: Section 2 presents the standard and ideal multistage NMPC formulations, along with the concept of NLP sensitivity. Section 3 presents the samNMPC methodology and implementation, and Section 4 discusses its stability properties. Section 5 includes the two case studies, and Section 6 concludes the paper.

2. Background development

2.1. Standard NMPC

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

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

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:

$$\min_{\mathbf{z}_l, \mathbf{v}_l} \quad \phi(\mathbf{z}_N, \mathbf{d}^0_{N-1}) + \sum_{l=0}^{N-1} \varphi(\mathbf{z}_l, \mathbf{v}_l, \mathbf{d}^0_l) \tag{2a}$$

$$\text{s.t.} \quad \mathbf{z}_{l+1} = \mathbf{f}(\mathbf{z}_l, \mathbf{v}_l, \mathbf{d}^0_l) \qquad l = 0, \ldots, N-1 \tag{2b}$$

$$\mathbf{z}_0 = \mathbf{x}_k \tag{2c}$$

$$\mathbf{z}_l \in \mathbb{X}, \mathbf{v}_l \in \mathbb{U}, \mathbf{z}_N \in \mathbb{X}_f \tag{2d}$$

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 $\mathbf{d}^0_l$ for all time steps.

The objective function (2a) is composed of the stage cost $\varphi : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \to \mathbb{R}$, and the terminal cost $\phi : \mathbb{R}^{n_x} \times \mathbb{R}^{n_d} \to \mathbb{R}$. Constraints (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 (2c), and (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 NMPC controller.

At each time step k, problem (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 $\mathbf{h} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$. After injecting $\mathbf{u}_k$, the plant evolves from t_k to t_{k+1} according to (1). The updated state $\mathbf{x}_{k+1}$ at t_{k+1} is used to solve problem (2) at the next time step, and the procedure repeats.

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 [5, 37]. However the plant-model mismatch arising due to uncertainty causes deteriorating performance in standard NMPC, particularly with respect to constraint satisfaction. A comprehensive discussion on standard NMPC can be found in [30].

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). The sensitivity-assisted multistage NMPC scheme offers this advantage, and will be discussed in Section 3.

2.2. Ideal Multistage NMPC

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 $\{\mathbf{x}_{[k,k+N-1]}\}_{\mathbb{D}}$ [10]. The min-max MPC solves for a single control profile that applies to all realizations of the uncertainty, including the worst-case realization [3].

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 [24, 25]). 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 [32], 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 [19] in the context of nonlinear systems to propose the multistage NMPC.

Consider that the continuous uncertainty set $\mathbb{D}$ is discretized into a set $\mathbb{M}$ of discrete realizations. The heuristic suggested in [19] 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:

$$\mathbb{M} = \{d_1^{\mathsf{max}}, d_1^{\mathsf{nom}}, d_1^{\mathsf{min}}\} \times \cdots \times \{d_{n_d}^{\mathsf{max}}, d_{n_d}^{\mathsf{nom}}, d_{n_d}^{\mathsf{min}}\} \tag{3}$$

where n_d is the dimensionality of the uncertain parameter vector, and $|\mathbb{M}| = 3^{n_d}$. Figure 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 as we move forward in time, 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 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

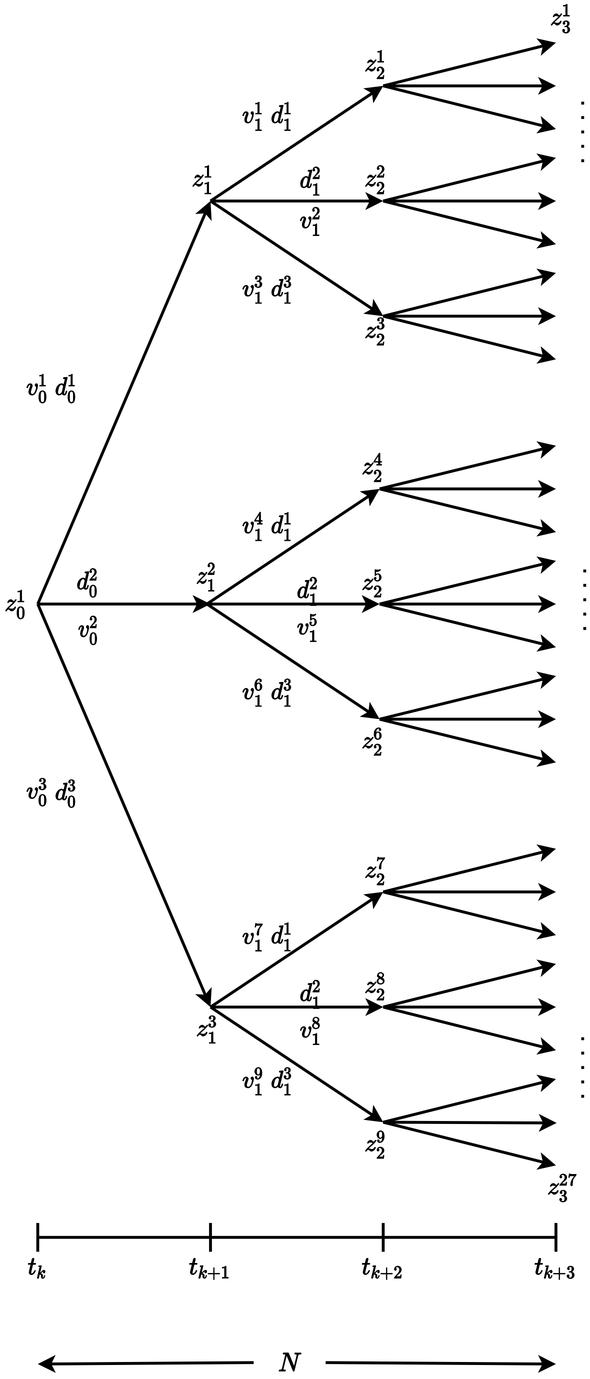

Figure 1: Fully branched scenario tree with $n_d = 1$ and $N = 3$.

grows exponentially large, and it becomes computationally infeasible to solve the resulting optimization problem. To resolve this, a robust horizon of length N_r was suggested in [19], 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 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:

$$\begin{aligned}
\min_{\mathbf{z}_l^c, \mathbf{v}_l^c} \quad & \sum_{c \in \mathbb{C}} \omega^c \Big(\phi(\mathbf{z}_N^c, \mathbf{d}_{N-1}^c) + \sum_{l=0}^{N-1} \varphi(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^c) \Big) && \text{(4a)} \\
\text{s.t.} \quad & \mathbf{z}_{l+1}^c = \mathbf{f}(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^c) \quad l = 0, \dots, N-1 && \text{(4b)} \\
& \mathbf{z}_0^c = \mathbf{x}_k && \text{(4c)} \\
& \mathbf{v}_l^c = \mathbf{v}_l^{c'} \quad \{(c, c') \mid \mathbf{z}_l^c = \mathbf{z}_l^{c'}\} && \text{(4d)} \\
& \mathbf{z}_l^c \in \mathbb{X}, \mathbf{v}_l^c \in \mathbb{U}, \mathbf{z}_N^c \in \mathbb{X}_f, \mathbf{d}_l^c \in \mathbb{D} && \text{(4e)} \\
& \mathbf{d}_{l-1}^c = \mathbf{d}_l^c \quad l = N_r, \dots N-1 && \text{(4f)} \\
& \forall c, c' \in \mathbb{C}
\end{aligned}$$

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 (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 (4f) imposes that the uncertain parameters remain constant after the robust horizon.

Equation (4d) represents the non-anticipativity constraints (NACs) which impose that all control inputs corresponding to branches of the same parent node

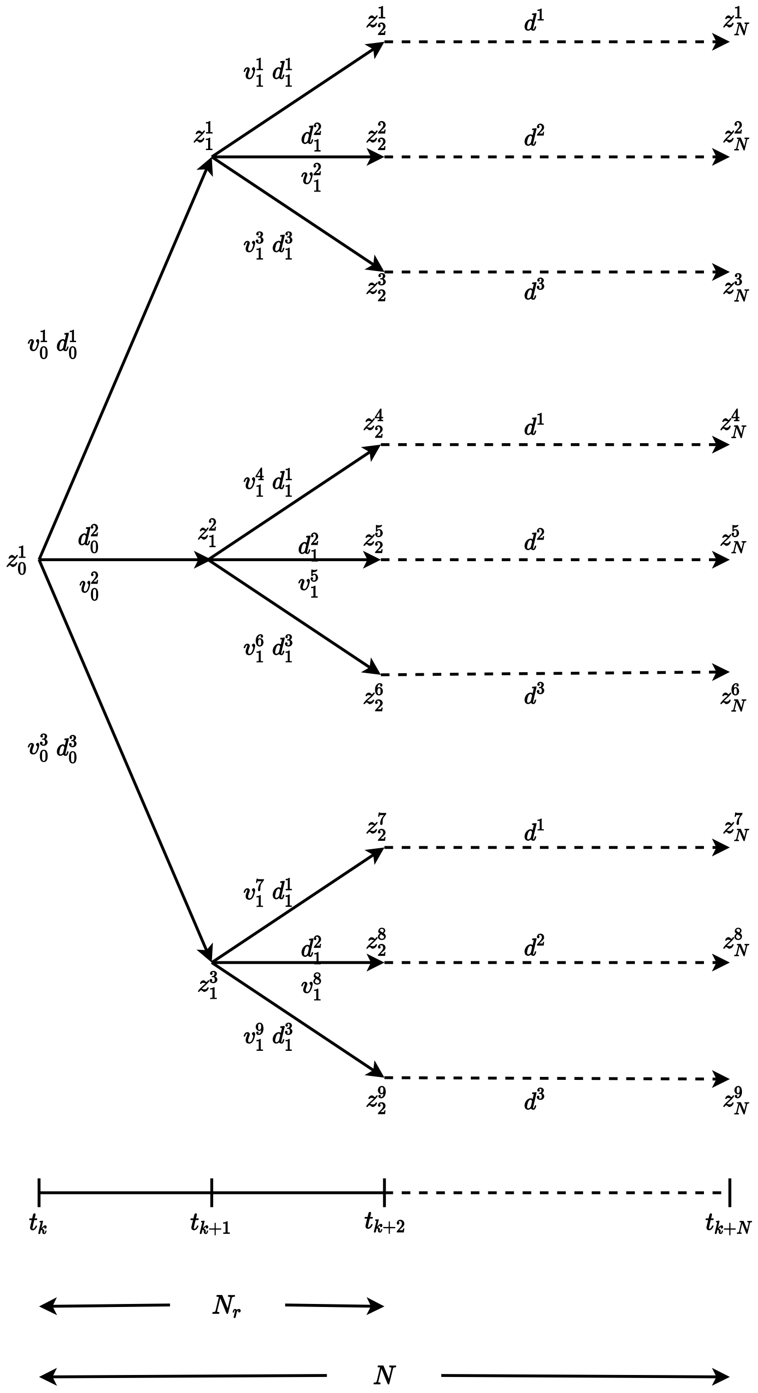

Figure 2: Robust horizon assumption: scenario tree with $n_d = 1$ and $N_r = 2$.

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

It must be noted that in ideal 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. In contrast, as will be discussed in Section 3, the scenario tree in the proposed sensitivity-assisted multistage NMPC does *not* scale exponentially with the number of uncertain parameters, allowing for computations with longer robust horizons.

2.3. NLP sensitivity

Because Problem (4) is parametric in its uncertainty $\mathbf{d}_k$, we investigate the sensitivity properties of (4) with respect to $\mathbf{d}_k$ by rewriting this problem in the more compact form:

$$\begin{aligned} \min_{x} \quad & F(x;p) \\ \text{s.t.} \quad & c(x;p) = 0 \\ & x \geq 0 \end{aligned} \tag{5}$$

where x are all the variables in (4), and p are all the uncertain parameters $\mathbf{d}$ of (4). The solution of (5) is given by a KKT point, which satisfies the Karush-Kuhn-Tucker conditions for (5):

Definition 1. *(KKT, [26]) KKT conditions for Problem (5) are given by:*

$$\begin{aligned} & \nabla F(x^*) + \nabla c(x^*)\lambda - \nu = 0 \\ & c(x^*) = 0 \\ & 0 \leq \nu \perp x^* \geq 0 \end{aligned} \tag{6}$$

for some multipliers (λ, ν)*, where* x^* *is a KKT point. We also define* $\mathcal{L}(x, \lambda, \upsilon) = F(x) + \lambda^T c(x) - \nu^T x$ *as the Lagrangian function of* (5)*.*

A constraint qualification (CQ) is required so that a KKT point is necessary for a local minimizer of (5) [26]. For Problem (5), the following CQ is widely invoked.

Definition 2. *(LICQ, [26]) The linear independence constraint qualification (LICQ) holds at* x^* *when the gradient vectors*

$$\nabla c(x^*; p) \text{ and } \nabla x_j^*; \quad j \in J \text{ where } J = \{j | x_j^* = 0\} \tag{7}$$

are linearly independent.

The LICQ implies that the set of multipliers (λ, ν) satisfying the KKT conditions are unique.

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

Definition 3. *(SSOSC, [31]) 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^**:*

$$q^T \nabla_{xx} \mathcal{L}(x^*, \lambda, \nu; p) q \ \geq \epsilon > 0 \quad \text{ for all } q \neq 0 \tag{8}$$

such that

$$\begin{aligned} \nabla c_i(x^*; p)^T q = 0, &\quad i = 1, .., n_c \\ q_j = 0, &\quad \text{for } \nu_j \ \geq \epsilon > 0, j \in J. \end{aligned} \tag{9}$$

Finally, for active constraints sets we give the following definition:

Definition 4. *(SC, [26]) At a KKT point* x^* *with multipliers* (λ, ν)*, the strict complementarity condition (SC) is defined by* $\nu_j + x_j^* > 0$ *for each* $j \in J$*.*

To guarantee *differentiability* of $x^*(p)$ with respect to p, we require that LICQ, SSOSC and SC hold at the KKT point [26]. The `IPOPT` algorithm substitutes the inequality constraints in (5) with a barrier function in the objective and solves a sequence of problems, indexed by m with $\lim_{m\to\infty}\mu_m \to 0$:

$$\begin{aligned} \min_x \quad & F(x;p) - \mu_m \sum_{i=1}^{n_x} ln(x_i) \\ \text{s.t.} \quad & c(x;p) = 0 \end{aligned} \tag{10}$$

As $\mu_m \to 0$, the solutions of (10) approach the solution of (5). For a negligibly small $\mu_m > 0$, the Lagrangian of (10) can be denoted as:

$$\mathcal{L}(x, \lambda, \nu;\, p) := F(x;p) + \lambda^T c(x;p) - \nu^T x \tag{11}$$

and the KKT conditions of (10) at p are:

$$\begin{aligned} & \nabla_x \mathcal{L}(x^*, \lambda^*, \upsilon^*;\, p) = \nabla_x F(x^*;p) + \nabla_x c(x^*;p)\lambda^* - \upsilon^* = 0 \\ & c(x^*;p) = 0 \\ & X^* V^* e = \mu e \end{aligned} \tag{12}$$

where $X = diag(x)$, $V = diag(\upsilon)$ and $e^T = [1, \ldots, 1]$. The primal-dual solution vector is

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

At $p = p_0$, if the solution of (10) $s^*(p_0)$ satisfies LICQ, SSOSC and SC; and the functions $F(.)$ and $c(.)$ are twice continuously differentiable in the neighborhood of this solution, then the following properties hold [7, 8]:

- $s^*(p_0)$ is an isolated local minimizer of (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 (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 $|F(p) - F(p_0)| \leq L_J |p - p_0|$.

These properties allow the application of Implicit Function Theorem to (12) at $s^*(p_0)$, which leads to the following linear system for sensitivity of s:

$$\mathcal{M}(s(\mu; p_0))\Delta s = -\mathcal{N}(s(\mu; p_0); p) \tag{13}$$

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

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

with $s(0; p) = s(\mu; p_0) + \Delta s + O(|p - p_0|^2) + O(\mu)$. 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 as $\Delta s = -\mathcal{M}(s(\mu; p_0))^{-1}\mathcal{N}(s(\mu; p_0); p)$ using an inexpensive backsolve if the factorized form of $\mathcal{M}(s(\mu; p_0))$ is available. The approximate solution at p can be calculated as $\tilde{s}(p) = s^*(\mu; p_0) + \Delta s$. The most expensive step in the NLP algorithm is to formulate and factorize the KKT matrix. The main advantage of calculating the sensitivity step with (13) is that we avoid the expensive solution of problem (5) at p, which is crucial in the implementation of NMPC.

2.4. Implication of soft constraints

To extend the discussion to the notion of relaxed inequality (soft) constraints, consider the following definitions of another CQ and the generalized SSOSC:

Definition 5. *(MFCQ, [26]) For Problem (5), the Mangasarian-Fromovitz constraint qualification (MFCQ) holds at the optimal point $x^*(p)$ 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*

$$\nabla c(x^*;p)^T q = 0,\ q_j > 0 \quad j \in J. \tag{14}$$

The MFCQ implies that the set of multipliers (λ, ν) is a compact convex polytope [9].

Definition 6. *(GSSOSC, [15]) 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* (5)*.*

MFCQ and GSSOSC are the weakest conditions under which a perturbed solution of (5) is locally unique [15], and Lipschitz continuity of $x^*(p)$ with respect to p can be guaranteed.

To put this in context, we now relax the inequalities in (5) with penalty variables r to form:

$$\begin{aligned} \min_{x} \quad & F(x;p) + M(r^T e) \\ \text{s.t.} \quad & c(x;p) = 0 \\ & x + r \geq 0 \\ & r \geq 0 \end{aligned} \tag{15}$$

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 (15) are the discrete dynamic equations and NACs in (4), 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 [12]. Now, with the relaxation of the inequality constraints, it is straightforward to show that the MFCQ always holds [12] at the solution of (15).

Finally, by adding the quadratic term

$$\|x - x^*\|_W^2 \tag{16}$$

to the objective in (15), where W is a positive definite weighting matrix, solutions of the KKT conditions of (15) 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 3) 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 [36]). 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 problem (4) holds only if $r = 0$ at the solution. We note that for all the case study solutions in Section 5, a penalty weight M could always be chosen large enough so that $r = 0$ held, and there were no infeasible solutions to (4).

With these properties in mind, the standard NMPC problem (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:

$$J_{nom}(\mathbf{x}_k) \equiv \min_{\mathbf{z}_l, \mathbf{v}_l} \quad \phi(\mathbf{z}_N, \mathbf{d}_{N-1}^0) + \sum_{l=0}^{N-1} \varphi(\mathbf{z}_l, \mathbf{v}_l, \mathbf{d}_l^0) + M_\phi \mathbf{e}^T \mathbf{r}_N + \sum_{l=0}^{N-1} M_\varphi \mathbf{e}^T \mathbf{r}_l \tag{17a}$$

$$\text{s.t.} \quad \mathbf{z}_{l+1} = \mathbf{f}(\mathbf{z}_l, \mathbf{v}_l, \mathbf{d}_l^0) \qquad l = 0, \ldots, N-1 \tag{17b}$$

$$\mathbf{z}_0 = \mathbf{x}_k \tag{17c}$$

$$\mathbf{x}^L - \mathbf{r}_l \leq \mathbf{z}_l \leq \mathbf{x}^U + \mathbf{r}_l;\ \mathbf{r}_l \geq 0 \tag{17d}$$

$$\mathbf{x}_f^L - \mathbf{r}_N \leq \mathbf{z}_N \leq \mathbf{x}_f^U + \mathbf{r}_N;\ \mathbf{r}_N \geq 0 \tag{17e}$$

$$\mathbf{v}_l \in \mathbb{U} \tag{17f}$$

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 (17d) and (17e), and the corresponding penalty terms are added to the objective function J_{nom} in (17a).

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 paper, and make our subsequent problem formulations parametric in this $\mathbf{d}^c$. Relaxing the bounds on the state variables in (4) leads to the following reformulation:

$$J_{ms}^N(\mathbf{x}_k, \mathbf{d}^c) \equiv \min_{\mathbf{z}_l^c, \mathbf{v}_l^c} \quad \sum_{c \in \mathbb{C}} \omega_c \Big(\phi(\mathbf{z}_N^c, \mathbf{d}_{N-1}^c) + \sum_{l=0}^{N-1} \varphi(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^c) \Big) +$$

$$\sum_{c \in \mathbb{C}} \omega_c \Big(M_\phi \mathbf{e}^T \mathbf{r}_N^c + \sum_{l=0}^{N-1} M_\varphi \mathbf{e}^T \mathbf{r}_l^c \Big) \tag{18a}$$

$$\text{s.t.} \quad \mathbf{z}_{l+1}^c = \mathbf{f}(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^c) \quad l = 0, \ldots, N-1 \tag{18b}$$

$$\mathbf{z}_0^c = \mathbf{x}_k \tag{18c}$$

$$\mathbf{x}^L - \mathbf{r}_l^c \leq \mathbf{z}_l^c \leq \mathbf{x}^U + \mathbf{r}_l^c;\ \mathbf{r}_l^c \geq 0 \tag{18d}$$

$$\mathbf{x}_f^L - \mathbf{r}_N^c \leq \mathbf{z}_N^c \leq \mathbf{x}_f^U + \mathbf{r}_N^c;\ \mathbf{r}_N^c \geq 0 \tag{18e}$$

$$\mathbf{v}_l^c = \mathbf{v}_l^{c'} \quad \{(c, c') \mid \mathbf{z}_l^c = \mathbf{z}_l^{c'}\} \tag{18f}$$

$$\mathbf{v}_l^c \in \mathbb{U},\ \mathbf{d}_l^c \in \mathbb{D} \tag{18g}$$

$$\mathbf{d}_{l-1}^c = \mathbf{d}_l^c \quad l = N_r, \ldots, N-1 \tag{18h}$$

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

where we note that (18a) and its optimal objective function J_{ms}^N are parametric in $(\mathbf{x}_k, \mathbf{d}^c)$.

3. Sensitivity-assisted multistage NMPC

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 $\{\mathsf{max}, \mathsf{nominal}, \mathsf{min}\}$ realizations of an uncertain parameter, we find the "worst-case" realization that is most likely to cause violation of an inequality constraint [11]. 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, with soft constraints, is:

$$
\begin{aligned}
J^N_{sam}(\mathbf{x}_k, \mathbf{d}^c) \equiv \min_{\substack{\mathbf{z}^c_l, \mathbf{v}^c_l \\ c \in \widehat{\mathbb{C}} \cup \{0\}}} \quad & \sum_{c \in \widehat{\mathbb{C}} \cup \{0\}} \omega_c \Big(\phi(\mathbf{z}^c_N, \mathbf{d}^c_{N-1}) + \sum_{l=0}^{N-1} \varphi(\mathbf{z}^c_l, \mathbf{v}^c_l, \mathbf{d}^c_l) \Big) + \\
& \sum_{c \in \widehat{\mathbb{C}} \cup \{0\}} \omega_c \Big(M_\phi \mathbf{e}^T \mathbf{r}^c_N + \sum_{l=0}^{N-1} M_\varphi \mathbf{e}^T \mathbf{r}^c_l \Big) + \\
& \sum_{c \in \bar{\mathbb{C}}} \omega_c \phi(\mathbf{z}^0_N + \Delta \mathbf{z}^c_N, \mathbf{d}^c_{N-1}) + \\
& \sum_{c \in \bar{\mathbb{C}}} \omega_c \sum_{l=0}^{N-1} \varphi(\mathbf{z}^0_l + \Delta \mathbf{z}^c_l, \mathbf{v}^0_l + \Delta \mathbf{v}^c_l, \mathbf{d}^c_l) \qquad (19a)
\end{aligned}
$$

$$
\begin{aligned}
\text{s.t.} \quad & \mathbf{z}^c_{l+1} = \mathbf{f}(\mathbf{z}^c_l, \mathbf{v}^c_l, \mathbf{d}^c_l) \quad l = 0, \ldots, N-1 && (19b) \\
& \mathbf{z}^c_0 = \mathbf{x}_k && (19c) \\
& \mathbf{x}^L - \mathbf{r}^c_l \leq \mathbf{z}^c_l \leq \mathbf{x}^U + \mathbf{r}^c_l;\ \mathbf{r}^c_l \geq 0 && (19d) \\
& \mathbf{x}^L_f - \mathbf{r}^c_N \leq \mathbf{z}^c_N \leq \mathbf{x}^U_f + \mathbf{r}^c_N;\ \mathbf{r}^c_N \geq 0 && (19e) \\
& \mathbf{v}^c_l = \mathbf{v}^{c'}_l \quad \{(c, c') \mid \mathbf{z}^c_l = \mathbf{z}^{c'}_l\} && (19f) \\
& \mathbf{v}^c_l \in \mathbb{U}, \mathbf{d}^c_l \in \mathbb{D} && (19g) \\
& \mathbf{d}^c_{l-1} = \mathbf{d}^c_l \quad l = N_r, \ldots N-1 && (19h) \\
& \forall c, c' \in \widehat{\mathbb{C}} \cup \{0\}
\end{aligned}
$$

where $\widehat{\mathbb{C}}$ and $\bar{\mathbb{C}}$ are the critical and noncritical scenario sets, respectively. The scenario $\{0\}$ represents the nominal scenario. 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 (18). The equality and inequality constraints (19b) – (19h) are imposed only for the nominal and critical scenarios. As shown in (19a), the noncritical scenarios are approximated with their NLP-sensitivity steps $\Delta \mathbf{z}^c$ and $\Delta \mathbf{v}^c$ in the objective function, and the soft constraints on state bounds are penalized with appropriate large weights.

Problem (19) is a partially linearized version of (18), 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.

3.1. 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 (19) 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 $l' = 0, \ldots, l-1$:

$$\max_{\mathbf{d}_{l'}} \quad g_j(\mathbf{z}_l, \mathbf{v}_l, \mathbf{d}_l) \tag{20a}$$

$$\text{s.t.} \quad \mathbf{z}_{\bar{l}+1} = \mathbf{f}(\mathbf{z}_{\bar{l}}, \mathbf{v}_{\bar{l}}, \mathbf{d}_{\bar{l}}) \quad \bar{l} = l', \ldots, l-1 \tag{20b}$$

$$\mathbf{z}_0 = x_k \tag{20c}$$

Problem (20) is solved around a reference trajectory $(\mathbf{z}_l, \mathbf{v}_l)|_{ref}$, $l = 1, \ldots, N$. From (20) we observe that we have the implicit relationship that:

$$g_j(\mathbf{z}_l, \mathbf{v}_l, \mathbf{d}_l) = g_j(\mathbf{z}_l(\mathbf{d}_{l'}), \mathbf{v}_l, \mathbf{d}_l) = g_j(\mathbf{d}_{l'}) \leq 0, \; l' + 1 \leq l = 1, \ldots, N. \tag{21}$$

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 (20) 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 $\dfrac{d\mathbf{g}}{d\mathbf{d}} \in$

$\mathbb{R}^{n_g \times n_d N_r}$ around the reference trajectory, we can then write:

$$\frac{d\mathbf{g}}{d\mathbf{d}}^T = \nabla_{\mathbf{z}}\mathbf{g}^T(\frac{d\mathbf{z}}{d\mathbf{d}})^T + \nabla_{\mathbf{d}}\mathbf{g}^T \tag{22}$$

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

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 (20), with fixed $\mathbf{v}_l$. Differentiation by the Implicit Function Theorem yields:

$$\begin{aligned} &\nabla_{\mathbf{z}}\mathbf{c}^T(d\mathbf{z}) + \nabla_{\mathbf{d}}\mathbf{c}^T(d\mathbf{d}) = 0 \\ &\frac{d\mathbf{z}}{d\mathbf{d}}^T = -(\nabla_{\mathbf{z}}\mathbf{c}^{-T})\nabla_{\mathbf{d}}\mathbf{c}^T. \end{aligned} \tag{23}$$

Substituting in (22) leads to:

$$\frac{d\mathbf{g}}{d\mathbf{d}}^T = -\nabla_{\mathbf{z}}\mathbf{g}^T(\nabla_{\mathbf{z}}\mathbf{c}^{-T})\nabla_{\mathbf{d}}\mathbf{c}^T + \nabla_{\mathbf{d}}\mathbf{g}^T \tag{24}$$

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 $(d_m)_{m=1,\cdots,n_d}$ is either d_m^{max} or d_m^{min}. Combining this assumption with strict monotonicity, the analytical solution of (20) can be stated for $l \in N_r$ and $m = 1, \cdots, n_d$:

$$\begin{aligned} d_{l,m}^{wc} &= \arg\max_{\mathbf{d}_l \in \mathbb{D}} \quad \frac{d\mathbf{g}}{d\mathbf{d}}^T \mathbf{d} \\ &= \begin{cases} d_{l,m}^{min}, & \text{if } \dfrac{d(g_j)}{d(d_{l,m})}|_{(\mathbf{z}_l,\mathbf{v}_l)|_{ref}} \leq 0 \\ d_{l,m}^{max}, & \text{otherwise} \end{cases} \end{aligned} \tag{25}$$

The result that worst-case realizations of the uncertainty are always on the extremes follows directly from the strict monotonicity 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 [2]. 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 (25) for the inequality constraints that are active or are close to the state trajectory. Indeed, if an element $|\frac{d(g_j)}{d(d_{l,m})}| \leq \epsilon$, 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 3, 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 (19).

3.2. 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 formulation (4) (fol-

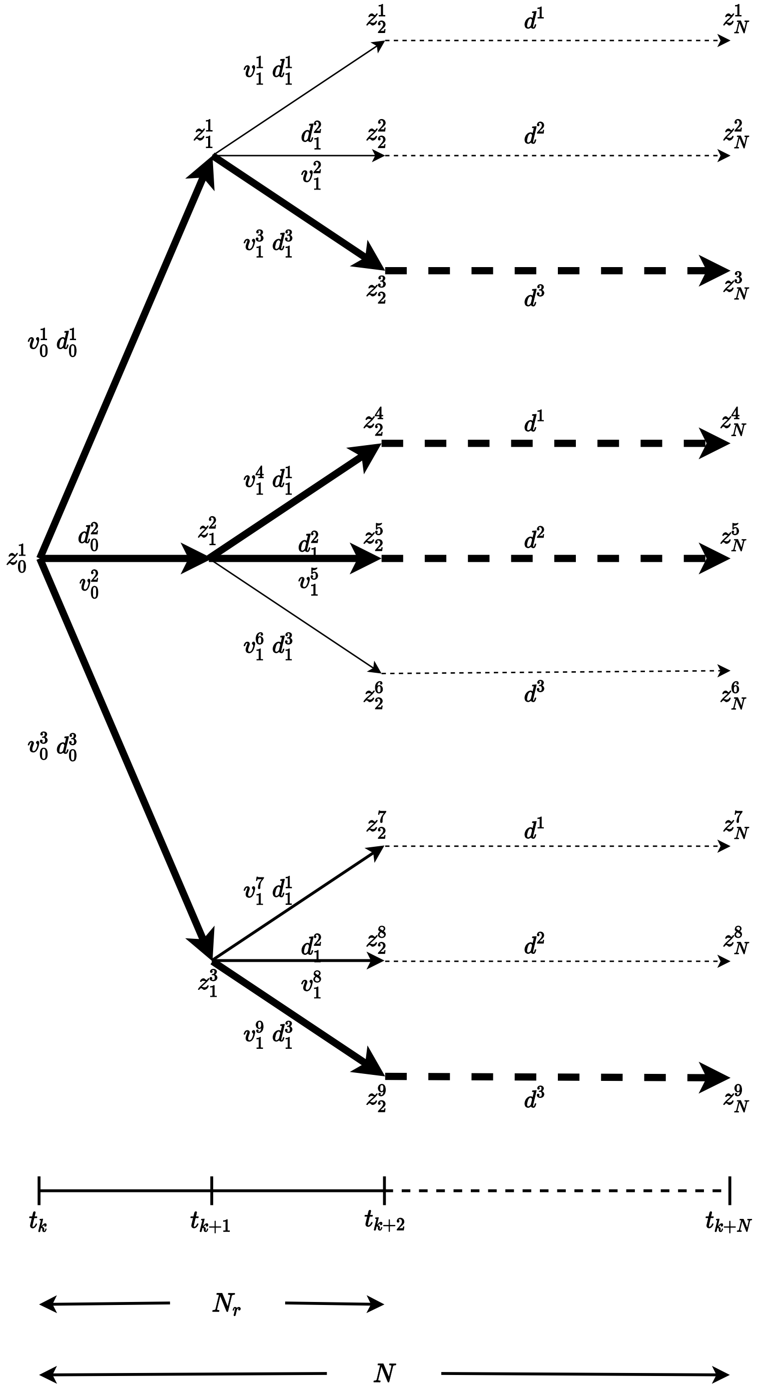

Figure 3: Example of a reduced scenario tree with $n_d = 1$ and $N_r = 2$. The nominal and critical scenarios are shown with the thick lines.

lowing the notation of Section 2.3):

$$\begin{bmatrix}
\mathbf{W}_0 & & & & \mathbf{A}_0 & & & & \widetilde{\mathbf{N}}_0 \\
& \mathbf{W}_1 & & & & \mathbf{A}_1 & & & \widetilde{\mathbf{N}}_1 \\
& & \ddots & & & & \ddots & & \vdots \\
& & & \mathbf{W}_{n_c} & & & & \mathbf{A}_{n_c} & \widetilde{\mathbf{N}}_{n_c} \\
\mathbf{A}_0^T & & & & & & & & \\
& \mathbf{A}_1^T & & & & & & & \\
& & \ddots & & & & & & \\
& & & \mathbf{A}_{n_c}^T & & & & & \\
\widetilde{\mathbf{N}}_0^T & \widetilde{\mathbf{N}}_1^T & \dots & \widetilde{\mathbf{N}}_{n_c}^T & & & & &
\end{bmatrix} \tag{26}$$

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 $\widetilde{\mathbf{N}}_c$:

$$\mathbf{A}_c = \nabla_{x^c} c_i(x) \quad \forall i \in \bar{\mathcal{M}} \tag{27a}$$

$$\widetilde{\mathbf{N}}_c = \nabla_{x^c} c_j(x) \quad \forall j \in \widehat{\mathcal{M}} \tag{27b}$$

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 (26):

$$\begin{bmatrix}
\mathbf{W}_0 & \mathbf{A}_0 & & & & & & & \widetilde{\mathbf{N}}_0 \\
\mathbf{A}_0^T & & & & & & & & 0 \\
& & \mathbf{W}_1 & \mathbf{A}_1 & & & & & \widetilde{\mathbf{N}}_1 \\
& & \mathbf{A}_1^T & & & & & & 0 \\
& & & & \ddots & & & & \vdots \\
& & & & & \ddots & & & \vdots \\
& & & & & & \mathbf{W}_{n_c} & \mathbf{A}_{n_c} & \widetilde{\mathbf{N}}_{n_c} \\
& & & & & & \mathbf{A}_{n_c}^T & & 0 \\
\widetilde{\mathbf{N}}_0^T & 0 & \widetilde{\mathbf{N}}_1^T & 0 & \dots & \dots & \widetilde{\mathbf{N}}_{n_c}^T & 0 &
\end{bmatrix} \tag{28}$$

the linear system (13) can be rewritten in the following block-bordered-diagonal (BBD) form:

$$\begin{bmatrix} \mathbf{K}_0 & & & & \mathbf{N}_0 \\ & \mathbf{K}_1 & & & \mathbf{N}_1 \\ & & \ddots & & \vdots \\ & & & \mathbf{K}_{n_c} & \mathbf{N}_{n_c} \\ \mathbf{N}_0^T & \mathbf{N}_1^T & \dots & \mathbf{N}_{n_c}^T & \end{bmatrix} \begin{bmatrix} \Delta s_0 \\ \Delta s_1 \\ \vdots \\ \Delta s_{n_c} \\ \gamma \end{bmatrix} = - \begin{bmatrix} t_0 \\ t_1 \\ \vdots \\ t_{n_c} \\ 0 \end{bmatrix} \tag{29}$$

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 (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 (29), $\mathbf{N}_c$ 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 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 NAC. Additionally, γ in (29) is the multiplier associated with NAC (4d) with the dimension $\gamma \in \mathbb{R}^{m_{NAC} \times n_u}$.

Solving the linear system (29) for the full multistage problem is computationally expensive for a large number of scenarios. We seek an approximate solution of (29) that is fast to compute. To be specific, we solve the standard NMPC formulation (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 (29).

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

$$|\Delta\mathbf{d}| = \max_{j,j' \in \mathbb{M}} |\mathbf{d}^j - \mathbf{d}^{j'}| \tag{30}$$

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 (29) 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 (29) 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 (29) 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 (29). For the KKT matrix in (29), the Schur complement is formed as:

$$S = \sum_{c\in\mathbb{C}} (\mathbf{N}_c^T \mathbf{K}_c^{-1} \mathbf{N}_c) \tag{31}$$

The NAC multipliers γ can then be obtained by solving:

$$S\,\gamma = -\sum_{c\in\mathbb{C}} (\mathbf{N}_c^T \mathbf{K}_c^{-1} t_c) \tag{32}$$

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

$$\mathbf{K}_c \Delta\mathbf{s}_c = -(t_c + \mathbf{N}_c\gamma),\ \forall c \in \mathbb{C} \tag{33}$$

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 (31) and (32) 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 (33) 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 3.1, the sensitivity steps of the noncritical scenarios $c \in \bar{\mathbb{C}}$ are included in the objective function of the samNMPC formulation (19). Note that the NACs of the noncritical scenarios are still satisfied in the approximated linear system (29), and thus they are implicitly taken into account in the samNMPC formulation (19).

3.3. Overall samNMPC algorithm and implementation

The overall samNMPC strategy is summarized in Algorithm 1.

```
Algorithm 1: sensitivity-assisted Multistage NMPC
Given: {max, nominal, min} of all uncertain parameters.
for $k = 1, 2, \cdots$ do
    Get the current state of the plant $\mathbf{x}_k$.
    Solve the standard NMPC problem (17) 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 (24)
      to form the critical scenario set $\widehat{\mathbb{C}}$.
    For noncritical scenarios: Solve the linear system (29) 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 (19), 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
```

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 $O(|\Delta\mathbf{d}|)$, as will be shown in Section 4.

The samNMPC algorithm is implemented using the software `JuMP` (version 0.19.2) [6], 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 [1]. The NLP solver used within this framework is `IPOPT`, which uses interior-point algorithms to solve NLPs. The `MA57` linear solver from the Harwell Subroutine Library [33] 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 the robust horizon. The resulting approximate linear system (29) 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 (19) 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. This would add to the computation time but would still be better than solving for the global worst-case scenario. Although such a rigorous treatment of critical scenarios is not considered in this work, the results of the case studies in Section 5 indicate that there were no constraint violations for noncritical scenarios.

4. Stability properties

Previous studies [18, 21, 22] have analyzed the recursive feasibility and stability properties of ideal multistage NMPC. However, these studies consider only the fully expanded scenario tree ($N_r = N$). In this section, 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, and then proceed to discuss the recursive feasibility and input-to-state practical stability (ISpS) for both the ideal multistage and samNMPC.

4.1. Preliminaries

We start with some fundamental concepts needed for the stability analysis of ideal multistage 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 (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, $\forall \mathbf{x} \in \mathbb{X}$*

$$V(\mathbf{x}) \geq \alpha_1(|\mathbf{x}|) \tag{34a}$$

$$V(\mathbf{x}) \leq \alpha_2(|\mathbf{x}|) \tag{34b}$$

$$\Delta V(\mathbf{x}) = V(\mathbf{f}(\mathbf{x}, \mathbf{h}(\mathbf{x}))) - V(\mathbf{x}) \leq -\alpha_3(|\mathbf{x}|) \tag{34c}$$

Definition 9. *(Control invariant set, [30]) 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 $\mathbf{d} \in \mathbb{D}$,*

$$|\mathbf{x}_k| \leq \gamma(|\Delta \mathbf{d}|) = \beta(|\mathbf{x}_0|, k) + \gamma(|\hat{\mathbf{d}}_k - \mathbf{d}^0|) + c_d, \; \forall\, k \geq 0, \; \forall \mathbf{x}_0 \in \mathcal{X} \tag{35}$$

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, [29]) 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 $\forall \mathbf{d} \in \mathbb{D}$,*

$$\begin{aligned} V(\mathbf{x}) &\geq \alpha_1(|\mathbf{x}|) && \text{(36a)} \\ V(\mathbf{x}) &\leq \alpha_2(|\mathbf{x}|) + c_0 && \text{(36b)} \\ \Delta V(\mathbf{x}, \mathbf{d}) &= V(\mathbf{f}(\mathbf{x}, \mathbf{h}(\mathbf{x}), \mathbf{d})) - V(\mathbf{x}) \\ &\leq -\alpha_3(|\mathbf{x}|) + \sigma(|\hat{\mathbf{d}}_k - \mathbf{d}^0|) + c_1 && \text{(36c)} \end{aligned}$$

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.

To facilitate the stability discussion, we consider the nominal case of the ideal multistage NMPC problem (18) where we set $\mathbf{d}_l^c = \mathbf{d}_l^0 \;\; \forall c \in \mathbb{C}$:

$$
\begin{aligned}
J_{ms}^N(\mathbf{x}_k, \mathbf{d}^0) \equiv \min_{\mathbf{z}_l^c, \mathbf{v}_l^c} \quad & \sum_{c \in \mathbb{C}} \omega_c \Big(\phi(\mathbf{z}_N^c, \mathbf{d}_{N-1}^0) + \sum_{l=0}^{N-1} \varphi(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^0) \Big) + \\
& \sum_{c \in \mathbb{C}} \omega_c \Big(M_\phi \mathbf{e}^T \mathbf{r}_N^c + \sum_{l=0}^{N-1} M_\varphi \mathbf{e}^T \mathbf{r}_l^c \Big) \qquad \text{(37a)}
\end{aligned}
$$

$$
\begin{aligned}
\text{s.t.} \quad & \mathbf{z}_{l+1}^c = \mathbf{f}(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^0) \quad l = 0, \ldots, N-1 && \text{(37b)} \\
& \mathbf{z}_0^c = \mathbf{x}_k && \text{(37c)} \\
& \mathbf{x}^L - \mathbf{r}_l^c \leq \mathbf{z}_l^c \leq \mathbf{x}^U + \mathbf{r}_l^c;\ \mathbf{r}_l^c \geq 0 && \text{(37d)} \\
& \mathbf{x}_f^L - \mathbf{r}_N^c \leq \mathbf{z}_N^c \leq \mathbf{x}_f^U + \mathbf{r}_N^c;\ \mathbf{r}_N^c \geq 0 && \text{(37e)} \\
& \mathbf{v}_l^c = \mathbf{v}_l^{c'} \quad \{(c, c') \mid \mathbf{z}_l^c = \mathbf{z}_l^{c'}\} && \text{(37f)} \\
& \mathbf{v}_l^c \in \mathbb{U}, \mathbf{d}_l^0 \in \mathbb{D} && \text{(37g)} \\
& \mathbf{d}_{l-1}^0 = \mathbf{d}_l^0 \quad l = N_r, \ldots N-1 && \text{(37h)} \\
& \forall c, c' \in \mathbb{C}
\end{aligned}
$$

We note that if we similarly set $\mathbf{d}_l^c = \mathbf{d}_l^0$, $\forall c \in \mathbb{C}$ in the samNMPC problem (19), then (37) is also equivalent to this nominal case of (19), i.e. $J_{sam}^N(\mathbf{x}_k, \mathbf{d}^0) = J_{ms}^N(\mathbf{x}_k, \mathbf{d}^0)$.

4.2. Recursive feasibility for ideal multistage 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). 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

$$
\sum_{j \in \mathbb{M}} \omega_l^j = 1,\ l = \{0, \ldots, N_r - 1\} \qquad (38)
$$

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

$$\omega^c = \prod_{l=0}^{N_r-1} \omega_l^{j_l} \tag{39}$$

where the scenario c is represented as $c = \{j_0, j_1, \dots, j_{N_r-1}\}$. We assume that $\omega_l^j = \omega_{l'}^j \quad \forall l, l' \in \{0, 1, \dots, N_r - 1\}$, 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* (17d) - (17e) *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* (3)*. We therefore assume that* (17d) - (17e) *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* $\phi(\mathbf{f}(\mathbf{x}, \mathbf{h}_f(\mathbf{x}), \mathbf{d}^j), \mathbf{d}^j) - \phi(\mathbf{x}, \mathbf{d}^j) \leq -\varphi(\mathbf{x}, \mathbf{h}_f(\mathbf{x}), \mathbf{d}^j), \forall \mathbf{x} \in (X)_f^j$*.*

- *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* (17)*, equivalent to problem* (37)*, the horizon* N *and the weighted terminal cost are chosen sufficiently large* *so that the solution of* (37) *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* (17d)*. Existence of these solutions was shown in [27].*

From these assumptions we can state the following result:

Theorem 1. *(Recursive feasibility of ideal multistage and samNMPC) Suppose Assumptions 1 and 2 hold, then the soft-constrained problems* (18) *and* (19) *with a robust horizon* $(N_r \leq N)$ *are recursively feasible.*

Proof. Since MFCQ and GSSOSC hold for the relaxed problems (18) and (19), Lipschitz continuity holds at the optimal solutions of these problems for all values of $\mathbf{x}_k$ and $\mathbf{d}_l^c$ (see Section 2.4).

Comparing (18) with problem (37), it follows from NLP sensitivity that the difference of their respective solutions are bounded by:

$$\left| \Big(\mathbf{z}^*(\mathbf{d}^c), \mathbf{v}^*(\mathbf{d}^c), \mathbf{r}^*(\mathbf{d}^c) \Big) - \Big(\mathbf{z}^*(\mathbf{d}^0), \mathbf{v}^*(\mathbf{d}^0), \mathbf{r}^*(\mathbf{d}^0) \Big) \right| \leq L_S |\Delta \mathbf{d}| \tag{40}$$

where L_S is a positive Lipschitz constant, and $|\Delta \mathbf{d}|$ is as defined in (30). Similarly, we note that the same relations hold when we compare the samNMPC problem (19) with (37) (remember that (37) 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 (18) and their nominal solutions from (37), is also bounded by $O(|\Delta \mathbf{d}|)$. Similarly, the difference between optimal values $(\mathbf{z}_l, \mathbf{v}_l, \mathbf{r}_l)$ from (19) and their nominal solutions from (37) is bounded by $O(|\Delta \mathbf{d}|)$.

Thus, irrespective of the evolution of the scenario tree (and even with $N_r < N$), a nonnegative, bounded $\mathbf{r}_k$ can always be found that satisfies (18d) – (18e) for ideal multistage NMPC, and all of the (relaxed) constraints remain satisfied. In the case of samNMPC, similar arguments can be made for (19d) – (19e).

Thus, ideal multistage NMPC (18) and samNMPC (19) are recursively feasible. ■

4.3. ISpS for ideal multistage NMPC

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

Theorem 2. *(Robust stability of ideal multistage NMPC) Under Assumptions 1 and 2, the cost function J_{ms} obtained from the solution of (18) 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 (37). Here, since every scenario in ideal multistage NMPC becomes the nominal scenario, this problem has nominal (asymptotic) stability [30]. Let the optimal control sequence obtained from (37) 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 1, recursive feasibility is guaranteed for (37) and is valid for $N_r < N$; 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:

$$\begin{aligned} \widetilde{J}_{ms}^{N+1}(\mathbf{x}_k, \mathbf{d}^0) \quad &= \quad \sum_{c\in\mathbb{C}} \omega_c \Big(\phi(\mathbf{z}_{N+1}^c, \mathbf{d}_{N+1}^0) + \sum_{l=0}^{N} \varphi(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^0) \Big) \\ &\quad + \sum_{c\in\mathbb{C}} \omega_c \Big(M_\phi \mathbf{e}^T \mathbf{r}_{N+1}^c + \sum_{l=0}^{N} M_\varphi \mathbf{e}^T \mathbf{r}_l^c \Big) \end{aligned} \tag{41}$$

and the following is valid from Assumption 2:

$$\begin{aligned} \widetilde{J}_{ms}^{N+1}(\mathbf{x}_k, \mathbf{d}^0) - J_{ms}^{N}(\mathbf{x}_k, \mathbf{d}^0) \quad &= \quad \sum_{c\in\mathbb{C}} \omega^c \Big(\phi(\mathbf{z}_{N+1}^c, \mathbf{d}_{N+1}^0) - \phi(\mathbf{z}_N^c, \mathbf{d}_N^0) \Big) + \\ &\quad \sum_{c\in\mathbb{C}} \omega^c \varphi(\mathbf{z}_N^c, \mathbf{v}_N^c, \mathbf{d}_N^0) \\ &\leq \quad 0 \end{aligned} \tag{42}$$

Moreover, from (38) and (42) we have:

$$\begin{aligned} J_{ms}^{N}(\mathbf{x}_k, \mathbf{d}^0) \quad &\geq \quad \widetilde{J}_{ms}^{N+1}(\mathbf{x}_k, \mathbf{d}^0) \\ &= \quad \varphi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_l^0) + \sum_{j\in\mathbb{M}} \omega_j J_{ms}^{N} \Big(f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_l^0), \mathbf{d}^0 \Big) \\ &= \quad \varphi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_l^0) + J_{ms}^{N} \Big(f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_l^0), \mathbf{d}^0 \Big) \end{aligned} \tag{43}$$

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

$$J_{ms}^{N}\Big(f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_l^0), \mathbf{d}^0\Big) - J_{ms}^{N}(\mathbf{x}_k, \mathbf{d}^0) \leq -\varphi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_l^0) \tag{44}$$

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

$$J_{ms}^{N}(\mathbf{x}_{k+1}, \mathbf{d}^0) - J_{ms}^{N}(\mathbf{x}_k, \mathbf{d}^0) \ \leq \ -\varphi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_l^0) + L_J|\hat{\mathbf{d}}_k - \mathbf{d}_k^0| \tag{45}$$

Finally, from (18) and (37) we have

$$\begin{aligned} J_{ms}^{N}(\mathbf{x}_k, \mathbf{d}^c) &= J_{ms}^{N}(\mathbf{x}_k, \mathbf{d}^0) + O(|\Delta\mathbf{d}|) & (46)\\ J_{ms}^{N}(\mathbf{x}_{k+1}, \mathbf{d}^c) &= J_{ms}^{N}(\mathbf{x}_{k+1}, \mathbf{d}^0) + O(|\Delta\mathbf{d}|) & (47) \end{aligned}$$

which can be combined with (45) to form:

$$\begin{aligned} J_{ms}^{N}(\mathbf{x}_{k+1}, \mathbf{d}^c) - J_{ms}^{N}(\mathbf{x}_k, \mathbf{d}^c) \ &\leq \ -\varphi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_l^0) + L_J|\hat{\mathbf{d}}_k - \mathbf{d}_k^0| + \\ &\quad \Big(J_{ms}^{N}(\mathbf{x}_{k+1}, \mathbf{d}^c) - J_{ms}^{N}(\mathbf{x}_{k+1}, \mathbf{d}^0)\Big) + \\ &\quad \Big(J_{ms}^{N}(\mathbf{x}_k, \mathbf{d}^0) - J_{ms}^{N}(\mathbf{x}_k, \mathbf{d}^c)\Big) \\ &\leq \ -\varphi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_l^0) + L_J|\hat{\mathbf{d}}_k - \mathbf{d}_k^0| + 2L_K|\Delta\mathbf{d}| \\ &= \ -\alpha(\mathbf{x}_k) + \sigma(|\hat{\mathbf{d}}_k - \mathbf{d}_k^0|) + c_d. \qquad (48) \end{aligned}$$

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

4.4. ISpS for samNMPC

To extend the stability discussions to samNMPC, we compare the formulations of the optimization problems (17) and (19). Since MFCQ and GSSOSC are satisfied for (17) and (19), Lipschitz continuity holds for $J_{nom}(\mathbf{x}_k)$ and $J_{sam}^{N}(\mathbf{x}_k, \mathbf{d}^c)$.

Theorem 3. *(Robust Stability of samNMPC) Under Assumptions 1 and 2, the cost function J_{sam} obtained from the solution of (19) is an ISpS-Lyapunov function, and the resulting closed-loop system is ISpS stable.*

Proof. From Theorem 1, recursive feasibility properties hold for samNMPC and are valid for $N_r \leq N$. To compare (17) and (19) we replicate (17) $|\mathbb{C}|$ times and we partitition the objective function into its critical, noncritical and nominal components, $J_{sam}(\mathbf{x}_k, \mathbf{d}^c) = \hat{J}(\mathbf{x}_k, \mathbf{d}^c) + \bar{J}(\mathbf{x}_k, \mathbf{d}^c) + \omega_0 J_{nom}(\mathbf{x}_k)$ using $\mathbb{C} = \widehat{\mathbb{C}} \cup \bar{\mathbb{C}} \cup \{0\}$ and the construction of (19).

For $\widehat{\mathbb{C}}$ and the solutions of (17) and (19) we note that:

$$
\begin{aligned}
\hat{J}(\mathbf{x}_k, \mathbf{d}^c) - \sum_{c \in \widehat{\mathbb{C}}} \omega_c \; J_{nom}(\mathbf{x}_k) \quad &= \quad \sum_{c \in \widehat{C}} \omega_c \Big((\phi(\mathbf{z}_N^c, \mathbf{d}_N^c) - \phi(\mathbf{z}_N^0, \mathbf{d}_N^0) \Big) + \\
& \quad\quad \sum_{c \in \widehat{C}} \omega_c M_\phi e^T (\mathbf{r}_N^c - \mathbf{r}_N) + \\
& \quad\quad \sum_{c \in \widehat{C}} \omega_c \sum_{l=0}^{N-1} \Big(\varphi(\mathbf{z}_l^c, \mathbf{v}_l^c, \mathbf{d}_l^c) - \varphi(\mathbf{z}_l^0, \mathbf{v}_l^0, \mathbf{d}_l^0)) \Big) + \\
& \quad\quad \sum_{c \in \widehat{C}} \omega_c \sum_{l=0}^{N-1} M_\varphi e^T (\mathbf{r}_l^c - \mathbf{r}_l) \\
&= \quad O(|\Delta \mathbf{d}|)
\end{aligned}
\tag{49}
$$

For $\bar{\mathbb{C}}$ we note from the solution of (17) and its KKT sensitivity that

$$
\begin{aligned}
\bar{J}(\mathbf{x}_k, \mathbf{d}^c) - \sum_{c \in \bar{\mathbb{C}}} \omega_c \; J_{nom}(\mathbf{x}_k) \quad &= \quad \sum_{c \in \bar{C}} \omega_c \Big(\phi(\mathbf{z}_N^0 + \Delta \mathbf{z}_N^c, \mathbf{d}_N^c) - \phi(\mathbf{z}_N^0, \mathbf{d}_N^0) \Big) + \\
& \quad\quad \sum_{c \in \bar{C}} \omega_c M e^T (\mathbf{r}_N^c - \mathbf{r}_N) + \\
& \quad\quad \sum_{c \in \bar{C}} \omega_c \sum_{l=0}^{N-1} \Big(\varphi(\mathbf{z}_l^0 + \Delta \mathbf{z}_l^c, \mathbf{v}_l^0 + \Delta \mathbf{v}_l^c, \mathbf{d}_l^c) - \varphi(\mathbf{z}_l^0, \mathbf{v}_l^0, \mathbf{d}_l^0) \Big) + \\
& \quad\quad \sum_{c \in \bar{C}} \omega_c \sum_{l=0}^{N-1} M e^T (\mathbf{r}_l^c - \mathbf{r}_l) \\
&= \quad O(|\Delta \mathbf{d}|).
\end{aligned}
\tag{50}
$$

From (19), (37), (49) and (50) we have

$$
\begin{aligned}
J_{sam}^N(\mathbf{x}_k, \mathbf{d}_l^c) \quad &= \quad J_{sam}^N(\mathbf{x}_k, \mathbf{d}_l^0) + O(|\Delta \mathbf{d}|) \\
J_{sam}^N(\mathbf{x}_{k+1}, \mathbf{d}_l^c) \quad &= \quad J_{sam}^N(\mathbf{x}_{k+1}, \mathbf{d}_l^0) + O(|\Delta \mathbf{d}|)
\end{aligned}
\tag{51}
$$

which can be combined with (45) to form:

$$\begin{aligned} J^N_{sam}(\mathbf{x}_{k+1}, \mathbf{d}^c_l) - J^N_{sam}(\mathbf{x}_k, \mathbf{d}^c_l) \quad &\leq \quad -\varphi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}^0_l) + L_J|\hat{\mathbf{d}}_k - \mathbf{d}^0_k|) + \\ &\quad\quad \Big(J^N_{sam}(\mathbf{x}_{k+1}, \mathbf{d}^c_l) - \sum_{c\in\mathbb{C}} \omega_c \; J_{nom}(\mathbf{x}_{k+1}) \Big) + \\ &\quad\quad \Big(\sum_{c\in\mathbb{C}} \omega_c \; J_{nom}(\mathbf{x}_k) - J^N_{sam}(\mathbf{x}_k, \mathbf{d}^c_l) \Big) \\ &\leq \quad -\varphi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}^0_l) + L_J|\hat{\mathbf{d}}_k - \mathbf{d}^0_k|) + 2L_K|\Delta\mathbf{d}| \\ &= \quad -\alpha(\mathbf{x}_k) + \sigma(|\hat{\mathbf{d}}_k - \mathbf{d}^0_k|) + c_d. \end{aligned} \tag{52}$$

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

It is worth mentioning here that, in Theorems 2 and 3, the constant c_d scales with $|\Delta\mathbf{d}|$ as shown in (48) and (52). Equation (30) 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 make the above results somewhat conservative.

5. Case Studies

5.1. CSTR example

We first consider the nonlinear benchmark CSTR problem [14], where the dynamics are described by the following equations:

$$\frac{dc_A}{dt} = F(c_{A0} - c_A) - k_1 c_A - k_3 c_A^2 \tag{53a}$$

$$\frac{dc_B}{dt} = -F c_B + k_1 c_A - k_2 c_B \tag{53b}$$

$$\frac{dT_R}{dt} = F(T_{in} - T_R) + \frac{k_W A}{\rho c_p V_R}(T_K - T_R) - \frac{k_1 c_A \Delta H_{AB} + k_2 c_B \Delta H_{BC} + k_3 c_A^2 \Delta H_{AD}}{\rho c_p} \tag{53c}$$

$$\frac{dT_K}{dt} = \frac{1}{m_K c_{pK}}(\dot{Q}_K + k_W A(T_R - T_K)) \tag{53d}$$

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

Table 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

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 1 and 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:

$$\varphi_l = (c_{Bl} - {c_B}_l^{ref})^2 + r_1 \Delta F_l^2 + r_2 \Delta \dot{Q_K} \tag{54}$$

Table 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

where $\Delta F_l = F_l - F_{l-1}$ and $\Delta \dot{Q}_{K\,l} = \dot{Q}_{K\,l} - \dot{Q}_{K\,l-1}$ 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 ±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 {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 4, 5 and 6, 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 4. 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

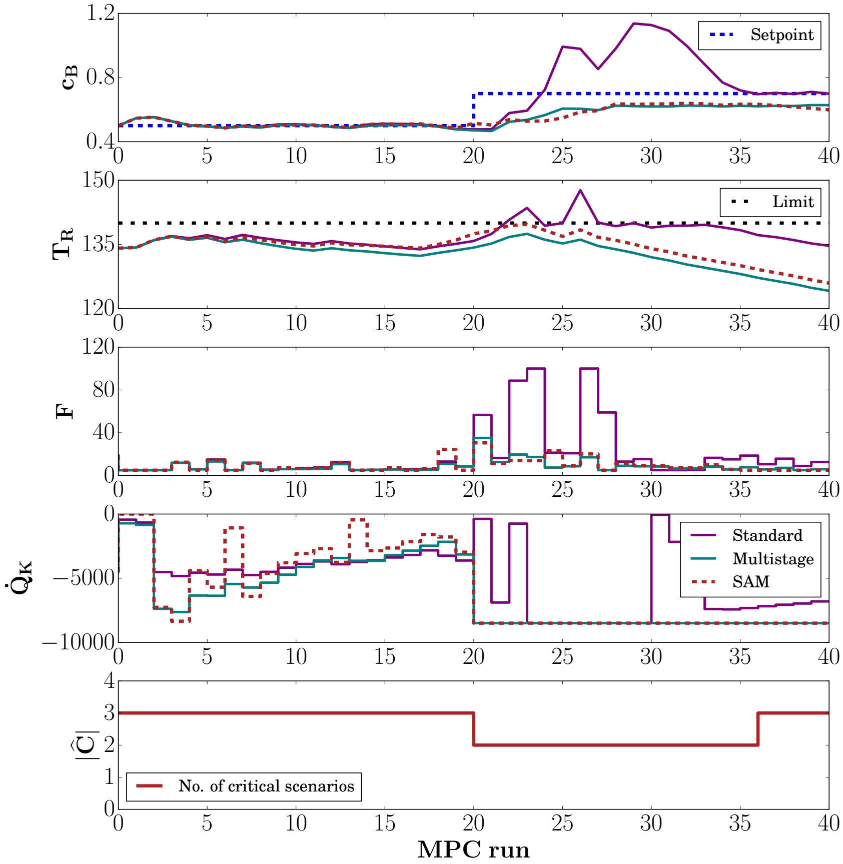

Figure 4: Setpoint tracking (c_B) and constraint satisfaction (T_R) for standard, ideal multistage and samNMPC with $N_r = 1$ (9 scenarios).

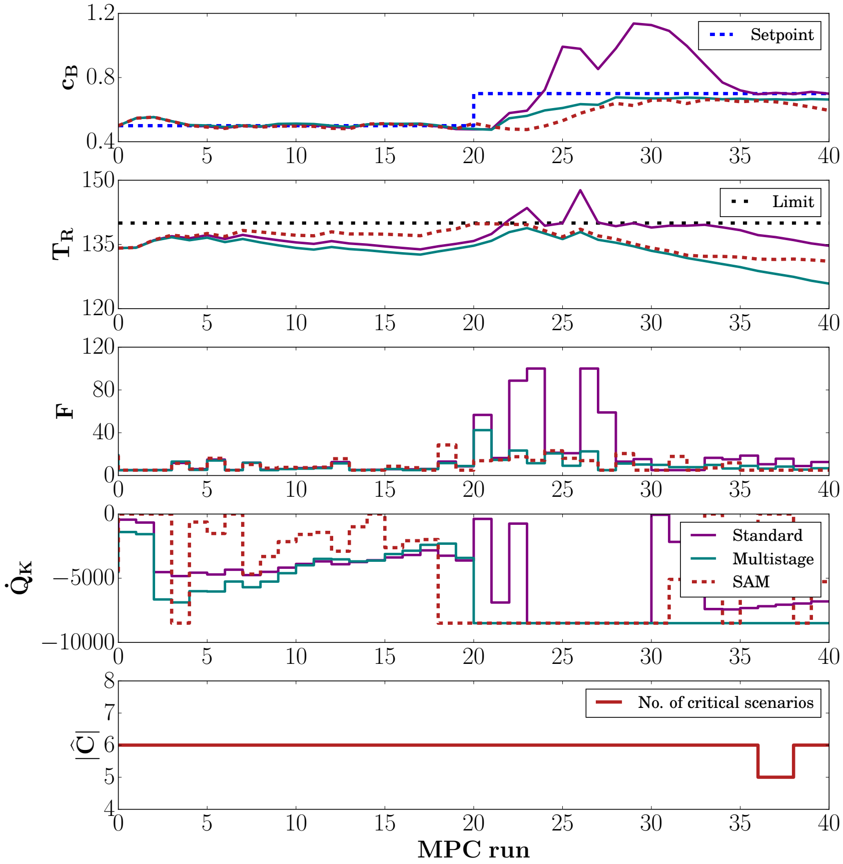

Figure 5: Setpoint tracking (c_B) and constraint satisfaction (T_R) for standard, ideal multistage and samNMPC with $N_r = 2$ (81 scenarios).

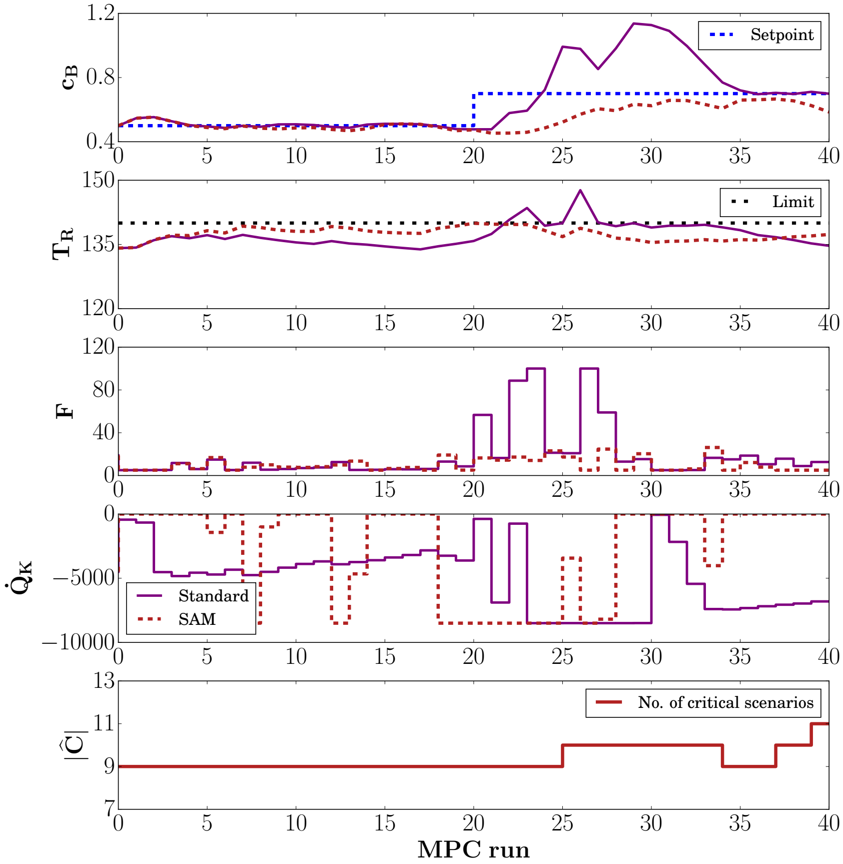

Figure 6: Setpoint tracking (c_B) and constraint satisfaction (T_R) for standard and samNMPC with $N_r = 3$ (729 scenarios).

Table 3: CSTR - Average computational performance in CPUs. For samNMPC, the times for solving linear system (29) and the NLP (19) are reported separately.

N_r	$\lvert\mathbb{C}\rvert$	Standard	Multistage	SAM - solving (29) / (19)	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

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), 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. Since the number of critical scenarios in samNMPC is much smaller than 81, it is slightly less conservative than the ideal multistage NMPC, while still satisfying 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 6.

Note that here the uncertainty in $E_{A,3}$ and c_{A0} is ±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 3 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 3 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 (29) and solving the approximate problem (19), which are shown separately in Table 3. 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 (29) 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 (31), 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 [13] 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 4 shows the comparison between solving (29) 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 (29).

Table 4: CSTR - Average computational performance in CPUs for solving (29) as large linear system and with Schur complement

N_r	$\lvert\mathbb{C}\rvert$	Solving (29) 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

5.2. Quadtank example

In our second case study, we consider the Quadtank problem [28], with a configuration of four tanks as shown in Figure 7.

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

$$\frac{dx_1}{dt} = -\frac{a_1}{A_1}\sqrt{2gx_1} + \frac{a_3}{A_1}\sqrt{2gx_3} + \frac{\gamma_1}{A_1}u_1 \tag{55a}$$

$$\frac{dx_2}{dt} = -\frac{a_2}{A_2}\sqrt{2gx_2} + \frac{a_4}{A_2}\sqrt{2gx_4} + \frac{\gamma_2}{A_2}u_2 \tag{55b}$$

$$\frac{dx_3}{dt} = -\frac{a_3}{A_3}\sqrt{2gx_3} + \frac{1-\gamma_2}{A_3}u_2 \tag{55c}$$

$$\frac{dx_4}{dt} = -\frac{a_4}{A_4}\sqrt{2gx_4} + \frac{1-\gamma_1}{A_4}u_1 \tag{55d}$$

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 5 and 6.

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 7. The MPC simulation is run for 150 time steps with each time step being $10s$.

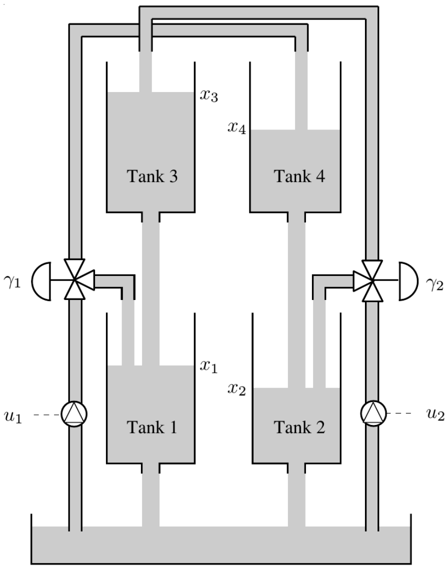

Figure 7: Quadtank schematic [28]

Table 5: 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 6: 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 7: 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

The objective function is formulated as:

$$\varphi_l = (x_{1l} - x_{1s})^2 + (x_{2l} - x_{2s})^2 + r(\Delta u_1^2 + \Delta u_2^2) \tag{56}$$

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. As with the previous case study, we consider that the true realizations of γ_1 and γ_2 take random values from their corresponding {max, nominal, min} values. The resulting plots for an increasing robust horizon are shown in Figures 8, 9 and 10, 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 8), and are reasonably close for $N_r = 2$ (Figure 9). 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 10.

A comparison of the computational times, averaged over 5 random sequences of uncertainty realizations, is shown in Tables 8 and 9. Again, samNMPC significantly outperforms ideal multistage NMPC in terms of speed, without sac-

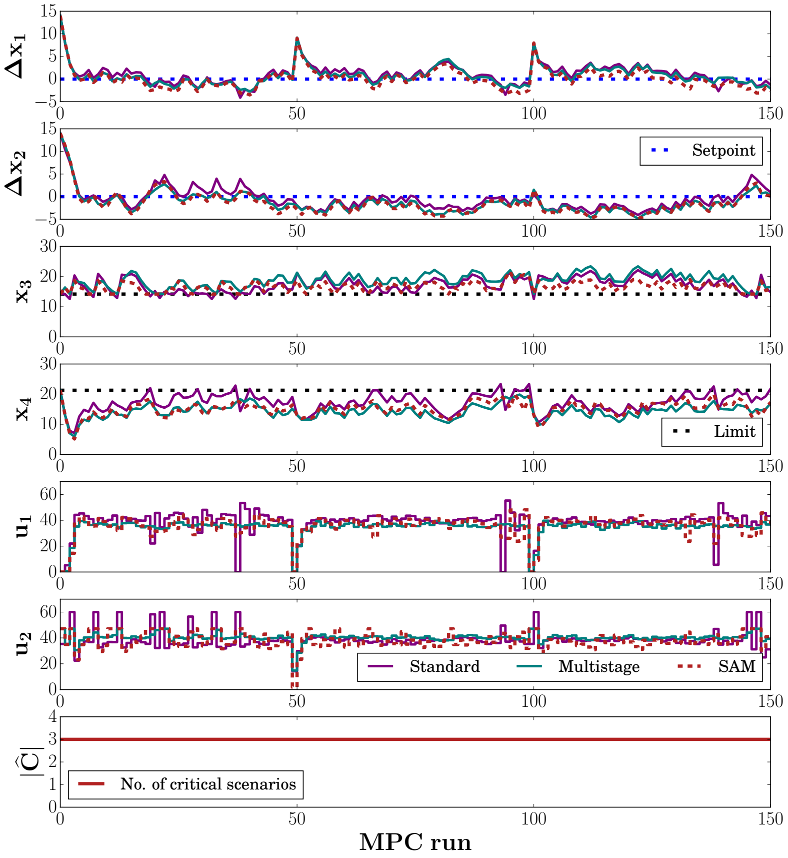

Figure 8: Setpoint tracking (x_1, x_2) and constraint satisfaction (x_3, x_4) for standard, ideal multistage and samNMPC with $N_r = 1$ (9 scenarios).

Table 8: Quadtank - Average computational performance in CPUs. For samNMPC, the times for solving linear system (29) and the NLP (19) are reported separately.

N_r	$\lvert\mathbb{C}\rvert$	Standard	Multistage	SAM - solving (29) / (19)	Avg. $\lvert\widehat{\mathbb{C}}\rvert$
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

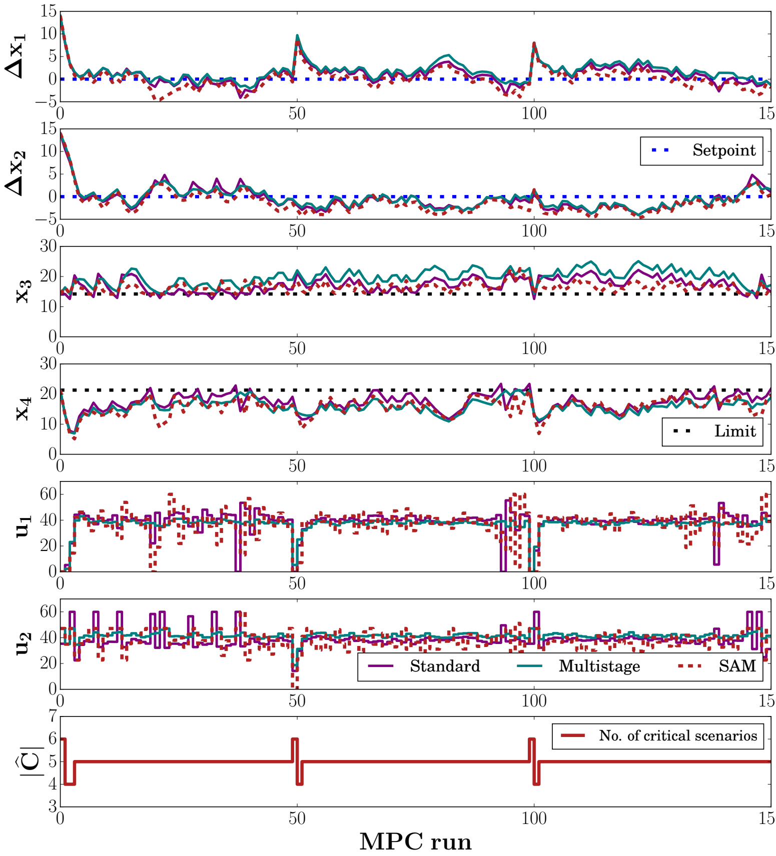

Figure 9: Setpoint tracking (x_1, x_2) and constraint satisfaction (x_3, x_4) for standard, ideal multistage and samNMPC with $N_r = 2$ (81 scenarios).

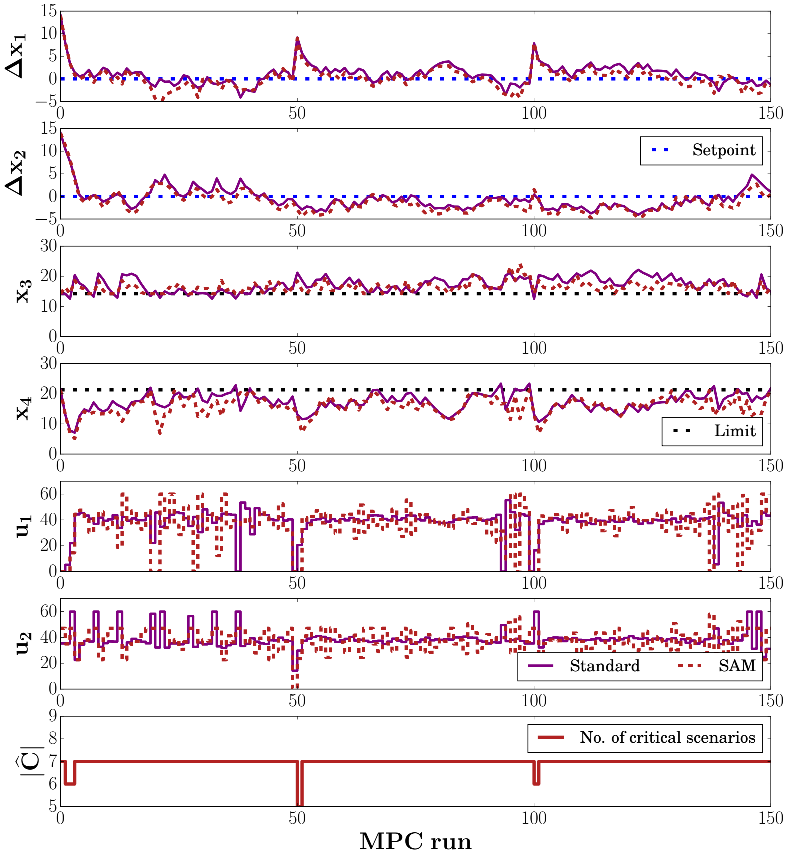

Figure 10: Setpoint tracking (x_1, x_2) and constraint satisfaction (x_3, x_4) for standard, ideal multistage and samNMPC with $N_r = 3$ (729 scenarios).

Table 9: Quadtank - Average computational performance in CPUs for solving (29) as large linear system and with Schur complement.

N_r	$\lvert\mathbb{C}\rvert$	Solving (29) 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

rificing robustness or setpoint tracking under uncertainty. Table 9 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.

6. Conclusion

This work 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.

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.

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 for longer robust horizons where ideal multistage NMPC becomes intractable. The use of parallel Schur complement decomposition can further speed up the solution time.

Future work in this domain includes extending the samNMPC algorithm to 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. An interesting approach would be to use a more accurate path-following sensitivity-based approximation along the lines of [34], instead of computing linear sensitivity steps, for the noncritical scenarios. Another option is to integrate the samNMPC approach with the advanced-step multistage NMPC [39], which would further reduce the computational effort. The computational effort can also be aided by employing decomposition strategies to solve this approximate multistage NMPC problem.

Acknowledgment

The authors greatly appreciate valuable inputs from Zawadi Mdoe, NTNU.

Declaration of Competing Interest

There are no conflicts of interest related to this work.

Funding

This publication has been funded in part 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, the user partners of HighEFF, as well as the Center for Advanced Process Decision-making (CAPD) at Carnegie Mellon University.

[1] Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1), 65–98.

[2] Biegler, L.T., Grossmann, I.E., and Westerberg, A.W. (1997). *Systematic methods for chemical process design*, chapter Process Flexibility. Prentice Hall PTR.

[3] Campo, P.J. and Morari, M. (1987). Robust model predictive control. In *1987 American Control Conference*, 1021–1026.

[4] Daosud, W., Kittisupakorn, P., Fikar, M., Lucia, S., and Paulen, R. (2019). Efficient robust nonlinear model predictive control via approximate multi-stage programming: A neural networks based approach. In A.A. Kiss, E. Zondervan, R. Lakerveld, and L. zkan (eds.), *29th European Symposium on Computer Aided Process Engineering*, volume 46 of *Computer Aided Chemical Engineering*, 1261 – 1266. Elsevier. doi: https://doi.org/10.1016/B978-0-12-818634-3.50211-3.

[5] de Nicolao, G., Magni, L., and Scattolini, R. (1996). On the robustness of receding-horizon control with terminal constraints. *IEEE Transactions on Automatic Control*, 41(3), 451–453. doi:10.1109/9.486649.

[6] Dunning, I., Huchette, J., and Lubin, M. (2017). Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2), 295–320. doi: 10.1137/15M1020575.

[7] Fiacco, A.V. (1976). Sensitivity analysis for nonlinear programming using penalty methods. *Mathematical Programming*, 10, 287 – 311. doi: https://doi.org/10.1007/BF01580677.

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

[9] Gauvin, J. (1977). A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming. *Math. Program.*, 12(1), 136138. doi:10.1007/BF01593777.

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

[11] Holtorf, F., Mitsos, A., and Biegler, L. (2019). Multistage nmpc with on-line generated scenario trees: Application to a semi-batch polymerization process. *Journal of Process Control*, 80, 167–179. doi: 10.1016/j.jprocont.2019.05.007.

[12] Jäschke, J., Yang, X., and Biegler, L.T. (2014). Fast economic model predictive control based on nlp-sensitivities. *Journal of Process Control*, 24, 1260–1272.

[13] Kang, J., Cao, Y., Word, D.P., and Laird, C. (2014). An interior-point method for efficient solution of block-structured nlp problems using an implicit schur-complement decomposition. *Computers & Chemical Engineering*, 71, 563 – 573. doi: https://doi.org/10.1016/j.compchemeng.2014.09.013.

[14] Klatt, K.U. and Engell, S. (1998). Gain-scheduling trajectory control of a continuous stirred tank reactor. *Computers & Chemical Engineering*, 22(4), 491 – 502. doi:https://doi.org/10.1016/S0098-1354(97)00261-5.

[15] Kojima, M. (1980). Strongly stable stationary solutions in nonlinear programs. In S.M. Robinson (ed.), *Analysis and Computation of Fixed Points*, 93 – 138. Academic Press. doi:https://doi.org/10.1016/B978-0-12-590240-3.50009-4.

[16] Krishnamoorthy, D., Foss, B., and Skogestad, S. (2019). A primal decomposition algorithm for distributed multistage scenario model pre-

dictive control. *Journal of Process Control*, 81, 162 – 171. doi: https://doi.org/10.1016/j.jprocont.2019.02.003.

[17] Leidereiter, C., Potschka, A., and Bock, H.G. (2015). Dual decomposition for qps in scenario tree nmpc. In *2015 European Control Conference (ECC)*, 1608–1613. doi:10.1109/ECC.2015.7330767.

[18] Lucia, S. (2014). *Robust Multi-stage Nonlinear Model Predictive Control*. Ph.D. thesis, Technical University of Dortmund.

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

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

[21] Lucia, S., Subramanian, S., Limon, D., and Engell, S. (2020). Stability properties of multi-stage nonlinear model predictive control. *Systems & Control Letters*, 143, 104743. doi: https://doi.org/10.1016/j.sysconle.2020.104743.

[22] Maiworm, M., Bäthge, T., and Findeisen, R. (2015). Scenario-based model predictive control: Recursive feasibility and stability. *IFAC-PapersOnLine*, 48(8), 50–56.

[23] Martí, R., Lucia, S., Sarabia, D., Paulen, R., Engell, S., and de Prada, C. (2015). Improving scenario decomposition algorithms for robust nonlinear model predictive control. *Computers & Chemical Engineering*, 79, 30–45.

[24] Mayne, D.Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, 50(12), 2967 – 2986.

[25] Mayne, D. (2015). Robust and stochastic MPC: Are we going in the right direction? *IFAC-PapersOnLine*, 48(23), 1–8.

[26] Nocedal, J. and Wright, S.J. (2006). *Numerical Optimization.* Springer, New York, NY, USA, second edition.

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

[28] Raff, T., Huber, S., Nagy, Z.K., and Allgower, F. (2006). 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*, 237–242.

[29] Raimondo, D.M., Limon, D., Lazar, M., Magni, L., and Camacho, E.F. (2009). Min-max model predictive control of nonlinear systems: A unifying overview on stability. *European Journal of Control*, 15(1), 5–21.

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

[31] Robinson, S.M. (1980). Strongly regular generalized equations. *Mathematics of Operations Research*, 5(1), 43–62. doi:10.1287/moor.5.1.43.

[32] Scokaert, P. and Mayne, D. (1998). Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic Control*, 43(8), 1136–1142. doi:10.1109/9.704989.

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

[34] Suwartadi, E., Kungurtsev, V., and Jäschke, J. (2017). Sensitivity-based economic nmpc with a path-following approach. *Processes*, 5(4), 8. doi: 10.3390/pr5010008. URL `http://dx.doi.org/10.3390/pr5010008`.

[35] Thombre, M., Mdoe, Z., and Jäschke, J. (2020). Data-driven robust optimal operation of thermal energy storage in industrial clusters. *Processes*, 8(2), 194. doi:10.3390/pr8020194.

[36] Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1), 25–57. doi:10.1007/s10107-004-0559-y.

[37] Yu, S., Reble, M., Chen, H., and Allgower, F. (2014). Inherent robustness properties of quasi-infinite horizon nonlinear model predictive control. *Automatica*, 50(9), 2269 – 2280. doi: https://doi.org/10.1016/j.automatica.2014.07.014.

[38] Yu, Z. (2020). *Advances in Decision-making Under Uncertainty with Nonlinear Model Predictive Control*. Ph.D. thesis, Carnegie Mellon University. doi:10.1184/R1/11604996.v1.

[39] Yu, Z.J. and Biegler, L.T. (2019). Advanced-step multistage nonlinear model predictive control: Robustness and stability. *Journal of Process Control*, 84, 192 – 206. doi:https://doi.org/10.1016/j.jprocont.2019.10.006.

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