IEEE Control Systems Letters paper presented at
2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Improving Primal Decomposition for Multistage MPC Using an Extended Newton method

Simen Bjorvand and Johannes Jäschke

Abstract— **Multistage model predictive control is a robust MPC formulation that takes into account parametric uncertainty by constructing a finite set of coupled scenarios. As the amount of scenarios increase so does computational cost and real-time implementation might not be possible. Scenario decomposition has been proposed to distribute computations and make real-time implementation possible, however, typically the subproblems are coordinated using the steepest descent method with slow convergence properties. In this paper a primal decomposition algorithm is improved by the use of a nonsmooth Newtons method for continuous nonsmooth equations. The proposed algorithm is applied to a gas-lift optimization system and compared to the standard primal decomposition method using steepest descent.**

I. INTRODUCTION

In model predictive control (MPC) an optimal control problem is formulated and solved minimizing some cost while obeying the model equations of the system. An optimal control trajectory is calculated and the current control action is acted upon the system. Uncertainties in the model are handled through feedback, but if the uncertainties are too large a robust MPC scheme is necessary. One of these schemes which accounts for uncertainties is the multistage MPC (msMPC) also called the scenario MPC [14].

The drawback of the msMPC is that the problem becomes very large as it handles more uncertainty and computational cost increase exponentially. Several methodologies have been proposed to address this challenge. In [22] and [19] parametric sensitivity based approaches were proposed while in [6] and [8] parallelizable linear algebra approaches exploiting the structure of the problem were proposed.

Another approach much studied in literature is to employ decomposition algorithms to the msMPC. In these methods the msMPC problem is decomposed into smaller subproblems with a coordination algorithm on top ensuring that the subproblems converge to the optimal solution of the centralized problem. Each subproblem can be solved fast but must be resolved a repeated number of times to coordinate. While these approaches do not necessarily reduce the overall computational time to compute a control action, the computations can be performed in parallel such that the overall computational delay is reduced. The msMPC is usually decomposed per scenario. A dual decomposition approach was proposed by [15] where a progressive hedging approach was used. This method was further improved in [16]. The drawback of dual decomposition approaches is that they do not provide a single feasible control action until the algorithm converges.

This was addressed in [10] where a primal decomposition scheme was used instead, which could supply a feasible but suboptimal control action even if terminated early. As early termination is not desirable, speeding up the algorithm is desirable. Solving the distributed subproblems is the computational bottleneck of the algorithm. Two ways to speed up the algorithm are to reduce the computational time of solving the subproblems and reduce the number of necessary resolves in the coordination algorithm. In [12] a sensitivity-based path-following algorithm was employed to speed up the computations of the subproblems. In this paper, we propose to speed up the coordination step of the primal decomposition algorithm by employing a nonsmooth Extended Newton Method.

Using a Newton method for speeding up dual decomposition of a linear msMPC was done in [5] and for nonlinear msMPC using dual decomposition was done [16]. A real-time iterations approach to msMPC combined with dual decomposition employed a nonsmooth Newtons Method in [13]. This work differs by employing a nonsmooth Newton method to a primal decomposition algorithm and showing local quadratic convergence for the proposed algorithm.

II. PRELIMINARIES

A. Multistage MPC

Here we present the concept and our notation for msMPC, see [14] for details. Consider a discretized nonlinear system

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

where $x_k \in \mathbb{R}^{n_x}$ are the states, $u_k \in \mathbb{R}^{n_u}$ are the control inputs and $p_k \in \mathbb{R}^{n_p}$ are uncertain model parameters at time iteration k. The function $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_x}$ represents the nonlinear model of the system. A sequence of inputs u_k for $k = 0, ..., N-1$, that minimizes an expected cost over a time horizon of length N is calculated.

The parameters p_k are sampled from a distribution. A set of M points from this distribution is selected to represent the possible realizations for the uncertain parameters. At each time step the each possible realizations are considered, which leads the msMPC to branch into M new branches. To reduce the size of the msMPC problem a robust horizon N_r is used, which are the number of steps where branching is considered. This lead to a structure of $N_s = M^{N_r}$ scenarios.

This work was supported by the Norwegian Research Council through FRIPRO Project SensPATH.

The authors are with the Department of Chemical Engineering, Norwegian University of Science and Technology, 7491 Trondheim, Norway (e-mail: simen.bjorvand@ntnu.no; johannes.jaschke@ntnu.no).

The scenarios are tied together at certain points in their prediction horizon. The nodes where the scenarios are tied together can be thought of as knots with the tree having $N_{Kn} = \sum_{k=0}^{N_r-1} M^k$ knots. In the nodes where the scenarios are tied together they have to make the same control action u_k. This is known as the nonanticipativity constraints.

The msMPC problem is formulated as

$$\min_{x,u,l} \sum_{s=1}^{N_s} w^s \sum_{k=0}^{N-1} j_k^s(x_k^s, u_k^s, p_k^s) \tag{2a}$$

$$\text{s.t.} \quad x_{k+1}^s = f(x_k^s, u_k^s, p_k^s) \tag{2b}$$

$$c(x_k^s, u_k^s) \leq 0 \qquad , k = 0, \ldots, N-1 \tag{2c}$$

$$P_u u^s - P_l^s l = 0 \quad , s = 1, \ldots, N_s \tag{2d}$$

where $j_k^s : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}$ are the stage costs, w^s the weight per scenario and $c : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \mapsto \mathbb{R}$ represents inequality constraints. Constraint (2d) represents the nonanticipativity constraints, where $P_u \in \mathbb{R}^{N_r n_u \times N n_u}$ such that $P_u u^s = [u_0^s, \ldots, u_{N_r}^s]^T$, the decision variables $l \in \mathbb{R}^{N_l}, N_l = N_{Kn} n_u$ are connecting variables representing the knots and $P_l^s \in \mathbb{R}^{N_r n_u \times N_l}$ are selection matrices enforcing the tree structure. For the scenario tree shown in fig 1 we have $P_l^1 = P_l^2 = P_l^3 = \begin{bmatrix} I_{n_u} & 0 & 0 & 0 \\ 0 & I_{n_u} & 0 & 0 \end{bmatrix}$, $P_l^4 = P_l^5 = P_l^6 = \begin{bmatrix} I_{n_u} & 0 & 0 & 0 \\ 0 & 0 & I_{n_u} & 0 \end{bmatrix}$, $P_l^7 = P_l^8 = P_l^9 = \begin{bmatrix} I_{n_u} & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{n_u} \end{bmatrix}$.

This optimization problem can be reformulated in a more compact form which will be used in the rest of this paper,

$$\min_{\omega,l} \sum_{s=1}^{N_s} J^s(\omega^s) \tag{3a}$$

$$\text{s.t.} \quad h^s(\omega^s) = 0 \tag{3b}$$

$$g^s(\omega^s) \leq 0 \tag{3c}$$

$$P_\omega \omega^s - P_l^s l = 0 \quad s = 1, \ldots, N_s, \tag{3d}$$

where $\omega^s = [x^{sT}, u^{sT}]^T \in \mathbb{R}^{n_{\omega^s}}$, $J^s : \mathbb{R}^{n_{\omega^s}} \mapsto \mathbb{R}$ represents the cost, $h^s : \mathbb{R}^{n_{\omega^s}} \mapsto \mathbb{R}^{n_h}$ the equality constraints, $g^s : \mathbb{R}^{n_{\omega^s}} \mapsto \mathbb{R}^{n_g}$ the inequality constraints and equation (3d) represents the nonanticipativity constraints for scenario s with $P_\omega \in \mathbb{R}^{N_r n_u \times n_{\omega^s}}$.

B. Primal decomposition of msMPC

In decomposition methods the large optimization problem is split into smaller subproblems dependent on some parameter variables, where a coordinator problem finds the optimal parameter variables to make the subproblems converge to a common solution. Many different decomposition methods have been proposed, and in this paper, we focus on primal decomposition as was used in [10]. Here these parameter variables are primal variables from the original problem [1].

Using primal decomposition the large msMPC problem (3) can be decomposed into smaller subproblems in the following way,

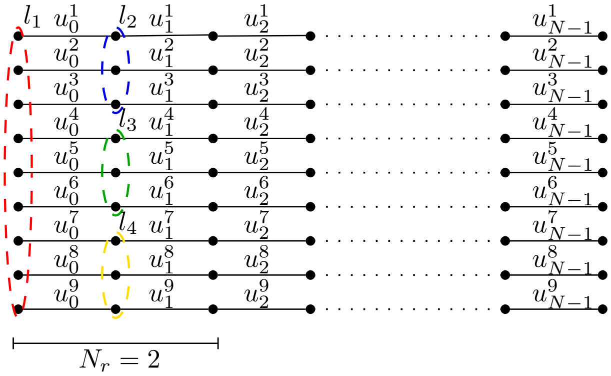
l_1
l_2
l_3
l_4
u_0^1 u_0^2 u_0^3 u_0^4 u_0^5 u_0^6 u_0^7 u_0^8 u_0^9
u_1^1 u_1^2 u_1^3 u_1^4 u_1^5 u_1^6 u_1^7 u_1^8 u_1^9
u_2^1 u_2^2 u_2^3 u_2^4 u_2^5 u_2^6 u_2^7 u_2^8 u_2^9
u_{N-1}^1 u_{N-1}^2 u_{N-1}^3 u_{N-1}^4 u_{N-1}^5 u_{N-1}^6 u_{N-1}^7 u_{N-1}^8 u_{N-1}^9
$N_r = 2$

Fig. 1. Schematic diagram of msMPC with a robust horison of $N_r = 2, M = 3$, where each set of connected nodes represent a scenario. The colored dotted circles represents the $N_{Kn} = 4$ knots where the nodes from the different scenarios are tied together.

$$\Theta^s(l) = \min_{\omega^s} \quad J^s(\omega^s) \tag{4a}$$

$$\text{s.t.} \quad h^s(\omega^s) = 0 \quad (\lambda^s) \tag{4b}$$

$$g^s(\omega^s) \leq 0 \quad (\mu^s) \tag{4c}$$

$$P_\omega \omega^s - P_l^s l = 0 \quad (\gamma^s), \tag{4d}$$

where the decision variables l in the centralized problem become the parameters for the subproblems (4). The variables λ^s, μ^s and γ^s denote the dual variables related to the equality constraints (4b), inequality constraints (4c), and nonanticipativity constraints (4d) respectively. The N_s subproblems (4) can be solved independently of each other in parallel for a given value of l. The centralized problem (3) is recovered by the following problem, where the coordinator variables from (4) become the decision variables

$$\min_{l} \sum_{s=1}^{N_s} \Theta^s(l). \tag{5}$$

C. Standard Approach (Steepest Descent)

From [1] we have that a subgradient of the subproblem (4) with respect to the parameter l can be found as

$$\nabla_l \Theta^s(l) = -P_l^{sT} \gamma^{s*}(l) \tag{6}$$

where γ^{s*} is the optimal dual variable related to (4d) given l. In the standard approach to solving the coordinator problem (5) as outlined in [10] the subgradient of (5) is calculated as

$$G(l) = -\sum_{s=1}^{N_s} P_l^{sT} \gamma^{s*}(l) \tag{7}$$

and a Steepest Descent (SD) method is applied to find the vector l such that the subgradient $G(l)$ is equal to zero. If l_k is the point given at iteration k the step to the next point is calculated as

$$\Delta l_k = -\alpha G(l_k) \tag{8}$$

where Δl_k is the step in the variable l and $\alpha > 0$ is the step length.

The Steepest Descent method only has linear convergence and is therefore slow to converge. This leads to resolving the subproblems (4) a large number of times, which is the computational bottleneck of the distributed msMPC algorithm. We propose to speed up the algorithm by using a nonsmooth Newton method with quadratic convergence.

D. Nonlinear Sensitivity Concepts

Before we procede we will introduce some concepts that will be used later. Let $z^s = [\omega^{sT}, \lambda^{sT}, \mu^{sT}, \gamma^{sT}]^T$ be a primal-dual pair for subproblem (4) and let

$$\begin{aligned} L^s(z^s,l) =& J^s(\omega^s) + h^s(\omega^s)^T\lambda^s + g^s(\omega^s)^T\mu^s \\ &+ (P_\omega \omega^s - P_l^s l)^T\gamma^s \end{aligned} \tag{9}$$

denote the subproblem Lagrangian. The point $z^{s*}(l)$ is called a KKT point for (4) if it is the root of the following system of nonsmooth equations, also known as the KKT conditions

$$\begin{aligned} &\nabla_{\omega^s} L^s(z^{s*};l) = 0 \\ &h^s(\omega^{s*}) = 0 \\ &\min(-g^s(\omega^{s*}), \mu^{s*}) = 0 \\ &P_\omega \omega^s - P_l^s l = 0. \end{aligned} \tag{10}$$

The set of active constraints is defined as $\mathcal{A}^s(l) = \{i|g_i^s(\omega^{s*}(l)) = 0, i = 1, \dots, n_{g^s}\}$, and inactive constraints as $\mathcal{A}_0^s(l) = \{1, \dots, n_{g^s}\}\backslash\mathcal{A}^s(l)$. The active constraints can be further subdivided into two sets. The set of strongly active constraints $\mathcal{A}_+^s(l) = \{i \in \mathcal{A}^s(l)|\mu_i^{s*} > 0\}$ and the weakly active constraints $\mathcal{A}_-^s(l) = \mathcal{A}^s(l)\backslash\mathcal{A}_+^s(l)$. Let $|\mathcal{D}|$ denote the size of an integer set $\mathcal{D}$.

A KKT point $z^{s*}(l)$ is an optimal solution of (4) if a constraint qualification holds as well as a second-order condition of optimality. This paper will consider the Linear Independence Constraint Qualification (LICQ) and the Strong Second Order Sufficient Condition (SSOSC). In addition the Strong Complementarity (SC) will be used below.

Definition: LICQ holds for a primal solution $\omega^{s*}(l)$ if the gradients of the equality constraints $\nabla_{\omega^s} h^s(\omega^{s*})$, the active inequality constraints $\nabla_{\omega^s} g_i^s(\omega^{s*}), i \in \mathcal{A}(l)$ and the rows in P_ω are linearly independent.

Definition: SSOSC holds for a primal-dual solution $z^{s*}(l)$ if $d^T\nabla^2_{\omega^{s*}} L^s(z^{s*};l)d > 0$ for $d \neq 0 \in \{d|\nabla_{\omega^s} h^s(\omega^{s*})^T d = 0, P_\omega d = 0, \nabla_{\omega^s} g_i^s(\omega^{s*})^T d = 0 \quad \forall i \in \mathcal{A}_+(l)\}$.

Definition: SC holds for a primal-dual solution $z^{s*}(l)$ if its weakly active set $\mathcal{A}_-(l)$ is empty, $|\mathcal{A}_-(l)| = 0$.

The following theorem introduced and proved in [3] establishes the sensitivity of a primal-dual solution $z^{s*}(l)$ of problem (4).

Theorem 1 [3]: (NLP sensitivity for a local primal-dual solution $z^{s*}(l_0)$ of problem (4)): If the functions J^s, h^s and g^s are twice continuously differentiable in ω^s in a neighborhood of $\omega^{s*}(l_0)$, LICQ, SSOSC and SC holds then:

- $\omega^{s*}(l_0)$ is local unique optimum, and the dual variables $\lambda^{s*}(l_0), \mu^{s*}(l_0)$ and $\gamma^{s*}(l_0)$ are unique.
- For an l in a neighborhood of l_0 there exist a continuously differentiable function $z^s(l) = [\omega^{s*}(l)^T, \lambda^{s*}(l)^T, \mu^{s*}(l)^T, \gamma^{s*}(l)^T]^T$ which is an optimal primal-dual solution of (4) for l.
- SC holds for a l in the neighborhood of l_0.

The following theorem from [4] expands on Theorem 1 when SC fails.

Theorem 2 [4]: (NLP sensitivity for a local primal-dual solution $z^{s*}(l_0)$ of problem (4)): If the the functions J^s, h^s and g^s are twice continuously differentiable in ω^s in a neighborhood of $\omega^{s*}(l_0)$, LICQ and SSOSC holds then:

- $\omega^{s*}(l_0)$ is local unique optimum, and the dual variables $\lambda^{s*}(l_0), \mu^{s*}(l_0)$ and $\gamma^{s*}(l_0)$ are unique.
- For an l in a neighborhood of l_0 there exist a continuous function $z^s(l) = [\omega^{s*}(l)^T, \lambda^{s*}(l)^T, \mu^{s*}(l)^T, \gamma^{s*}(l)^T]^T$ which is an optimal primal-dual solution of (4) for l.
- The value function of subproblem (4) $\Theta^s(l)$ is differentiable with respect to l at l_0 with the derivative being $\nabla_l \Theta^s(l_0) = -P_l^{sT}\gamma^{s*}(l_0)$.

III. Primal Decomposition of msMPC using an Extended Newton Method

Let l^* be an optimal solution of (5).

Assumption 1: The functions J^s, h^s and g^s are all twice continuously differentiable for all N_s subproblems.

Assumption 2: LICQ and SSOSC holds for the primal-dual solutions $z^{s*}(l^*)$ for all N_s subproblems (4).

From Theorem 1 and 2 if Assumption 1 and 2 hold there is a neighborhood $\mathcal{L} \subset \mathbb{R}^{N_l}$ of l^* where the optimal primal-dual solution $z^{s*}(l)$ is given by a PC^1 function for all subproblems $s \in \{1, \dots, N_s\}$.

Note that even with smooth functions, the primal-dual solution of (4) can be nonsmooth because the active inequality constraints change with the parameter l.

Definition: A function $\sigma : V \subseteq \mathbb{R}^n \mapsto \mathbb{R}^m$ is PC^1 if it is continuous and if for all $y_0 \in V$ there is a neighborhood $W \subset V$ of y_0 and finite number of C^1 functions $\sigma^1, \dots, \sigma^N$ such that $\sigma(y) \in \{\sigma^1(y), \dots, \sigma^N(y)\} \quad \forall y \in W$.

From Theorem 2 we have that the gradient of the subproblems (4) was given by (6).The gradient $G(l)$ of the coordinator problem (5) is then given by (7) meaning it is a PC^1 function for $l \in \mathcal{L}$. The optimal solution of (5) is given as the root of the gradient $G(l)$, so we propose to use the Extended Newton (EN) method from [7].

A. Algorithm

The main contribution of this work is algorithm 1. The algorithm uses the Extended Newton method from [7] to solve the coordinator problem (5). The Extended Newton method extends the Newton method to PC^1 functions and it retains quadratic convergence for a starting point near the solution.

$G(l)$ will be used as the PC^1 function to explain the Extended Newton method. Let $\bar{\mathcal{L}}^s \subseteq \mathcal{L}$ be the set where SC holds for the optimal solution of (4) for subproblem s. From Theorem 1 we have that γ^{*s} is continuously differentiable for $l \in \bar{\mathcal{L}}^s$. Then $\bar{\mathcal{L}} = \bigcap_{s=1}^{N_s} \bar{\mathcal{L}}^s$ is the set where $G(l)$ is continuously differentiable.

Let $l_k \in \mathcal{L}$ be a point given at iteration k of the Extended Newton Method. In an ordinary Newtons method the step Δl_k would be calculated by solving the following equation

$$H(l_k)\Delta l_k = -\alpha G(l_k) \tag{11}$$

Algorithm 1 Primal Decomposition of msMPC using an Extended Newtons method

```
Require: tolerance $\epsilon > 0$, step length $\alpha \in (0, 1]$, initial guess $l_k \leftarrow l_0$, $||\Delta l|| > \epsilon$
  while $||\Delta l|| > \epsilon$ do
    $\mathcal{D} = \{\}$
    for s in 1:$N_s$ do
      $\omega^{s*}, \lambda^{s*}, \mu^{s*}, \gamma^{s*}, \mathcal{A}_+^s, \mathcal{A}_-^s \leftarrow$ Solve (4) for $l = l_k$
      if $|\mathcal{A}_-^s| = 0$ then
        $\nabla_l \gamma^{s*}(l_k) \leftarrow$ solve (14)
      else
        $\mathcal{D} = \mathcal{D} \cup \{s\}$
      end if
    end for
    if $\sum_{s=1}^{N_s} |\mathcal{A}_-^s| > 0$ then
      for s in $\mathcal{D}$ do
        $\nabla_l \gamma^{s*}(l_k) \leftarrow J_L \gamma^{s*}(l_k, P)$ from Algorithm 1 from [18] with $P = I$
      end for
    end if
    $G(l_k) \leftarrow -\sum_{s=1}^{N_s} P_l^{sT} \gamma^{s*}(l_k)$ (7)
    $H(l_k) \leftarrow -\sum_{s=1}^{N_s} \nabla_l \gamma^{s*}(l_k) P_l^s$
    $\Delta l = -\alpha H(l_k)^{-1} G(l_k)$
    $l_k \leftarrow l_k + \Delta l$
  end while
```

where $\alpha \in (0, 1]$ is a step length and $H(l_k) = \nabla_l G(l_k)$ is the Jacobian of the gradient. Since the Jacobian only exist for $l_k \in \bar{\mathcal{L}}$ the Extended Newton Method for PC^1 functions instead uses $H(l_k) \in \partial_B G(l_k)$, where $\partial_B G(l_k)$ is the B-subdifferential ("Bouligand") of the gradient $G(l_k)$ [20].

$$\partial_B G(l_k) = \{M \in \mathbb{R}^{N_l \times N_l} : \exists (\bar{l}_k) \in \bar{\mathcal{L}} : \bar{l}_k \rightarrow l_k, \nabla_l G(\bar{l}_k) \rightarrow M\} \tag{12}$$

Note that the B-subdifferential becomes a singleton when the function is differentiable, i.e. $\partial_B G(l_k) = \nabla_l G(l_k)$ for $l_k \in \bar{\mathcal{L}}$.

The following theorem proved in [7] establishes quadratic convergence of the Extended Newton method.

Theorem 3 [7]: Let l^* be a solution for PC^1 system of equations $G(l) = 0$. Given that all elements of the B-subdifferential $\partial_B G(l)$ are nonsingular and Lipschitz continuous for $l \in \mathcal{L}$, then there exists a positive number r such that if $||l_0 - l^*|| \leq r$ then the sequence of steps generated by the Extended Newton method converge to l^* quadratically, where l_0 is the starting point.

The main result of this paper is given by the following theorem.

Theorem 4 (Main Result): Let l^* be the optimal solution of the coordinator problem (5). If assumption 1 and 2 hold and all members of the B-subdifferential $\partial_B G(l)$ are nonsingular and Lipschitz continuous, then there is a positive number r such that if $||l_0 - l^*|| \leq r$ Algorithm 1 converge quadratically to l^*.

Proof: Invoking Theorem 1 and 2, l^* is found by solving a PC^1 system of equations $G(l) = 0$. Algorithm 1 ensures a member of the B-subdifferential $\partial_B G(l)$ is used in the Extended Newton method, the rest follows from Theorem 3.

B. Computing the Derivatives (Smooth Case)

For $l \in \bar{\mathcal{L}}$ the Jacobian can be calculated as

$$H(l_k) = -\sum_{s=1}^{N_s} \nabla_l \gamma^{s*}(l_k) P_l^s. \tag{13}$$

When SC holds the sensitivity $\nabla_l \gamma^{s*}(l)$ can be found by applying the implicit function theorem to the KKT conditions (10) to obtain and solve the following linear system of equations,

$$\begin{bmatrix} \nabla^2_{\omega^s} \mathcal{L}(z^{s*}(l_k)) & \nabla_{\omega^s} h^s(\omega^{s*}) & \nabla_{\omega^s} g_+^s(\omega^{s*}) & P_\omega^T \\ \nabla_{\omega^s} h^{sT}(\omega^{s*}, l_k) & 0 & 0 & 0 \\ \nabla_{\omega^s} g_+^{sT}(\omega^{*s}, l_k) & 0 & 0 & 0 \\ P_\omega & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} \nabla_l \omega^{s*T} \\ \nabla_l \lambda^{s*T} \\ \nabla_l \mu_+^{s*T} \\ \nabla_l \gamma^{s*T} \end{bmatrix} = -\begin{bmatrix} 0 \\ 0 \\ 0 \\ -P_l^s \end{bmatrix}, \tag{14}$$

where g_+ and μ_+ are the vectors of strongly active inequality constraints and their related dual variables.

C. Computing the Derivatives (Nonsmooth Case)

When $l_k \in \mathcal{L} \backslash \bar{\mathcal{L}}$ a member of the B-subdifferential of $G(l_k)$ must be computed. This is not a trivial task as B-subdifferentials do not obey strict calculus rules, e.g. if a member of the B-subdifferential sets $\nabla_l \gamma^{s*}(l_k) \in \partial_B \gamma^{s*}(l_k)$ were found for each scenario $s \in \{1, \ldots, N_s\}$ then $-\sum_{s=1}^{N_s} \nabla_l \gamma^{s*}(l_k) P_l^s$ would not necessarily be in $\partial_B G(l_k)$.

To find a member of the B-subdifferential lexicographical derivatives will be used. A Lexicographical derivative is found for some full rank direction matrix $P \in \mathbb{R}^{N_l \times N_l}$, and for PC^1 functions is an element of the B-subdifferential $J_L G(l_k, P) \in \partial_B G(l_k)$ [18]. In addition, strict calculus rules holds for lexicographical derivative meaning that

$$J_L G(l_k, P) = -\sum_{s=1}^{N_s} J_L \gamma^{s*}(l_k, P) P_l^s. \tag{15}$$

The lexicographical derivatives of the optimal dual variables $J_L \gamma^{s*}(l_k, P)$ can be found by algorithm 1 from [18]. This means that $H(l_k) = J_L G(l_k, P)$ from (15) is used in the extended Newton method when $l_k \in \mathcal{L} \backslash \bar{\mathcal{L}}$. Note that $J_L \gamma^{s*}(l_k, P) = \nabla_l \gamma^{s*}(l_k)$ for $l \in \bar{\mathcal{L}}^s$ for $s \in \{1, \ldots, N_s\}$ meaning equation (14) can be used for subproblems with no weakly active inequality constraints.

IV. Case Study

The proposed algorithm is demonstrated in an example and compared to the algorithm used in [10] which used the Steepest Descent method. The algorithm is applied to a uncertain gas-lift well network. The system consists of two wells, both with their own uncertain gas-oil-ratio GOR connected to a common manifold. The manifold is connected to a riser. The schematic figure of the system is seen in fig 2. The goal is to produce as much oil w_{po} as possible while keeping the gas production rate w_{pg} below the max capacity of the topside facility. This is done by injecting gas ,$w_{gl,1}, w_{gl,2}$, into the two wells. More details about the system as well as the dynamic model of the system can be found in [9].

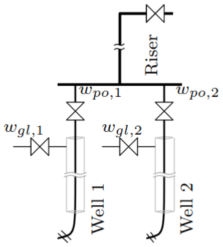

Fig. 2. A schematic figure of the gas-lift system [10].

TABLE I

UNCERTAIN PARAMETERS BOUNDS

	low	nominal	high
$GOR1$	0.06	0.10	0.14
$GOR2$	0.11	0.12	0.13

The uncertain parameters being the GOR in the two wells are assumed to take on values in the bounds between a low and high value given in Table I, as well as the nominal value for the GOR.

A. MPC setup

In the model predictive controller the following cost function is used per scenario,

$$\sum_{k=0}^{N-1} -\$_o w_{po,k} + \gamma_{pg}(w_{pg,k} - \hat{w}_{pg,max})^2 \tag{16}$$
$$+ \gamma_r (w_{gl,1,k+1} - w_{gl,1,k})^2 + \gamma_r (w_{gl,2,k+1} - w_{gl,2,k})^2$$

where $\$_o$ is the price of oil, γ_{pg} and $\hat{w}_{pg,max}$ are a parameters to track the maximum gas production rate to prevent it from exceeding the max capacity, and γ_r is a parameter penalizing large changes in the inputs.

To design the scenarios for the msMPC we considered the cases where parameters take maximum and minimum values, as well as the midpoint (nominal value) for 2 parameters, this leads to $M = 5$ realizations. A robust horizon of $N_r = 1$ is used leading to $N_s = M^{N_r} = 5$ scenarios in the msMPC.

A prediction horizon $N = 24$ is used in the msMPC with a sampling time of 300 seconds. The dynamic model is discretized using a 3rd order Gauss-Radau collocation scheme. The nonlinear programs are embedded using JuMP [2] and solved using Ipopt [21].

In the primal decomposition algorithm the coordinator variables l are warm started using the solution of the previous iteration. For the steepest decent algorithm $\alpha = 3\times10^{-5}$ was used as the step length. For the Extended Newton method $\alpha = 1$ was used as the step length. Both the Steepest Decent and Newton algorithms were terminated when the change in the coordinator variables were $||\Delta l||_\infty < 10^{-4}$.

B. Simulation setup

The gas-lift well network system was simulated for 5 hours. In the first 50 minutes, the GOR for both wells remains constant. In the next 200 minutes, the GOR for the two wells changes, while they remain constant the last 50 minutes. Both decentralized schemes are compared to the centralized approach.

V. Simulation Results and Discussion

The simulation results are shown in Figure 3, where the total produced oil rate is shown at the top. The gas-lift injection rates for the two wells are plotted beneath. From the figure, both the Steepest Descent and Extended Newton methods give almost identical results. A comparison with the centralized msMPC is plotted in Figure 4. The absolute error between the centralized and decentralized schemes is shown for both the total produced oil rate and gas-lift injection rates. As can be seen both the Steepest Decent and Extended Newton schemes converges to the same solution as the centralized scheme, but the Steepest Decent method introduces a small error.

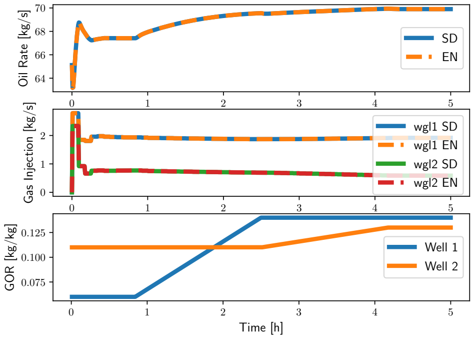

Fig. 3. Simulation results of the decomposed msMPC schemes.

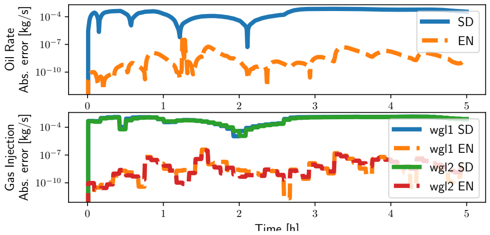

Fig. 4. Comparison of decomposed msMPC schemes with centralized approach.

In Figure 5 the required amount of iterations to converge is plotted for both the decentralized schemes. As can be seen the proposed algorithm converges in much fewer iterations than the SD algorithm. This holds especially true in the transient region at the start where the amount of necessary iterations is approximately reduced by a factor of 10. We chose to present the computational performance results in terms of iterations and not in terms of time, because the actual running time is dependent on many other factors than the algorithm itself.

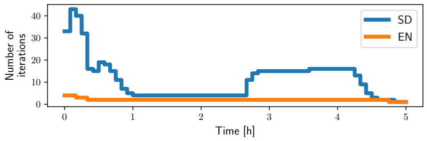

Fig. 5. Comparison of number iterations to converge for the decentralized msMPC schemes.

Both in terms of computational time and accuracy the Extended Newton method outperforms the Steepest Descent method which is expected as the Extended Newton method has local quadratic convergence while Steepest Decent only has linear convergence. Since the coordinator algorithm is warm started with the solution of the previous iteration the algorithm might already start in the neighborhood of the solution leading to fast convergence. Another benefit of the Extended Newton method is the step is scaled meaning it is easy to find a stable step size $\alpha = 1$. This is not the case for the Steepest Descent method where a step size that both makes fast progress as well as being stable is difficult to find. In fact a better step size could probably have been found for this simulation study.

VI. Discussion and Conclusion

The main computational load of the proposed algorithm is to solve the distributed subproblems, however, an extra computational load is introduced by the proposed algorithm. That is computing the Jacobian matrix used in the Extended Newton step. For nondegenerate, subproblems this involves solving equation (14) which can potentially be a large linear system of equations, which can be expensive. If the problems are solved using Ipopt the matrix factorization of the KKT matrix in equation (14) can be reused to calculate the Jacobian fast with sIpopt [17]. This extra computational cost should be negligible compared to solving the subproblems. For degenerate subproblems on the other hand a series of QPs has to be solved as outlined in [18], which might be computationally expensive.

For nonconvex problems the proposed decomposition algorithm can converge to any stationary point of the coordinator problem and not necessarily a minimum. This is known property of primal decomposition. However it will converge there faster (quadratically) than steepest descent (linearly) and in the simulations conducted in this paper it was found to convere to same optimum as the centralized solution.

A related issue when applying primal decomposition to solve the msMPC problem is that the continuity properties of the subproblems only hold with respect to a local solution. The subproblems are nonlinear programs meaning they are likely non-convex. This means each subproblem might have several local solutions. To avoid this problem we suggest adding regularization to the cost function to force one local solution to be preferential to the others. We also suggest warm-starting each subproblem with the previous iterations solution so they start in a neighborhood of the same local solution.

Another potential challenge with using primal decomposition is that coordinator variables l might lead to the subproblems (4) being infeasible. This was addressed in [11], where it was proposed to use the worst case predicted control action from the previous iteration to initialize the coordinator variables l and to use feasibility ensuring backtracking algorithm to make sure the updates are feasible.

References

[1] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[2] I. Dunning, J. Huchette, and M. Lubin. Jump: A modeling language for mathematical optimization. *SIAM Rev.*, 59:295–320, 2017.

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

[4] K. Jittorntrum. *Solution point differentiability without strict complementarity in nonlinear programming*, pages 127–138. Springer Berlin Heidelberg, Berlin, Heidelberg, 1984.

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

[6] E. Klintberg and S. Gros. A parallelizable interior point method for two-stage robust mpc. *IEEE Transactions on Control Systems Technology*, 25(6):2087–2097, 2017.

[7] M. Kojima and S. Shindo. Extension of newton and quasi-newton methods to systems of pc1 equations. *Journal of the Operations Research Society of Japan*, 29(4):352–375, 1986.

[8] D. Kouzoupis, E. Klintberg, M. Diehl, and S. Gros. A dual newton strategy for scenario decomposition in robust multistage mpc. *International Journal of Robust and Nonlinear Control*, 28(6):2340–2355, 2018.

[9] D. Krishnamoorthy, B. Foss, and S. Skogestad. Real-time optimization under uncertainty applied to a gas lifted well network. *Processes*, 4(4), 2016.

[10] D. Krishnamoorthy, B. Foss, and S. Skogestad. A distributed algorithm for scenario-based model predictive control using primal decomposition. *IFAC-PapersOnLine*, 51(18):351–356, 2018.

[11] D. Krishnamoorthy, B. Foss, and S. Skogestad. A primal decomposition algorithm for distributed multistage scenario model predictive control. *Journal of Process Control*, 81:162–171, 2019.

[12] D. Krishnamoorthy, E. Suwartadi, B. Foss, S. Skogestad, and J. Jäschke. Improving scenario decomposition for multistage mpc using a sensitivity-based path-following algorithm. *IEEE Control Systems Letters*, 2(4):581–586, 2018.

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

[14] 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*, 23(9):1306–1319, 2013.

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

[16] 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 and Chemical Engineering*, 79:30–45, 2015.

[17] H. Pirnay, R. López-Negrete, and L. T. Biegler. Optimal sensitivity based on ipopt. *Mathematical Programming Computation*, 4(4):307–331, 2012.

[18] P. Stechlinski, J. Jäschke, and P. I. Barton. Generalized sensitivity analysis of nonlinear programs using a sequence of quadratic programs. *Optimization*, 68(2-3):485–508, 2019.

[19] M. Thombre, Z. J. Yu, J. Jäschke, and L. T. Biegler. Sensitivity-assisted multistage nonlinear model predictive control: Robustness, stability and computational efficiency. *Computers and Chemical Engineering*, 148:107269, 2021.

[20] M. Ulbrich. *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems in Function Spaces*. Society for Industrial and Applied Mathematics, 2011.

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

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