Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Sensitivity-assisted Robust Nonlinear Model Predictive Control with Scenario Generation

Zhou (Joyce) Yu, Lorenz T. Biegler

Carnegie Mellon University, Pittsburgh, PA 15206 USA (e-mail: zhouy1, lb01@andrew.cmu.edu).

Abstract: We propose a sensitivity-assisted multistage Nonlinear Model Predictive Control strategy, called *samNMPC*, to address multistage stochastic programs for robust NMPC. Our approach divides the scenario sets in the stochastic programming formulation into critical and noncritical sets. Critical scenarios are selected by scenario generation based on worst-case constraint determination, while stage costs for noncritical scenarios are determined by sensitivity-based approximations. The resulting multi-stage NMPC problem leads to a first order accurate control profile that satisfies all constraints under uncertainty. Moreover, computational costs of this formulation scale independently of the number of disturbance variables, and only linearly with the robust horizon and number of constraints. Our proposed approach is illustrated on a CSTR (continuous stirred tank reactor) case study with two uncertain parameters. Compared to competing approaches, *samNMPC* delivers robust performance of multi-stage NMPC with significantly less computational cost.

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

1. INTRODUCTION

Model predictive control (MPC) is a modern control concept that has been popular in the refinery, chemicals, automotive and aerospace industries (Qin and Badgwell, 2003). Robust MPC has gained much attention over the past two decades, especially for systems that require satisfaction of stability, performance metrics and system constraints under model variations and noise signals (Bemporad and Morari, 1999). To meet both robust stability and performance requirements, Bemporad and Morari classify two ways to design a robust MPC controller: formulating an optimal control objective and uncertainty set that leads to robust stability, or explicitly applying robust contraction constraints to guarantee stability. Min-max MPC (Campo and Morari, 1987) and tube-based MPC (Mayne et al., 2005) follow these two options, respectively. On the one hand, min-max MPC suffers from conservatism due to the rare occurence of the worst case. On the other hand, tube-based MPC needs an ancillary controller that is difficult to compute for nonlinear models.

To design a robust NMPC controller with less conservatism, Lucia et al. (2013) have developed multistage NMPC that shows advantages in constraint handling under the presence of uncertainties. In addition, Yu and Biegler (2019) prove robust stability of multistage NMPC for both ideal and sensitivity-based constructions. However, the optimization problem size increases significantly with respect to the number of uncertain parameters and the length of robust horizon. To ease the online computational stress, decompositions and approximation algorithms have been developed to limit the growth in problem size, while preserving the properties of multistage NMPC. Leidereiter et al. (2015) apply a dual decomposition approach that solves QPs in the inner layer and applies a non-smooth Newton method in the outer layer. Similarly, Krishnamoorthy et al. (2019) develop a primal decomposition algorithm that ensures the satisfaction of non-anticipativity constraints. Alternatively, Daosud et al. (2019) approximate cost-to-go functions of different scenarios by neural networks and apply to a semi-batch reactor. This approach is fast, but requires tuning of the robust horizon. Finally, Holtorf et al. (2019) introduce an online scenario generation method to approximate multistage NMPC with far fewer scenarios, but only by optimizing a worst case cost function.

This study develops a sensitivity-assisted multistage NMPC (*samNMPC*) method that emulates the multistage NMPC with scenario generation, but considers conventional stage costs by performing a NLP sensitivity calculation. Section 2 discusses the detailed algorithm, followed by Section 3, which presents a CSTR example and compares the performance with other state-of-the-art robust NMPC methods. Lastly, Section 4 summarizes the paper and outlines future work.

2. SAM-NMPC ALGORITHM

Consider the discrete-time nonlinear dynamic model:

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

where $x_k \in \mathbb{X} \subset \mathbb{R}^{n_x}, u_k \in \mathbb{U} \subset \mathbb{R}^{n_u}$ are state and control variables at time step k, and $d_k \in \mathbb{D} \subset \mathbb{R}^{n_d}$ represents the time-varying model parameter. As a typical choice, each element of d_k can take three possible values:

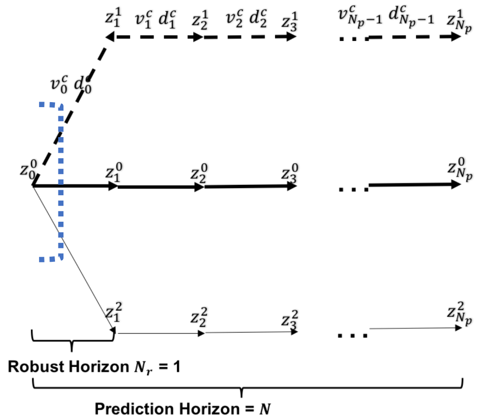

Fig. 1. Typical scenario tree where $n_d = 1, N_r = 1$. Dashed lines show the critical scenario $\widehat{C} = \{1\}$.

{max, nominal, min}. The sensitivity-assisted multistage NMPC (*samNMPC*) is built upon the same scenario tree as in conventional multistage NMPC, but combines with a sensitivity-based approximation algorithm that avoids the exponential growth of the online optimization problem with respect to number of uncertain parameters. We start by describing conventional multistage NMPC.

2.1 Multistage NMPC

Multistage NMPC (Lucia et al., 2013) has been developed at the intersection of stochastic programming and modern control. A scenario tree is formed to represent the state evolutions for all possible uncertain model parameters. In practice, a robust horizon N_r (shorter than prediction horizon N) is also applied to manage a tractable problem size. A scenario tree with robust horizon $N_r = 1$ can be seen as Fig. 1, where z_l^c, v_l^c, d_l^c denote the state, control variables, and parameter at stage l and scenario c, respectively. Note that the dotted bracket depicts the non-anticipativity constraint (NAC). NACs are required within the robust horizon to enforce the same control variable for every scenario originating from the same node (state). To translate Fig. 1 to an optimization problem with prediction horizon N, the following formulation is solved for each horizon k

$$J_N(x_k) = \min_{z_l^c, v_l^c} \quad \sum_{c \in \mathcal{C}} p^c \Big(\phi(z_N^c, d_{N-1}^c) + \sum_{l=0}^{N-1} \varphi(z_l^c, v_l^c, d_l^c) \Big) \tag{1a}$$

$$\text{s.t.} \quad z_{l+1}^c = f(z_l^c, v_l^c, d_l^c) \quad l = 0, ..., N-1 \tag{1b}$$

$$z_0^c = x_k \tag{1c}$$

$$v_l^c = v_l^{c'} \quad \{(c, c') | z_l^c = z_l^{c'}\} \tag{1d}$$

$$d_{l-1}^c = d_l^c \text{ for } l = N_r, \dots N-1 \tag{1e}$$

$$z_l^c \in \mathbb{X}, v_l^c \in \mathbb{U}, z_N^c \in \mathbb{X}_f, d_l^c \in \mathbb{D} \tag{1f}$$

$$\forall c, c' \in \mathcal{C} \tag{1g}$$

where the objective function (1a) contains a weighted terminal costs $\phi(\cdot,\cdot)$ and the sum of stage costs $\varphi(\cdot,\cdot,\cdot)$ over time. (1d) denotes the non-anticipativity constraint (NAC) and (1e) shows the same uncertain parameter values are used for the rest of prediction horizon beyond robust horizon.

2.2 NLP sensitivity properties

To explore sensitivity properties for multistage NMPC optimization problem (1), we apply a barrier NLP solver such as IPOPT (Wächter and Biegler, 2006). We can rewrite (1) as the following generic parametric program:

$$\min_{\mathbf{x}} \quad F(\mathbf{x}; p) \quad \text{s.t.} \quad h(\mathbf{x}, p) = 0, \ \mathbf{x} \geq 0 \tag{2}$$

where the variable vector $\mathbf{x}$ includes all primal variables in (1), and p_0 and p_1 represent the parameter in the current scenario and in perturbed scenarios, respectively. IPOPT handles the inequality constraints implicitly through a barrier function in the objective with parameter μ, and solves the following problem:

$$\min_{\mathbf{x}} \quad F(\mathbf{x}; p) - \mu \sum_{i=1}^{n_{\mathbf{x}}} ln(\mathbf{x}_i) \quad \text{s.t.} \quad h(\mathbf{x}, p) = 0. \tag{3}$$

After solving a sequence of problems (3), with $\mu \to 0$ and $p = p_0$ the solutions of (3) approach $\mathbf{x}^* = \mathbf{x}(p_0)$, the solution of (2). To see how $\mathbf{x}^*$ varies with respect to perturbations of p, we cite the following property:

Theorem 1. (NLP Sensitivity) (Fiacco, 1976, 1983). If $f(\cdot,\cdot)$, $\varphi(\cdot,\cdot)$ and $\phi(\cdot)$ of the parametric NLP problem (2) are twice continuously differentiable in a neighborhood of the nominal (primal and dual) solution $s^*(p_0)$ and this solution satisfies the linear independence constraint qualifications (LICQ), strong second order sufficient conditions (SSOSC) and strict complementarity (SC), then the solution $s^*(p_0)$ is differentiable in p.

Moreover, for $\mu > 0$ but negligibly small, the primal-dual optimality conditions (i.e. KKT conditions) of (3) are solved directly at p_0,

$$\begin{aligned} &\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \lambda^*, \upsilon^*; p_0) \\ &= \nabla_{\mathbf{x}} F(\mathbf{x}^*; p_0) + \nabla_{\mathbf{x}} h(\mathbf{x}^*; p_0) \lambda^* - \upsilon^* = 0 \\ &h(\mathbf{x}^*; p_0) = 0, \quad \mathbf{X}^* \mathbf{V}^* e = \mu e \end{aligned} \tag{4}$$

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

$$s(\mu; p)^T = [\mathbf{x}(\mu; p)^T, \lambda(\mu; p)^T, \upsilon(\mu; p)^T].$$

Applying Theorem 1 and the implicit function theorem to differentiate (4) leads to the following linear system for sensitivity of s.

$$\mathcal{M}(s(\mu; p_0)) \Delta s = -\mathcal{N}(s(\mu; p_0); p) \tag{5}$$

where

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

is the KKT matrix, and

$$\mathcal{N}(s(\mu; p_0); p)^T = [\nabla_{\mathbf{x}} \mathcal{L}(s(\mu; p_0); p)^T, \ h(\mathbf{x}(\mu; p_0); p)^T, \ 0]$$

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 sensitivities Δs can be computed as $\Delta s = -\mathcal{M}(s(\mu; p_0))^{-1} \mathcal{N}(s(\mu; p_0); p)$ using a cheap backsolve if the factorized form of $\mathcal{M}(s(\mu; p_0))$ is available.

Applying NLP sensitivity to multistage NMPC problem (1), one can rewrite (5) in the following block-bordered-diagonal (BBD) form,

$$\begin{bmatrix} \mathbf{K}_0 & & \dots & \mathbf{N}_0 \\ & \mathbf{K}_1 & \dots & \mathbf{N}_1 \\ \vdots & \vdots & \ddots & \vdots \\ & & \mathbf{K}_{\bar{c}} & \mathbf{N}_{\bar{c}} \\ \mathbf{N}_0^T & \mathbf{N}_1^T & \dots \mathbf{N}_{\bar{c}}^T & \end{bmatrix} \begin{bmatrix} \Delta \mathbf{s}_0 \\ \Delta \mathbf{s}_1 \\ \vdots \\ \Delta \mathbf{s}_{\bar{c}} \\ \gamma \end{bmatrix} = - \begin{bmatrix} r_0 \\ r_1 \\ \vdots \\ r_{\bar{c}} \\ 0 \end{bmatrix} \quad (6)$$

where $\bar{c} = |\mathcal{C}|$, $\mathbf{K}_c = \begin{bmatrix} \mathbf{W}_c & \mathbf{A}_c \\ \mathbf{A}_c^T & 0 \end{bmatrix}$, $\Delta \mathbf{s}_c = \begin{bmatrix} \Delta \mathbf{x}^c \\ \Delta \lambda^c \end{bmatrix}$, $r_c^T = [\nabla_{\mathbf{x}^c} \mathcal{L}(\mathbf{x}^c, d^c)^T,\ h(\mathbf{x}^c, d^c)^T]$ for each $c \in \mathcal{C}$. $\mathbf{W}_c = \nabla_{\mathbf{x}^c \mathbf{x}^c} \mathcal{L}(\mathbf{x}^c, d^c) + X_c^{-1} V_c$ is the augmented Hessian for scenario c. $\mathbf{x}^c = [z_0^c, v_0^c, z_1^c, v_1^c, ..., z_{N-1}^c, v_{N-1}^c, z_N^c]^T$ denotes all primal variables associated with scenario c, and λ^c for all multipliers associated with scenario c.

In (6), $\mathbf{N}_c$ represents the NAC constraint that contains scenario c, where $\mathbf{N}_c = [\widetilde{\mathbf{N}}_c,\ 0]^T \in \mathbb{R}^{n+m_c} \times \mathbb{R}^{m_{NAC} * n_u}$ and n and m_c are the number of primal variables and constraints, respectively, in each scenario, and m_{NAC} is the number of NAC constraints. $\mathbf{N}_c$ 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 (6) is the multiplier associated with NAC (1d) with the dimension $\gamma \in \mathbb{R}^{m_{NAC} * n_u}$.

The linear system (6) can be solved with the Schur Complement,

$$\sum_{c \in \mathcal{C}} (\mathbf{N}_c^T \mathbf{K}_c^{-1} \mathbf{N}_c) \gamma = - \sum_{c \in \mathcal{C}} (\mathbf{N}_c^T \mathbf{K}_c^{-1} r_c) \quad (7)$$

$$\mathbf{K}_c \Delta \mathbf{s}_c = -(r_c + \mathbf{N}_c \gamma),\ \forall c \in \mathcal{C} \quad (8)$$

where $\Delta \mathbf{s}_c$ is the sensitivity with scenario $c \in \mathcal{C}$, which can then be used to calculate a perturbed solution for that scenario as $\tilde{\mathbf{s}}_c(p) = \mathbf{s}_c(p_0) + \Delta \mathbf{s}_c$.

The structure of multistage NMPC problems can be exploited to obtain a fast approximate solution with respect to perturbed parameters. Solving the BBD linear system with the help of Schur-complement can be two orders of magnitude faster than solving a NLP problem. Furthermore, it would expedite the process even more if the factorization of the KKT matrix in (6) can be obtained. The next section discusses an inexpensive way to do this by solving the nominal NMPC problem.

2.3 Nominal NMPC

Standard NMPC (or nominal NMPC) considers only the nominal model in the controller, and it solves the following single scenario problem:

$$\min_{z_l, v_l} \quad \phi(z_N, \bar{d}_{N-1}) + \sum_{l=0}^{N-1} \varphi(z_l, v_l, \bar{d}_l) \quad (9a)$$

$$\text{s.t.} \quad z_{l+1} = f(z_l, v_l, \bar{d}_l) \qquad l = 0, ..., N-1 \quad (9b)$$

$$z_0 = x_k \quad (9c)$$

$$z_l \in \mathbb{X}, v_l \in \mathbb{U}, z_N \in \mathbb{X}_f \quad (9d)$$

Problem (9) has a smaller problem size than Problem (1); it is more computationally efficient, but it may violate constraints and perform poorly due to plant-model mismatch (Yu and Biegler, 2019).

From the standard NMPC problem (9), one obtains the NLP sensitivity from $\mathbf{K}_0 \Delta \mathbf{s}_0 = -r_0$, where $\Delta \mathbf{s}_0^T = [(\Delta \mathbf{x}^0)^T,\ (\Delta \lambda^0)^T]$, $r_0 = [\nabla_{\mathbf{x}^0} \mathcal{L}(\mathbf{x}^0, d^0)^T,\ h(\mathbf{x}^0, d^0)^T]$. From the nominal NMPC problem, the solution to (9) can be used to form the sensitivity system (6). If we substitute the approximation $\mathbf{K}_c = \mathbf{K}_0$ in (6), we can then solve for the approximate sensitivity solution $\Delta \mathbf{s}_c^T = [(\Delta z^c)^T,\ (\Delta v^c)^T,\ (\Delta \lambda^c)^T] = [(\bar{z}^c - z^0)^T,\ (\bar{v}^c - v^0)^T,\ (\bar{\lambda}^c - \lambda^0)^T]$, which provides perturbed solutions for all scenarios $\tilde{\mathbf{s}}(p)$. This computation is particularly efficient since the KKT matrix of the nominal NMPC problem ($\mathbf{K}_0$) has already been factored and reused in (7) and (8).

2.4 Critical scenarios

The remaining feature of the algorithm deals with feasible performance under uncertainty. Robustness of multistage NMPC is often enforced by considering scenarios with extreme-value parameters (i.e. max or min). If one can predict which parameter values are likely to violate constraints, then the scenarios associated with these values should be treated differently in (1) than in scenarios where the chance of violating constraints is low. We call the former *critical scenarios*, and the latter *non-critical scenarios*. To determine critical scenarios we apply the approach in Holtorf et al. (2019) to the following dynamic model and constraints:

$$x_{k+1} = f(x_k, u_k, d_k), x_k \in \mathbb{X}, u_k \in \mathbb{U}, \quad (10)$$

and discretize the uncertainty description (i.e. {max, nominal, min}) to develop a scenario tree. At the current state of the plant x_k, critical scenarios are then determined by solving the following optimization problem:

$$\max_{d_l} \quad g_j(z_l, v_l, d_l) \quad (11)$$

$$s.t. \ z_{l+1} = f(z_l, v_l, d_l), \quad l = 0, \dots, N-1$$

$$z_0 = x_k$$

where the inequality constraints $g_j(\cdot, \cdot, \cdot)$ represent the state variable bounds $x \in \mathbb{X}$ and j denotes the index of inequality constraints. If a fixed trajectory of $(z_l, v_l)_{l=0,...,N-1}$ is used, and monotonicity is assumed, i.e., the reduced gradients $dg_j/d(d_l)$ always have the same sign, then the solution of (11) is easily determined by setting d_l to appropriate upper or lower bounds for each stage l and each inequality constraint g_j. Based on a reference trajectory for $(z_l, v_l)_{l=0,...,N-1}$, one decides the critical value (max or min) of each uncertain parameter based on the following criteria: For $l \in N_r$, and $m = 1, ..., n_d$

$$\begin{aligned} d_{l,m}^{wc} &= \arg\max_{d \in \mathbb{D}} \nabla_d g_j|_{l,(z_l,v_l,d_l)|_{ref}}^T d_l \\ &= \begin{cases} d_{l,m}^{min}, \text{if } \dfrac{d(g_j)}{d(d_m)}|_{l,(z_l,v_l,d_l)|_{ref}} \le 0 \\ d_{l,m}^{max}, \text{otherwise} \end{cases} \end{aligned} \quad (12)$$

Note the number of critical scenarios is bounded by the number of active inequality constraints over the horizon, which is generally much smaller than the full-size multistage tree, which consists of $3^{n_d N_r}$ scenarios (Holtorf et al., 2019).

We then classify the set of scenarios $\mathcal{C} = \{0\} \cup \widehat{C} \cup \bar{C}$, as $\{0\}$ - the nominal scenario; $\widehat{C}$ - critical scenarios where active inequalities are encountered; $\bar{C}$ - non-critical scenarios with the primal variables $(\bar{z}_l^c, \bar{v}_l^c)$ determined only from NLP sensitivity, where $g_j(\bar{z}_l^c, \bar{v}_l^c) < 0$ is expected

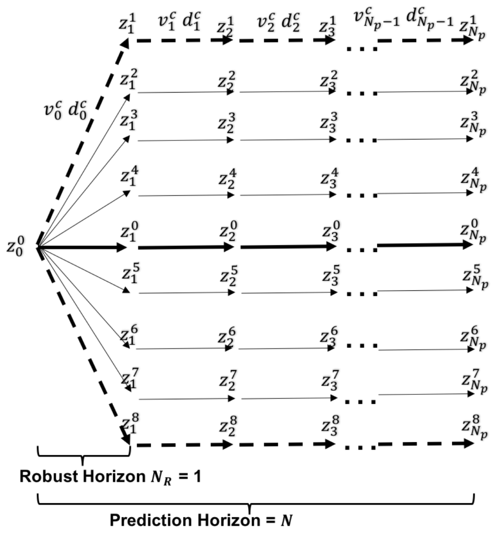

Fig. 2. Scenario tree with $n_d = 2, N_r = 1$. Critical scenarios $\widehat{C} = \{1, 8\}$ are shown with dashed lines.

to hold. Figs. 1 and 2 illustrate how the nominal and critical scenarios appear as part of the full multistage scenario tree. A key feature of this framework is that critical scenarios are dynamically updated for each horizon k. More information on the scenario generation approach can be found in Holtorf et al. (2019).

- If constraint g_j is insensitive to uncertainty parameter d_m, i.e. $|\frac{d(g_j)}{d(d_m)}| < \epsilon$, then critical scenarios associated with this constraint need not be considered for d_m.
- At each time step k in NMPC, active constraints are reevaluated at the updated trajectory (z_l, v_l), together with corresponding worst case parameter values for that constraint.
- Based on an active constraint g_j, a set of worst case parameter values d^{wc} is selected, and the corresponding critical scenarios $c \in \widehat{C}$ are included in NLP (13), while non-critical scenarios $c \in \bar{C}$ are approximated from the sensitivity calculation in (6).

2.5 Approximate multistage problem

The resulting approximate problem is given by

$$\min_{z_l, v_l} \sum_{c \in \widehat{C} \cup \{0\}} (\phi(z_N^c, d_{N-1}^c) + \sum_{l=0}^{N-1} \varphi(z_l^c, v_l^c, d_l^c)) \; +$$

$$\sum_{c \in \bar{C}} (\phi(z_N^0 + \Delta z_N^c, d_{N-1}^c) + \sum_{l=0}^{N-1} \varphi(z_l^0 + \Delta z_l^c, v_l^0 + \Delta v_l^c, d_l^c))$$

$$\text{s.t.} \quad z_{l+1}^c = f(z_l^c, v_l^c, d_l^c) \qquad c \in \widehat{C} \cup \{0\}, \; l = 0, ..., N-1$$

$$z_0^c = x_k \qquad c \in \widehat{C}$$

$$v_l^c = v_l^{c'} \quad \{(c, c') | z_l^c = z_l^{c'}\} \quad c, c' \in \widehat{C} \cup \{0\}$$

$$z_l^c \in \mathbb{X}, v_l^c \in \mathbb{U}, z_N^c \in \mathbb{X}_f \qquad (13)$$

where $\widehat{C}$ and $\bar{C}$ are critical and non-critical scenarios, respectively, and $\Delta z_l^c, \Delta v_l^c$ are obtained from the sensitivity step (6) based on the nominal scenario. The NLP problem (13) can be considered as a partially linearized version of problem (1), where non-critical scenarios are no longer constrained by the equations. Instead, the state and control variables of non-critical scenarios through sensitivities in their stage costs calculated from (8) and from NACs that are still satisfied in (6). In this way, non-critical scenarios only appear in the objective function and do not increase the number of variables and constraints in the NLP problem. Nevertheless, the smaller NLP (13) still finds first order accurate solutions for multistage NMPC.

2.6 The overall approach for samNMPC

At time step k,

(1) Solve Problem (9) to get the nominal solution and evaluate $\mathbf{K}_0$.
(2) Solve (6) using $\mathbf{K}_c = \mathbf{K}_0$ to get Δz^c and Δv^c.
(3) Find the worst case d^{wc} from (12) and form the critical scenario set $\widehat{C}$
(4) Solve (13) with explicit critical scenarios, and non-critical scenarios represented by sensitivity solutions.
(5) Set $u(k) = v_0^c, \; c \in \widehat{C} \cup \{0\}$ and inject into the plant
(6) Set $k = k + 1$, and go to Step 1

2.7 Implementation

The *samNMPC* method is implemented in CasADi (Andersson et al., 2019). At each step, after solving the nominal NLP using IPOPT (Wächter and Biegler, 2006), the optimal nominal solution is stored as well as the factorization of the nominal KKT matrix $\mathbf{K}_0$. In addition, sparse matrices $\mathbf{N}_c, c \in \mathcal{C}$ are automatically generated based on the number of NACs. The factorization of $\mathbf{K}_0$ can be reused to solve (7) and (8). Using Schur-complement decomposition avoids forming a large linear system, by decoupling the scenarios instead in a parallelizable manner. The linear systems incurred by sensitivity calculations are solved by MA27 from the Harwell Subroutine Library (*http://www.hsl.rl.ac.uk*) and the interface to MA27 is provided by CasADi under linear solver class *linsol*.

3. CASE STUDIES

The scenario generation algorithm has been applied to a benchmark CSTR example from Klatt and Engell (1998).

$$\begin{aligned}
\frac{dc_A}{dt} &= F(c_{A0} - c_A) - k_1 c_A - k_3 c_A^2 \\
\frac{dc_B}{dt} &= -F c_B + k_1 c_A - k_2 c_B \\
\frac{dT_R}{dt} &= F(T_{in} - T_R) + \frac{k_W A}{\rho c_p V_R}(T_K - T_R) - \\
&\quad \frac{k_1 c_A \Delta H_{AB} + k_2 c_B \Delta H_{BC} + k_3 c_A^2 \Delta H_{AD}}{\rho c_p} \\
\frac{dT_K}{dt} &= \frac{1}{m_K c_{pK}}(\dot{Q}_K + k_W A(T_R - T_K))
\end{aligned} \qquad (14)$$

The system has four states $[c_A, c_B, T_R, T_k]$ and two controls $[F, \dot{Q}_K]$. The control objective is to track the setpoint of c_B as $c_B = 0.5 mol/L$ in the first 20 steps, and $c_B = 0.7 mol/L$ for the rest. The stage cost is computed as $\varphi = \sum_l (c_{B,l} - c_B^{ref})^2 + r_1 * (F_l - F_{l-1})^2 + r_2 * (\dot{Q}_{K,l} - \dot{Q}_{K,l-1})^2$. The uncertainty parameters in this case study are $E_{A,3}$ and c_{A0}. Note that the uncertain parameter may change between time steps, but can only choose among a finite set of three values. Also, $N_r = 1$ for all cases in this study. Larger robust horizons will be considered in future work.

The inequalities for this problem are the variable bounds for the state and control variables, and the bounds on control variables are hard constraints. From sensitivity analysis (11) for each state bound, we observe that c_B and T_K are insensitive to both uncertainties $E_{A,3}$ and c_{A0}, which means that the perturbation of both parameters will not affect the value of c_B and T_K (within the robust horizon). On the other hand, for the first step only, c_A is sensitive to c_{A0} and T_R is sensitive to both $E_{A,3}$ and c_{A0}. This implies that when $E_{A,3}$ is uncertain, the worst $E_{A,3}$ value is determined by the sign of $\frac{dT_R}{d(E_{A,3})}|_{(x_l,u_l,d_l^0)}$. When c_{A0} is uncertain, the worst parameter value is determined by the signs of both $\frac{dc_A}{d(c_{A0})}|_{(x_l,u_l,d_l^0)}$ and $\frac{dT_R}{d(c_{A0})}|_{(x_l,u_l,d_l^0)}$.

3.1 Single uncertain parameter: $E_{A,3} \pm 10\%$

For this case, the only uncertain parameter in the system is the activation energy $d = E_{A,3}$ and its three possible values are $\{\max, \text{nom}, \min\}$. From the sensitivity analysis we know that the only affected state is T_R and the sensitivity $\frac{dT_R}{d(E_{A,3})}|_{(x_l,u_l,d_l^0)}$ is computed to determine the worst case parameter value. From the case study, the sign of $\frac{dT_R}{d(E_{A,3})}|_{(x_l,u_l,d_l^0)}$ is usually negative, which renders the worst case value for $d^{wc} = E_{A,3}^{min}$. In this case, scenario generation multistage has only two scenarios (instead of three scenarios in conventional multistage and min-max problems).

Table 1 provides an overall performance comparison for different robust NMPC schemes by averaging 10 random parameter realizations in the plant. The objective function (Obj.) is the sum of the stage costs over time. The CPU seconds are recorded as wall time for each algorithm. Fig. 3 plots trajectories of state and control variables, and objective functions for different NMPC controllers in one sample run.

Fig. 3 shows a close resemblance of tracking performance between multistage and sensitivity-assisted multistage (*samNMPC*), especially in terms of state trajectories and objective values where two lines overlap almost entirely. This observation is also supported by the similar objectives (sum of stage costs) for conventional and approximate multistage in Table 1 for the one parameter case. At the same time, *samNMPC* requires less computational than conventional multistage NMPC. In particular, the sensitivity step, obtained by solving a linear system, only comprises 3% of the total computational time and presents a very light computational footprint compared to solving NLPs.

For the other two control schemes, nominal NMPC seems to have a lower tracking error on average along with fast computing, but under uncertainty, it often suffers non-robust control performance and constraint violations. For instance, Fig. 3 shows that both c_A and T_R violate the upper bound of states constraints. On the other hand, min-max NMPC is able to stay feasible, but performs more

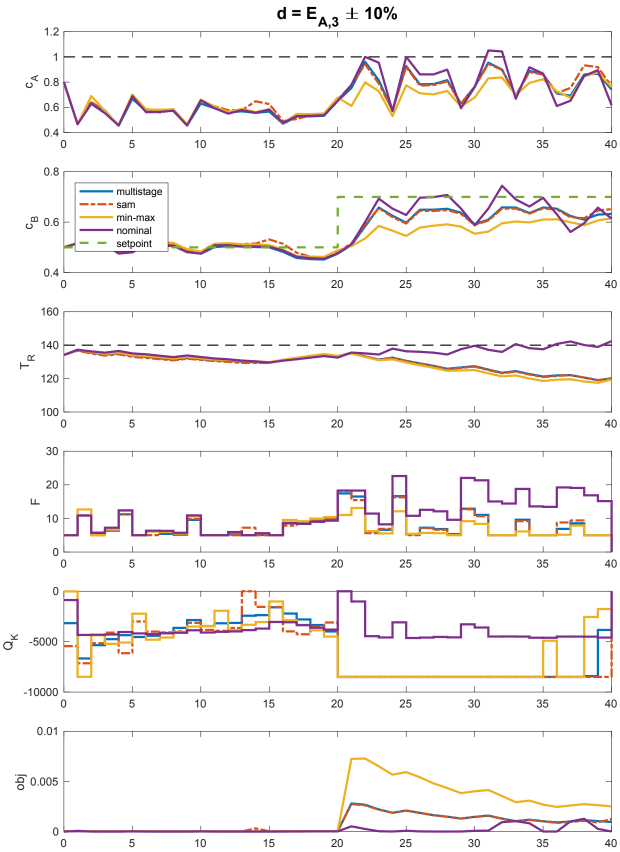

Fig. 3. Trajectories for state and controls for different robust NMPC schemes with $d = E_{A,3}$.

conservatively, which is also corroborated by the objective (stage cost) trajectories shown in Fig. 3.

Table 1. Average performance (10 random runs) of robust NMPC schemes with robust horizon $N_r = 1$, $d = [E_{A,3} \pm 10\%]$ (top) and $d = [c_{A0} \pm 30\%, E_{A,3} \pm 10\%]$ (bottom).

		min-max	multistage	nominal	*samNMPC*
3c	Obj.	0.08467	0.03080	0.00655	0.03085
	CPUs	0.367	0.251	0.0611	0.178
9c	Obj.	0.13794	0.04223	0.01876	0.01679
	CPUs	0.839	0.926	0.0806	0.366

3.2 Two uncertain parameters: $E_{A,3} \pm 10\%, c_{A,0} \pm 30\%$

For this section, both uncertainties are considered, which renders a scenario tree of 9 scenarios. Similarly, the sensitivity analysis leaves the problem size of sensitivity-assisted multistage (*samNMPC*) the same as with 3 scenarios. For the min-max and multistage formulation, the problem size is as large as 6500 variables due to the consideration of all 9 scenarios, so that the computational time is relatively large.

For the case of two uncertain parameters, *samNMPC* and conventional multistage perform closely in terms of state and control trajectories. Most importantly, *samNMPC* achieves a robust tracking performance with only a fraction of computational resources (again, linear algebra

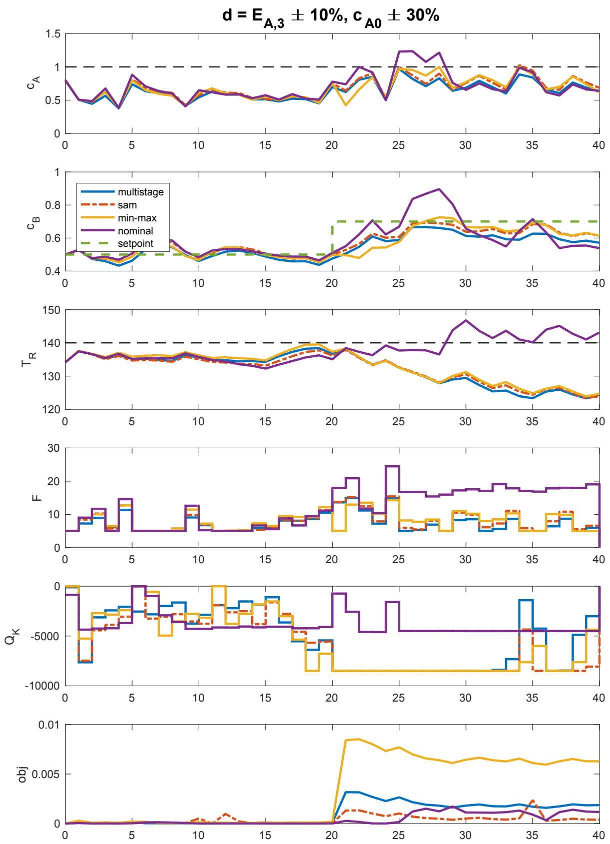

Fig. 4. Trajectories for state and controls for different robust NMPC schemes with $d = E_{A,3}, c_{A0}$.

calculations only cost 0.023 second, which contributes to less than 10% of the total computational time).

The performance comparison of the four NMPC controllers can be seen in Figure 4. Not surprisingly, nominal NMPC suffers large constraint violations, while min-max NMPC delivers a conservative control policy with constraints satisfied. Better performance can be seen from multi-stage NMPC and *samNMPC* because the scenarion tree allows recourse control variables. In fact, the random sampling of input disturbances allows *samNMPC* to report an even better objective than multi-stage NMPC.

4. CONCLUSIONS

We construct a sensitivity-assisted multistage NMPC (*samNMPC*) strategy that closely approximates robust multi-stage NMPC performance but requires far less computation. The conventional multi-stage NMPC problem size grows exponentially with respect to the number of uncertain parameters and robust horizon; hence it becomes difficult to solve online. By separating all scenarios into critical and non-critical categories, one limits the growth of the optimization problem. Only critical scenarios are added to the problem and non-critical scenarios are represented by sensitivity corrections from the nominal scenario. In this way, the size of the NLP is determined by the number of critical scenarios, which is bounded by the number of inequalities and the robust horizon length. Thus, the NLP (13) is much smaller than the traditional multistage formulation (1). We apply this approximation strategy to a CSTR benchmark problem, where two case studies with one and two uncertain parameters are explored. *samNMPC* achieves robustness and similar tracking performance with respect to conventional multistage NMPC, but with only a fraction of computational effort. Future research directions include extending the approximate multistage NMPC algorithm to many more uncertainty parameters, where conventional multistage NMPC becomes intractable.

REFERENCES

Andersson, J.A., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2019). Casadi: a software framework for nonlinear optimization and optimal control. *Math. Prog. Comp.*, 11(1), 1–36.

Bemporad, A. and Morari, M. (1999). Robust model predictive control: A survey. In *Robustness in identification and control*, 207–226. Springer.

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

Daosud, W., Kittisupakorn, P., Fikar, M., Lucia, S., and Paulen, R. (2019). Efficient robust nonlinear model predictive control via approximate multi-stage programming. In *Computer Aided Chemical Engineering*, volume 46, 1261–1266. Elsevier.

Fiacco, A.V. (1976). Sensitivity analysis for nonlinear programming using penalty methods. *Math. Prog.*, 10(1), 287–311.

Fiacco, A.V. (1983). *Introduction to sensitivity and stability analysis in nonlinear programming.* Elsevier.

Holtorf, F., Mitsos, A., and Biegler, L.T. (2019). Multi-stage NMPC with on-line generated scenario trees. *J. Proc. Cont.*, 80, 167–179.

Klatt, K.U. and Engell, S. (1998). Gain-scheduling trajectory control of a continuous stirred tank reactor. *Comp. Chem. Eng.*, 22, 491–502.

Krishnamoorthy, D., Foss, B., and Skogestad, S. (2019). A primal decomposition algorithm for distributed multistage scenario model predictive control. *J. Proc. Cont.*, 81, 162–171.

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. IEEE.

Lucia, S., Finkler, T., and Engell, S. (2013). Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty. *J. Proc. Cont.*, 23(9), 1306–1319.

Mayne, D.Q., Seron, M.M., and Raković, S. (2005). Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2), 219–224.

Qin, S.J. and Badgwell, T.A. (2003). A survey of industrial model predictive control technology. *Control engineering practice*, 11(7), 733–764.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Prog.*, 106(1), 25–57.

Yu, Z.J. and Biegler, L.T. (2019). Advanced-step multistage NMPC: robustness and stability. *J. Proc. Cont.*, 84, 192–206.