Improving Scenario Decomposition for Multistage MPC using a Sensitivity-based Path-following Algorithm

Dinesh Krishnamoorthy, Eka Suwartadi, Bjarne Foss, Sigurd Skogestad and Johannes Jäschke*

***Abstract*—This paper proposes a computationally efficient algorithm for robust multistage model predictive control (MPC). In multistage scenario MPC, the evolution of uncertainty in the prediction horizon is represented via a scenario tree. The resulting large-scale optimization problem can be decomposed into several smaller subproblems where, for example, each subproblem solves a single scenario. Since the different scenarios differ only in the uncertain parameters, the distributed scenario MPC problem can be cast as a parametric nonlinear programming (NLP) problem. By using the NLP sensitivity, we do not need to solve all the subproblems as full NLPs. Instead they can be solved exploiting the parametric nature by a path-following predictor-corrector algorithm that approximates the NLP. This results in a computationally efficient multistage scenario MPC framework. Simulation results show that the sensitivity-based distributed multistage MPC provides a very good approximation of the fully centralized scenario MPC.**

***Index Terms*—Stochastic optimal control, Optimization, Uncertain systems, Predictive control for nonlinear systems**

I. INTRODUCTION

MODEL Predictive Control (MPC) under uncertainty has been receiving significant attention recently, and several different approaches have been proposed to handle the uncertainty. One such approach is the multistage scenario MPC, which was introduced as "feedback min-max MPC" in [1] and later developed further for nonlinear systems as "multistage MPC" in [2], which will be the main focus of this paper. In multistage scenario MPC, the uncertainty set is sampled to obtain a finite number of realizations of the uncertain parameters, and the evolution of the uncertainty in the prediction horizon is represented via a scenario tree. The notion of feedback is then explicitly considered by allowing the different optimal control trajectories to vary for each scenario (closed-loop optimization).

It is important to note that the multistage scenario MPC considered in this work must not be confused with other scenario-based MPC approaches proposed in [3], [4], [5] etc. One of the main difference between these approaches and the multistage MPC used in this work is that, they compute a single control trajectory over all the scenarios. Hence, there is no notion of feedback in the optimization problem (open-loop optimization with closed loop implementation). In contrast, the multistage MPC approach computes different control trajectories for different scenarios subject to the non-anticipativity constraints (closed-loop optimization with closed-loop implementation) [1], [2]. The authors in [6] and [1] also noted that in the presence of uncertainty, a better strategy is to optimize over different control trajectories rather than a single control trajectory.

However, the main drawback of the multistage scenario MPC applied here, is the computational cost. The problem size grows exponentially as the

- number of uncertain parameters increases,
- number of finite realizations of the uncertainty increases,
- length of the prediction horizon increases.

Solving large nonlinear optimization problems can thus be prohibitively expensive in many applications. One way to address this issue is by blocking the uncertainty evolution after a certain number of samples (known as robust horizon) in the prediction horizon as described and justified in [2].

Another way to address the issue of computational cost is by solving a distributed scenario optimization problem. Different scenario decomposition approaches were proposed recently to exploit the fact that each scenario in the scenario tree is an independent problem except for the non-anticipativity constraints, which couples the different scenarios. Hence the different subproblems can be solved independently (in parallel), and a master problem can be used to co-ordinate the different scenarios iteratively.

Dual decomposition strategies for distributed multistage scenario optimization were presented in [7], where the individual subproblems are solved by relaxing the non-anticipativity constraints. The master problem iteratively adjusts the Lagrange multiplier (the dual variables) to co-ordinate the different scenarios. The non-anticipativity constraints are only satisfied upon convergence of the dual master variable. Such methods can however, take relatively large number of iterations to converge and hence cannot be implemented in real time. In such cases, the authors in [8] proposed to use an aggregated variable such that the worst-case constraint violation for the individual scenarios are minimized.

Recently, we proposed a primal decomposition approach [9] which, unlike the dual decomposition approaches, ensures that the non-anticipativity constraints are always feasible throughout the master problem iterations, since it produces a pri-

The authors gratefully acknowledge the financial support from SUBPRO, which is financed by the Research Council of Norway, major industry partners and NTNU. E.S. and J.J acknowledges support from IKTPLUSS Young Researcher Grant.

* Corresponding author. The authors are with the Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), 7491, Trondheim, Norway. dinesh.krishnamoorthy@ntnu.no, eka.suwartadi@ntnu.no, bjarne.foss@ntnu.no, skoge@ntnu.no, johannes.jaschke@ntnu.no

mal feasible solution with monotonically decreasing objective value at each iteration [10]. This is an attractive property for closed-loop implementation of MPC.

To this end, both the dual and the primal decomposition approaches involve solving each scenario independently and a master problem co-ordinates the different scenarios. Although performance improvements have been reported by decomposing the scenario decomposition approaches in [7],[8] and [9], it still requires solving a nonlinear programming problem (NLP) for each scenario. Even with today's computing power, solving nonlinear dynamic optimization problems online can be computationally intensive.

In this paper, we propose to further improve the scenario decomposition algorithms by using NLP sensitivity-based path-following approaches [11], [12]. Since the scenarios differ only in the uncertain parameters, we propose to re-cast the scenario decomposition problem in the framework of parametric NLP. Here, using the solution of one full NLP, the subsequent scenario subproblems are solved by tracking the optimal path along the parameter range that leads to the scenarios by a sequence of predictor-corrector QPs. In simple terms, each of these QPs tells us how the optimal solution changes when the parameter changes by a small value, and the NLP solution change for a larger parameter change can be found by solving several QPs. We apply this idea to the distributed multistage scenario MPC problem to compute how the optimal solution changes from one scenario to the other.

The main contribution of this paper is the use of an NLP sensitivity-based path-following method to efficiently solve the distributed multistage scenario MPC algorithm. The main result is presented as a Corollary of Theorem 1 and Algorithm 1. To the best of our knowledge, parametric optimization concepts have not been used previously to solve scenario decomposition problems for multistage scenario MPC.

The reminder of the paper is organized as follows. We briefly introduce the multistage scenario MPC problem in Section II. In Section III, we present the sensitivity-based distributed scenario MPC with a path-following algorithm (main result). The proposed method is demonstrated on a chemical reactor case example in Section IV.

II. MULTISTAGE SCENARIO MPC

A. Centralized Approach (C_{NLP})

Consider a discrete-time nonlinear system

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

where $\mathbf{x} \in \mathbb{R}^{n_x}$ represents the states, $\mathbf{u} \in \mathbb{R}^{n_u}$ represents the control inputs and $\boldsymbol{p} \in \mathbb{R}^{n_p}$ represents the uncertain parameters. Further, $\mathbf{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_x}$ represents the system equations. We assume that the uncertain parameters have a known distribution $\boldsymbol{p} \in \mathcal{U}$. To account for the uncertainty, the uncertainty space $\mathcal{U}$ is discretized to get M discrete realizations of (1) at each time sample. A scenario tree is then generated that represents the evolution of the uncertainty in the prediction horizon. The path from the root node to a leaf node is called a scenario. The branching at each time step results in exponential growth of the number of scenarios, resulting in a very large optimization problem. For that reason, the branching is often stopped after a certain time period, known as robust horizon N_r, after which the uncertain parameters as treated as constants. Accordingly, we then have $S = M^{N_r}$ scenarios.

The resulting dynamic optimization problem is then written as,

$$\min_{\mathbf{x}_{k,j}, \mathbf{u}_{k,j}} \sum_{j=1}^{S} \left[\sum_{k=0}^{N-1} \mathbf{J}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}) \right] \tag{2a}$$

$$\text{s.t} \quad \mathbf{x}_{k+1,j} = \mathbf{f}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j}) \tag{2b}$$

$$\mathbf{g}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j}) \leq 0 \tag{2c}$$

$$\mathbf{x}_{0,j} = \hat{\boldsymbol{x}} \tag{2d}$$

$$\sum_{j=1}^{S} \bar{\mathbf{E}}_j \mathbf{u}_j = 0 \tag{2e}$$

$$\forall j \in \{1, \cdots, S\}, \forall k \in \{1, \cdots, N\}$$

where the subscript (k, j) represents the j^{th} scenario at time sample k. $\mathbf{J}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j})$ is the cost function and $\mathbf{g}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j})$ represents the nonlinear inequality constraints. The initial conditions are enforced in (2d), where $\hat{\boldsymbol{x}}$ denotes the state measurements/estimates at the current time step. To keep the presentation simple, we assume full state feedback without measurement noise. The non-anticipativity constraints are enforced in (2e) which implies that the states that branch at the same parent node, must have the same control input. Note that $\mathbf{u}_j$ here represents the sequence of optimal control inputs for the j^{th} scenario, i.e. $\mathbf{u}_j = \left[\mathbf{u}_{0,j}^T \cdots \mathbf{u}_{N-1,j}^T\right]^T \in \mathbb{R}^{n_u N}$ and $\bar{\mathbf{E}}_j$ is given by

$$\bar{\mathbf{E}} = \left[\begin{array}{c|c|c|c|c} E_{1,2} & -E_{1,2} & & & \\ & E_{2,3} & -E_{2,3} & & \\ & & \ddots & \ddots & \\ & & & E_{S-1,S} & -E_{S-1,S} \end{array} \right] = \left[\begin{array}{c|c|c|c} \bar{\mathbf{E}}_1 & \bar{\mathbf{E}}_2 & \cdots & \bar{\mathbf{E}}_S \end{array} \right] \tag{3}$$

Let $n_{o,(j,j+1)}$ denote the number of common nodes between two consecutive scenarios j and $j+1$, then $E_{j,j+1} \in \mathbb{R}^{n_u n_{o,(j,j+1)} \times n_u N}$ as described in [9] and [13]. An added advantage of formulating the non-anticipativity constraints using this chain structure is that it results in sparse structures, which can be exploited by many solvers as explained in [13]. The reader is referred to [14] for recursive feasibility of the multistage scenario MPC problem.

B. Distributed Multistage Scenario MPC (D_{NLP})

It can be seen from (2) that the cost is additively separable in the scenarios and that the different scenarios are independent except for the non-anticipativity constraints (2e)Computing separate state and control trajectories $\mathbf{x}_{k,j}$ and $\mathbf{u}_{k,j}$ in (2) for each scenario j then facilitates parallel computations, where the different scenarios are decomposed into smaller subproblems. A master problem then co-ordinates the different subproblems. As motivated in [9], unlike dual decomposition, primal decomposition always ensures feasibility of the non-anticipativity constraints during the master problem iterations, which is crucial for closed loop implementation. Therefore, we only consider primal decomposition in this work and refer the reader to [7], [8] and [13] for dual decomposition approaches.

Primal decomposition iterates directly on the coupling variables [10]. By introducing a new auxiliary coupling variable $\boldsymbol{t}_l \in \mathbb{R}^{n_u}, \quad \forall l \in \{1, \ldots, \sum_{m=1}^{N_r} M^{m-1}\}$ to ensure non-anticipativity constraints, the subproblem for each j^{th} scenario can be formulated as

$$\boldsymbol{\Phi}(\boldsymbol{t}_l, \boldsymbol{p}_j) = \min_{\mathbf{x}_{k,j}, \mathbf{u}_{k,j}} \sum_{k=0}^{N-1} \mathbf{J}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}) \tag{4a}$$

$$\text{s.t} \quad \mathbf{x}_{k+1,j} = \mathbf{f}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j}) \tag{4b}$$

$$\mathbf{g}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j}) \leq 0 \tag{4c}$$

$$\mathbf{x}_{0,j} = \hat{\boldsymbol{x}} \tag{4d}$$

$$\bar{\mathbf{E}}_j \mathbf{u}_j = \bar{\mathbf{t}} \qquad \forall k \in \{1, \cdots, N\} \tag{4e}$$

where $\bar{\mathbf{t}}$ is used to enforce non-anticipativity constraints and is given by

$$\bar{\mathbf{t}} = \left[\begin{array}{c|c|c|c|c} t_{1,2} & -t_{1,2} & & & \\ & t_{2,3} & -t_{2,3} & & \\ & & \ddots & \ddots & \\ & & & t_{S-1,S} & -t_{S-1,S} \end{array} \right] \tag{5}$$

and $t_{j,j+1} \in \mathbb{R}^{n_u n_{o,(j,j+1)}}$ is composed of auxiliary variables $\boldsymbol{t}_l \in \mathbb{R}^{n_u}$ as described in detail in [9]. The master problem to update the auxiliary variables $\boldsymbol{t}_l$ is then given by,

$$\min_{\boldsymbol{t}_l} \sum_{j=1}^{S} \boldsymbol{\Phi}(\boldsymbol{t}_l, \boldsymbol{p}_j) \tag{6}$$

Proposition 1. *The solution to the master problem* (6) *can be expressed as the gradient descent step,*

$$\boldsymbol{t}_l^+ = \boldsymbol{t}_l + \alpha(\sum_{j=1}^{S} \lambda_j), \qquad \forall l \in \{1, \ldots, \sum_{m=1}^{N_r} M^{m-1}\} \tag{7}$$

Proof. The descent direction for the master problem (6) is given by the subgradient $-\nabla_{\boldsymbol{t}_l} \boldsymbol{\Phi}(\boldsymbol{t}_l, \boldsymbol{p}_j) = \sum_{j=1}^{S} \lambda_j$, where λ_j is the Lagrange multiplier that corresponds to the non-anticipativity constraints (4e). Equation (7) then provides a gradient descent direction to the master problem with a suitable step length α. See [10],[15] for more information. □

The master problem and the scenario subproblems are iteratively solved until $\boldsymbol{t}_l$ converges. One commonly used stopping criteria is that the change in $\boldsymbol{t}_l$ between two subsequent iterations, denoted by $\Delta \boldsymbol{t}_l = \|\boldsymbol{t}_l^+ - \boldsymbol{t}_l\|$ must be less than a certain user-defined tolerance ϵ as described in Algorithm 1 and also in [9].

III. Sensitivity-Based Distributed Multistage Scenario MPC

A. Sensitivity in Parametric NLP

To keep the presentation simple, we now reformulate (4) as a generic parametric NLP of the form,

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & \mathcal{J}(\boldsymbol{X}, \boldsymbol{p}) \\ \text{s.t.} \quad & c_i(\boldsymbol{X}, \boldsymbol{p}) = 0, \qquad \forall i \in \mathbb{E} \\ & c_i(\boldsymbol{X}, \boldsymbol{p}) \leq 0, \qquad \forall i \in \mathbb{I} \end{aligned} \tag{8}$$

where $\boldsymbol{X} \in \mathbb{R}^{n_{\boldsymbol{X}}}$ denotes the optimization (primal) variables of (4), $\boldsymbol{p} \in \mathbb{R}^{n_p}$ is the vector of uncertain parameters and the objective function is denoted by $\mathcal{J} : \mathbb{R}^{n_{\boldsymbol{X}}} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}$. The equality and inequality constraints $c : \mathbb{R}^{n_{\boldsymbol{X}}} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_c}$ are denoted by the sets $\mathbb{E} = \{1, \ldots, v\}$ and $\mathbb{I} = \{v+1, \ldots, n_c\}$, respectively.

The Lagrangian of (8) is defined as

$$\mathcal{L}(\mathbf{X}, \boldsymbol{\lambda}, \boldsymbol{p}) := \mathcal{J}(\boldsymbol{X}, \boldsymbol{p}) + \boldsymbol{\lambda}^T c(\boldsymbol{X}, \boldsymbol{p}) \tag{9}$$

where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers (dual variable). The Karush-Kuhn-Tucker (KKT) conditions for this problem can be stated as

$$\begin{aligned} \nabla_{\boldsymbol{X}} \mathcal{L}(\boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{p}) &= 0, \\ c_i(\boldsymbol{X}, \boldsymbol{p}) &= 0, \ i \in \mathbb{E}, \\ c_i(\boldsymbol{X}, \boldsymbol{p}) &\leq 0, \ i \in \mathbb{I}, \\ \boldsymbol{\lambda}_i^T c_i(\boldsymbol{X}, \boldsymbol{p}) &= 0, \ i \in \mathbb{I}, \\ \boldsymbol{\lambda}_i &\geq 0, \ i \in \mathbb{I}. \end{aligned} \tag{10}$$

Definition 1 (KKT point)**.** *Any point* $(\boldsymbol{X}^*, \boldsymbol{\lambda}^*)$ *that satisfies the KKT conditions* (10) *for a given parameter vector* $\boldsymbol{p}$ *is called a KKT point for* $\boldsymbol{p}$*.*

The active inequality constraints in (8) are denoted by the set $\mathbb{A}(\boldsymbol{X}, \mathbf{p}) = \{c_i(\boldsymbol{X}, \mathbf{p}) = 0, i \in \mathbb{I}\}$ and the active set is then given by $\mathbb{E} \cup \mathbb{A}$. For a given KKT point $(\boldsymbol{X}^*, \boldsymbol{\lambda}^*)$, the active set $\mathbb{A}$ has two subsets, namely a weakly active set $\mathbb{A}_0(\boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{p}) = \{i \in \mathbb{A}(\boldsymbol{X}, \boldsymbol{p}) \mid \boldsymbol{\lambda}_i = 0\}$ and a strong active set $\mathbb{A}_+(\boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{p}) = \{i \in \mathbb{A}(\boldsymbol{X}, \boldsymbol{p}) \mid \boldsymbol{\lambda}_i > 0\}$. Consequently, the inactive set $\mathbb{A}_-(\boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{p}) = \{c_i(\boldsymbol{X}, \mathbf{p}) < 0, i \in \mathbb{I}\}$ is the complement of set $\mathbb{A}$.

A constraint qualification is required to hold in order for the KKT conditions to be a necessary condition of optimality and in this work we consider the linear independence constraint qualification (LICQ) which is defined as follows.

Definition 2 (LICQ)**.** *Given a vector* $\boldsymbol{p}$ *and a point* $\boldsymbol{X}$*, the linear independence constraint qualification (LICQ) holds at* $(\boldsymbol{X}, \boldsymbol{p})$ *if the set of vectors* $\{\nabla_{\boldsymbol{X}} c_i(\boldsymbol{X}, \boldsymbol{p})\}_{i \in \mathbb{E} \cup \mathbb{A}(\boldsymbol{X}, \boldsymbol{p})}$ *are linearly independent.*

Definition 3 (SSOSC)**.** *The strong second order sufficient condition (SSOSC) holds at any KKT point* $(\boldsymbol{X}^*, \boldsymbol{\lambda}^*)$*, if* $\boldsymbol{d}^T \boldsymbol{H}(\boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{p}) \boldsymbol{d} > 0$ *for all* $\boldsymbol{d} \neq 0$ *such that* $\nabla_{\boldsymbol{X}} c_i(\boldsymbol{X}, \boldsymbol{p})^T \boldsymbol{d} = 0$ *for* $i \in \mathbb{E} \cup \mathbb{A}_+$*, where the Hessian of the Lagrangian* (9) *is given by*

$$\boldsymbol{H}(\boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{p}) = \nabla^2_{\boldsymbol{X}\boldsymbol{X}} \mathcal{J}(\boldsymbol{X}, \boldsymbol{p}) + \sum_{i=1}^{n} \nabla^2_{\boldsymbol{X}\boldsymbol{X}} c_i(\boldsymbol{X}, \boldsymbol{p}) \boldsymbol{\lambda}_i.$$

The LICQ and SSOSC guarantees that a KKT point is a strict local minimum.

Assumption 1. $\boldsymbol{X}^*$ *satisfies the KKT conditions* (10) *for a given parameter vector* $\boldsymbol{p}_0$ *and the linear independence constraint qualification (LICQ) and strong second order sufficient condition (SSOSC) hold at* $(\boldsymbol{X}^*, \boldsymbol{p}_0)$*.*

The reader is referred to Lemma 1 in [13] for detailed description on how the assumption of LICQ and positive definiteness of the Hessian translates to the multistage scenario MPC problem (4).

Theorem 1. *Let $\mathcal{J}, c$ be twice differentiable in $\boldsymbol{p}$ and $\boldsymbol{X}$ near a solution of* (8) $(\boldsymbol{X}^*, \boldsymbol{p}_0)$ *and let Assumption 1 hold, then the solution $(\boldsymbol{X}^*(\boldsymbol{p}), \boldsymbol{\lambda}^*(\boldsymbol{p}))$ is Lipschitz continuous in the neighbourhood of $(\boldsymbol{X}^*, \boldsymbol{\lambda}^*, \boldsymbol{p}_0)$ and the solution $\boldsymbol{X}^*(\boldsymbol{p})$ is directionally differentiable. Additionally, the directional derivative uniquely solves the following quadratic problem (QP):*

$$\begin{aligned} \min_{\Delta \boldsymbol{X}} \quad & \frac{1}{2} \Delta \boldsymbol{X}^T \nabla^2_{\boldsymbol{X}\boldsymbol{X}} \mathcal{L}\left(\boldsymbol{X}^*, \boldsymbol{p_0}, \boldsymbol{\lambda}^*\right) \Delta \boldsymbol{X} \\ & + \Delta \boldsymbol{X}^T \nabla_{\boldsymbol{X}\boldsymbol{p}} \mathcal{L}\left(\boldsymbol{X}^*, \boldsymbol{p_0}, \boldsymbol{\lambda}^*\right) \Delta \boldsymbol{p} \qquad (11) \\ \text{s.t.} \quad & \\ & \nabla_{\boldsymbol{X}} c_i\left(\boldsymbol{X}^*, \boldsymbol{p_0}\right)^T \Delta \boldsymbol{X} \\ & \quad + \nabla_{\boldsymbol{p}} c_i\left(\boldsymbol{X}^*, \boldsymbol{p_0}\right)^T \Delta \boldsymbol{p} = 0 \qquad i \in \mathbb{A}_+ \cup \mathbb{E}, \\ & \nabla_{\boldsymbol{X}} c_i\left(\boldsymbol{X}^*, \boldsymbol{p_0}\right)^T \Delta \boldsymbol{X} \\ & \quad + \nabla_{\boldsymbol{p}} c_i\left(\boldsymbol{X}^*, \boldsymbol{p_0}\right)^T \Delta \boldsymbol{p} \leq 0 \qquad i \in \mathbb{A}_0 \end{aligned}$$

Proof. See [16] and [17, Section 5.2]. □

The theorem above implies that a quadratic programming (QP) problem (11), often referred as *pure-predictor QP*, can be solved instead of a full NLP problem, in order to compute an approximate solution of (8) in the neighborhood of perturbation $\boldsymbol{p}_0$. This is the key to the sensitivity-based approach that we now use to efficiently solve the distributed multistage scenario MPC problem.

B. Path-following predictor-corrector QP

A corrector term can be added to the objective function in (11) to improve the approximation accuracy, as shown in [12]. With the technical assumptions that the parameter enters linearly in the constraints, we can formulate the following QP.

$$\begin{aligned} \min_{\Delta \boldsymbol{X}} \quad & \frac{1}{2} \Delta \boldsymbol{X}^T \nabla^2_{\boldsymbol{X}\boldsymbol{X}} \mathcal{L}\left(\boldsymbol{X}^*, \boldsymbol{p_0} + \boldsymbol{\Delta p}, \boldsymbol{\lambda}^*\right) \Delta \boldsymbol{X} \\ & + \Delta \boldsymbol{X}^T \nabla_{\boldsymbol{X}\boldsymbol{p}} \mathcal{L}\left(\boldsymbol{X}^*, \boldsymbol{p_0} + \boldsymbol{\Delta p}, \boldsymbol{\lambda}^*\right) \Delta \boldsymbol{p} \\ & + \nabla_{\boldsymbol{X}} \mathcal{J}^T \Delta \boldsymbol{X} \qquad (12) \\ \text{s.t.} \quad & \\ & c_i\left(\boldsymbol{X}^*, \boldsymbol{p}_0 + \Delta \boldsymbol{p}\right) + \nabla_{\boldsymbol{p}} c_i\left(\boldsymbol{X}^*, \boldsymbol{p}_0 + \Delta \boldsymbol{p}\right)^T \Delta \boldsymbol{p} + \\ & \nabla_{\boldsymbol{X}} c_i\left(\boldsymbol{X}^*, \boldsymbol{p}_0 + \Delta \boldsymbol{p}\right)^T \Delta \boldsymbol{X} = 0 \ , i \in \mathbb{A}_+ \cup \mathbb{E}, \\ & c_i\left(\boldsymbol{X}^*, \boldsymbol{p}_0 + \Delta \boldsymbol{p}\right) + \nabla_{\boldsymbol{p}} c_i\left(\boldsymbol{X}^*, \boldsymbol{p}_0 + \Delta \boldsymbol{p}\right)^T \Delta \boldsymbol{p} \\ & \quad + \nabla_{\boldsymbol{X}} c_i\left(\boldsymbol{X}^*, \boldsymbol{p}_0 + \Delta \boldsymbol{p}\right)^T \Delta \boldsymbol{X} \leq 0 \ , i \in \mathbb{A}_0. \end{aligned}$$

The QP formulation (12) is known as the *predictor-corrector QP*. It can be thought of a combination of a first-order sensitivity step and an SQP step towards the solution for the new parameter value. In the small neighborhood of $\boldsymbol{p}_0$, the predictor-corrector QP formulation was shown to provide good approximations of the NLP solution. However, the different models M used in the scenario optimization need not necessarily be in the small neighbourhood of each other. Therefore, in order to allow for large perturbations (i.e. large $\Delta \boldsymbol{p}$, we propose to apply a *path-following* approach [12], where we solve a series of QP problems sequentially similar to an Euler integration scheme for ordinary differential equations. [1]

Given an optimal solution $\boldsymbol{X}^*(\boldsymbol{p}_{j-1})$ for a parameter vector $\boldsymbol{p}_{j-1}$, we want to compute the optimal solution for a parameter vector $\boldsymbol{p}_j$. The path-following predictor-corrector QP then updates $\boldsymbol{X}$ for the parameter sequence $\boldsymbol{p}$ according to

$$\boldsymbol{p}\left(\nu_\kappa\right) = \left(1 - \nu_\kappa\right) \boldsymbol{p}_{j-1} + \nu_\kappa \boldsymbol{p}_j \qquad (13)$$

where $\nu_0 = 0$ until it reaches $\nu_\kappa = 1$. In other words $\nu_0 = 0 < \nu_1 < \nu_2 < \cdots < \nu_\kappa = 1$. Given a sufficiently small step $\Delta\nu$, the path-following predictor-corrector QP, after solving a series of QP problems, provides the optimal solution $\boldsymbol{X}^*(\boldsymbol{p}_j)$ for a parameter vector $\boldsymbol{p}_j$. In this paper, for simplicity, we use a fixed step size $\Delta\nu = \nu_{\kappa+1} - \nu_\kappa$.

C. Sensitivity-based path-following distributed multistage scenario MPC

Based on these developments, we are now ready to formulate the sensitivity-based distributed multistage scenario MPC algorithm.

Assumption 2. *There exists a continuous path of unique optimal solutions between the subproblems $\boldsymbol{\Phi}(\boldsymbol{t}_l, \boldsymbol{p}_{j-1})$ and $\boldsymbol{\Phi}(\boldsymbol{t}_l, \boldsymbol{p}_j)$.*

Corollary 1 (Main result)**.** *Let $\left[\boldsymbol{X}^*(\boldsymbol{p}_{j-1}), \boldsymbol{\lambda}^*(\boldsymbol{p}_{j-1})\right]$ be the solution for one scenario subproblem obtained by solving the NLP $\boldsymbol{\Phi}(\boldsymbol{t}_l, \boldsymbol{p}_{j-1})$ and let Assumptions* (1) *and* (2) *hold. Further, let $\boldsymbol{p}_j$ be in the neighbourhood of $\boldsymbol{p}_{j-1}$, then the solution for all other scenario subproblems $\boldsymbol{\Phi}(\boldsymbol{t}_l, \boldsymbol{p}_j)$ with the same set of auxiliary variables $\boldsymbol{t}_l$ is Lipschitz continuous in the neighbourhood of $\left[\boldsymbol{X}^*(\boldsymbol{p}_{j-1}), \boldsymbol{\lambda}^*(\boldsymbol{p}_{j-1})\right]$ and can be obtained by repeatedly solving the predictor-corrector QP* (12)*.*

Proof. Since the only difference between the scenarios $\boldsymbol{\Phi}(\boldsymbol{t}_l, \boldsymbol{p}_{j-1})$ and $\boldsymbol{\Phi}(\boldsymbol{t}_l, \boldsymbol{p}_j)$ is the parameter vector $\boldsymbol{p}_j$, it follows from Theorem 1 that the NLP problem $\boldsymbol{\Phi}(\boldsymbol{t}_l, \boldsymbol{p}_j)$ can be approximated by repeatedly solving the QP problem (12) for a small parameter perturbation $\Delta \boldsymbol{p}$ along the path from $\boldsymbol{p}_{j-1}$ to $\boldsymbol{p}_j$. □

Corollary 1 above suggests that instead of solving S number of NLPs, the multistage scenario MPC problem can be solved using M^{N_r-1} number of NLPs and the remaining subproblems can be solved as QPs. The number of common nodes between two consecutive scenarios $n_{o,(j,j+1)}$ is used to check if the two scenarios have the same set of auxiliary variables $\boldsymbol{t}_l$. The sensitivity-based distributed scenario MPC algorithm then consists of the following three steps.

1) For a given primal master variable $\boldsymbol{t}_l$, solve the NLP problem $\boldsymbol{\Phi}(\boldsymbol{t}_l, \boldsymbol{p}_{j-1})$ for one subproblem with the parameter vector $\boldsymbol{p}_{j-1}$ to obtain the optimal primal and dual variables $\boldsymbol{X}^*(\boldsymbol{p}_{j-1})$ and $\boldsymbol{\lambda}^*(\boldsymbol{p}_{j-1})$, respectively.
2) For the subsequent scenario subproblems with the same set of auxiliary variables, compute an approximation of

[1]Note that the path-following in [12] was applied to advance step MPC, whereas in this paper, we apply it to the distributed scenario MPC problem.

the NLP problem $\boldsymbol{\Phi}(\boldsymbol{t}_l, \boldsymbol{p}_j)$ using the QP (12) in a path-following manner as described in Section III-B.

3) Using the computed Lagrange multipliers corresponding to the non-anticipativity constraints (4e) $\lambda \subset \boldsymbol{\lambda}$ from all the subproblems, update the primal master variable $\boldsymbol{t}_l$ according to (7).

A sketch of the proposed sensitivity-based multistage scenario MPC procedure is described in Algorithm 1.

Algorithm 1 Sensitivity-based distributed multistage scenario MPC

Define tolerance $\epsilon > 0$, $\Delta\nu \leq 1$.
Input: At each time step, initial state $\hat{\boldsymbol{x}}$, initial $\boldsymbol{t}_l^0$ and $\Delta \boldsymbol{t}_l > \epsilon$, initial α

```
while Δt_l > ε do
    for j = 1, 2, ..., S do
        if (j = 1) ∨ (n_{o,(j-1,j)} ≤ N_r − 1) then
            [X*(p_j), λ*(p_j)] ← solution NLP Φ(t_l, p_j)
        else                    ▷ Approximate NLP using QP (12).
            [ΔX*, λ*(p_j)] ← QP_PF(X*, λ*, p_{j-1}, p_j).
            Set X*(p_j) = X*(p_{j-1}) + ΔX*.
        end if
    end for
    Update t_l^+ = t_l + α(Σ_{j=1}^{S} λ_j)
    Update Δt_l = ||t_l^+ − t_l||
end while
```

```
function QP_PF(X*(p_{j-1}), λ*(p_{j-1}), p_{j-1}, p_j)
    Define 𝔸_+.
    Set ν_κ = 0.
    while ν_κ < 1 do
        [ΔX*, λ*] ← solution QP (12) with p = p(ν_κ)
        X* = X* + ΔX*
        ν_{κ+1} ← ν_κ + Δν
        p(ν_κ) = (1 − ν_κ) p_{j-1} + ν_κ p_j
    end while
    return ΔX*, λ*
end function
```

Output: $\boldsymbol{X}^*(\boldsymbol{p}_j), \forall j \in \{1, \ldots, S\}$

IV. Illustrative Example

In this work, we consider an exothermic chemical reactor case study from [18] that is widely used in process control literature, where component A is converted to product B ($A \leftrightharpoons B$). The reaction rate is given as $r = k_1 C_A - k_2 C_B$ where $k_1 = C_1 e^{\frac{-E_1}{RT}}$ and $k_2 = C_2 e^{\frac{-E_2}{RT}}$. The dynamic model consists of two mass balances and an energy balance:

$$\dot{C}_A = \frac{1}{\tau}(C_{A,i} - C_A) - r \tag{14a}$$

$$\dot{C}_B = \frac{1}{\tau}(C_{B,i} - C_B) + r \tag{14b}$$

$$\dot{T} = \frac{1}{\tau}(T_i - T) + \frac{-\Delta H_{rx}}{\rho C_p} r \tag{14c}$$

where time constant $\tau = 60$s, C_A and C_B are concentrations of the two components in the reactor and $C_{A,i}$ and $C_{B,i}$ are in the inflow. T_i is the inlet temperature and T is the reaction temperature. Other model parameters for the process can be found in [18]. The objective is to compute the optimal inlet temperature T_i such that we can minimize the operational cost while keeping the reactor temperature $T \leq 425$K. We assume the concentration of component B in the feed stream is uncertain and consider five discrete realizations, namely, $C_{B,i} \in \{0, 0.05, 0.1, 0.15, 0.2\}$molL^{-1}.

We use a multistage scenario MPC with a prediction horizon of $T = 300$s divided equally into $N = 20$ samples. The system model (14) is discretized using third order direct collocation and the resulting finite horizon multistage MPC problem was implemented in `MATLAB` using `CasADi` algorithmic differentiation tool version 3.1.0 [19]. The NLP problem was solved using the `IPOPT` solver and the QP problems were solved using `TOMLAB MINOS`. The optimization problem then consists,

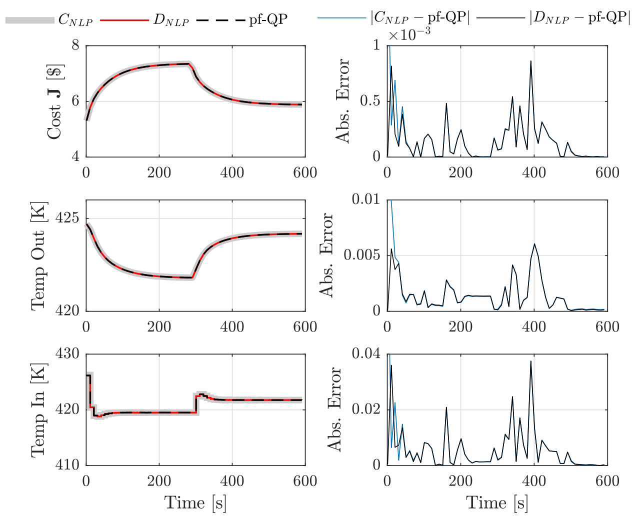
1) $\mathbf{J}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}) = \left(-2.009 C_B + (1.657 \times 10^{-3} T_i)^2\right)$,
2) discretized system model,
3) uncertain parameter $\boldsymbol{p} = C_{B_i}$ discretized into $M = 5$ finite models, namely, $C_{B_i} \in \{0, 0.05, 0.1, 0.15, 0.2\}$,
4) process constraints $\mathbf{g}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}) = T - 425$, and
5) non-anticipativity constraints (2e).

We note that in the considered case study, the constraint $T \leq 425$K becomes active at steady state only when $C_{B_i} \in \{0, 0.05\}$ and not when $C_{B_i} \in \{0.1, 0.15, 0.2\}$. Therefore the active constraint set changes between the different scenarios. The true realization of C_{B_i} used in the simulations changes from $C_{B_i} = 0.15$molL^{-1} to $C_{B_i} = 0.05$molL^{-1} at time $t = 300$s.

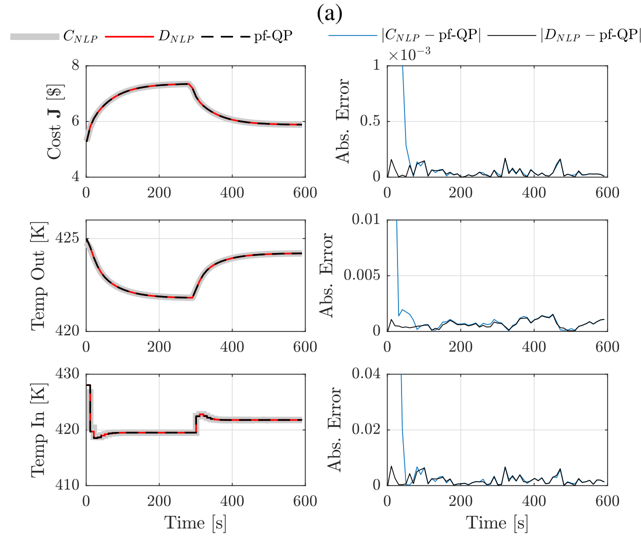
1) Simulation 1: In the first simulation we consider a robust horizon of $N_r = 1$ and hence we have $S = 5$ scenarios. We first compute the solution of a fully centralized approach (C_{NLP}), i.e. (2) to be used as a benchmark. The multistage scenario MPC is then solved using the primal decomposition approach i.e. (4), where all the scenario subproblems are solved as NLP problems (D_{NLP}). We then solve the optimization problem using the proposed path-following QP (pf-QP), where the first scenario was solved as NLP problem and the subsequent four scenarios are solved using the path-following predictor-corrector QP (12) as described in Algorithm 1 with a fixed step size $\Delta\nu = 0.5$. Hence two QPs were solved to approximate each subproblem. For the distributed scenario approaches, the tolerance was chosen to be $\epsilon = 0.001$ and a feasibility ensuring backtracking algorithm was used to select a suitable step length α.

The closed loop implemented solution for the proposed sensitivity-based distributed scenario MPC are compared with the fully centralized scenario MPC (C_{NLP}) and the distributed scenario MPC solved using full NLPs (D_{NLP}) along with the corresponding absolute errors in Fig.1a.

2) Simulation2: In the second simulation we consider the same problem, but a robust horizon of $N_r = 2$ leading to a scenario tree with $S = 25$ scenarios. By using the path-following predictor-corrector QP (12), we solve 5 scenarios using NLPs and 20 scenarios were solved using path-following QPs. The closed loop implemented solution for the proposed sensitivity-based distributed scenario MPC (pf-QP) are compared with the



(a)



(b)

Fig. 1: Closed loop simulation results for fully centralized approach C_{NLP} (Thick gray lines), distributed approach with full NLP D_{NLP} (solid red lines) and the proposed path-following approach pf-QP (black dashed lines) for (a) $N_r = 1$, $S = 5$ scenarios (b) $N_r = 2$, $S = 25$ scenarios. The corresponding absolute errors are plotted in the right hand side subplots.

fully centralized scenario MPC (C_{NLP}) and the distributed scenario MPC solved using full NLPs (D_{NLP}). The closed-loop results and the corresponding absolute errors are shown in Fig.1b.

The average CPU times for each subproblem for the two simulation cases are reported in Table I. Note that the computation time depends heavily on the implementation and computation time of the QP may be further improved by using dedicated high performance QP solvers instead of an off-the shelf solver.

The simulation results in Fig.1a and Fig.1b clearly demonstrates that the proposed sensitivity-based distributed Scenario MPC is able to provide a very good approximation of the centralized scenario MPC and full NLP distributed scenario MPC. The simulations also demonstrate that the proposed approach can handle changes in active constraint set between the different subproblems.

TABLE I: CPU times (in sec)

	$N_r = 1$			$N_r = 2$		
	max	avg	min	max	avg	min
NLP	0.137	0.073	0.053	0.127	0.085	0.077
pf-QP	0.093	0.011	0.0082	0.062	0.024	0.012

References

[1] P. Scokaert and D. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Transactions on Automatic control*, vol. 43, no. 8, pp. 1136–1142, 1998.

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

[3] G. C. Calafiore and L. Fagiano, "Robust model predictive control via scenario optimization," *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 219–224, 2013.

[4] G. Schildbach, L. Fagiano, C. Frei, and M. Morari, "The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations," *Automatica*, vol. 50, no. 12, pp. 3009–3018, 2014.

[5] L. Fagiano, G. Schildbach, M. Tanaskovic, and M. Morari, "Scenario and adaptive model predictive control of uncertain systems," *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 352–359, 2015.

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

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

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

[9] D. Krishnamoorthy, B. Foss, and S. Skogestad, "A distributed algorithm for scenario-based model predictive control using primal decomposition." *IFAC ADCHEM*, pp. In–Press, 2018 (Keynote paper).

[10] D. P. Bertsekas, *Nonlinear programming*. Athena Scientific, 1999.

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

[12] E. Suwartadi, V. Kungurtsev, and J. Jäschke, "Sensitivity-based economic NMPC with a path-following approach," *Processes*, vol. 5, no. 1, p. 8, 2017.

[13] E. Klintberg, J. Dahl, J. Fredriksson, and S. Gros, "An improved dual newton strategy for scenario-tree MPC," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 3675–3681.

[14] M. Maiworm, T. Bäthge, and R. Findeisen, "Scenario-based model predictive control: Recursive feasibility and stability," *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 50–56, 2015.

[15] S. Boyd, L. Xiao, A. Mutapcic, and J. Mattingley, "Notes on decomposition methods," *Notes for EE364B, Stanford University*, pp. 1–36, 2007.

[16] S. M. Robinson, "Strongly regular generalized equations," *Mathematics of Operations Research*, vol. 5, no. 1, pp. 43–62, 1980.

[17] J. F. Bonnans and A. Shapiro, "Optimization problems with perturbations: A guided tour," *SIAM review*, vol. 40, no. 2, pp. 228–264, 1998.

[18] C. G. Economou, M. Morari, and B. O. Palsson, "Internal model control: Extension to nonlinear system," *Industrial & Engineering Chemistry Process Design and Development*, vol. 25, no. 2, pp. 403–411, 1986.

[19] J. Andersson, "A General-Purpose Software Framework for Dynamic Optimization," PhD thesis, October 2013.