Preprints, 10th IFAC International Symposium on
Advanced Control of Chemical Processes
Shenyang, Liaoning, China, July 25-27, 2018

A Distributed Algorithm for Scenario-based Model Predictive Control using Primal Decomposition ⋆

Dinesh Krishnamoorthy * **Bjarne Foss** ** **Sigurd Skogestad** *

* *Dept. of Chemical Engineering, Norwegian Univ. of Science & Technology (NTNU) , NO-7491 Trondheim , (e-mail: dinesh.krishnamoorthy@ntnu.no, skoge@ntnu.no).*
** *Dept. of Engineering Cybernetics, Norwegian Univ. of Science & Technology (NTNU), NO-7491 Trondheim , (e-mail: bjarne.foss@ntnu.no)*

Abstract: In this paper, we consider the decomposition of scenario-based model predictive control problem. Scenario MPC explicitly considers the concept of recourse by representing the evolution of uncertainty by a discrete scenario tree, which can result in large optimization problems. Due to the inherent nature of the scenario tree, the problem can be decomposed into each scenario. The different subproblems are only coupled via the non-anticipativity constraints which ensures that the first control input is the same for all the scenarios. This constraint is relaxed in the dual decomposition approaches, which may lead to infeasibility of the non-anticipativity constraints if the master problem does not converge within the required time. In this paper, we present an alternative approach using primal decomposition which ensures feasibility of the non-anticipativity constraints throughout the iterations. The proposed method is demonstrated using gas-lift optimization as case study.

Keywords: Scenario Optimization, Primal decomposition, Uncertainty, Distributed optimization

1. INTRODUCTION

Model predictive control (MPC) has proven to be a highly successful control methodology in the process control industry due to its ability to handle large and complex multivariable systems, subject to process and operating constraints. MPC typically uses models that represents the system and computes an optimal input trajectory based on model predictions in order to minimize a certain cost function over the prediction horizon. Recently, there has been an increasing trend in the use of *Economic NMPC*, where the economic objectives are incorporated into the MPC problem.

The presence of plant-model mismatch or process variations can easily lead to constraint violations or suboptimal operation. Different approaches have been proposed in the literature to handle uncertainty in the MPC problem, such as min-max MPC (Campo and Morari, 1987), which computes an optimal input trajectory that minimizes the cost of the worst-case realization of the uncertainty. This, however, leads to a very conservative solution, since the optimization is performed in an open-loop fashion. It ignores the fact that new information will be available and a new control trajectory will be re-computed in the future. In other words, min-max MPC ignores one of the important aspect of uncertainty handling, namely, *feedback*. Feedback min-max MPC scheme was proposed by Scokaert and Mayne (1998) to overcome the limitations of the open-loop min-max MPC. Feedback min-max MPC is a closed-loop optimization scheme, where the notion of feedback is explicitly taken into account by optimizing over different control policies rather than a single control trajectory by representing the evolution of the uncertainty by a scenario tree. This approach was later studied in detail for nonlinear systems in the context of multistage NMPC problem and was shown to reduce the conservativeness at the cost of computational time (Lucia et al., 2013a).

One of the main challenges of this method is that the computational size of the problem grows exponentially with 1) length of the prediction horizon, 2) number of uncertain parameters and disturbances and 3) number of discrete models for each uncertain variable that is considered in generating the different scenarios. This poses a challenge for real-time implementation, despite advancements in computational power and efficient numerical solvers.

One solution to this problem is to stop the branching after a certain number of samples in the prediction horizon (known as robust horizon) in order to curb the number of scenarios as described in Lucia et al. (2013a). Another solution is to exploit the fact that each scenario can be written as an independent subproblem except for the so-called non-anticipativity constraints. Hence decomposition methods can be employed by solving the subproblems

⋆ The authors gratefully acknowledge the financial support from SUBPRO, which is financed by the Research Council of Norway, major industry partners and NTNU.

independently and later use a master problem to co-ordinate the individual subproblems iteratively.

Scenario decomposition using dual decomposition was proposed by Lucia et al. (2013b) and Martí et al. (2015). Dual decomposition (also known as Lagrangian decomposition) method solves the subproblems by relaxing the coupling constraints. A master problem then co-ordinates the individual subproblems iteratively. The previously relaxed constraints are feasible only upon convergence. Martí et al. (2015) indicates that such methods require a relatively large number of iterations between the master problem and the subproblem to converge, leading to challenges with practical implementation. The use of augmented lagrangian methods can help improve the convergence properties, however this makes the problem non-separable (Boyd et al., 2011).

The risk of dual decomposition is then that the master problem may not converge within the required time. This leads to infeasibility of the non-anticipativity constraints, the implications of which are that the different subproblems may give different control inputs at the first sample time in the prediction horizon. This is not acceptable for real-time closed-loop implementation. In this paper, we propose an alternative approach to scenario decomposition using the primal decomposition approach which ensures the non-anticipativity constraints are always feasible. This is because, in contrast to dual decomposition, primal decomposition produces a primal feasible solution with monotonically decreasing objective value at each iteration.

The key challenge in any real-time implementation of optimizing controllers such as MPC is clearly, how best to deal with time. Quoting Kerrigan et al. (2015), "*The correctness of a computation is a function of time*". The late-arrival of a solution in many cases may simply not be acceptable. In real-time optimization, *approximate solution now is better than an accurate solution tomorrow.* This strategy is adopted in many optimization algorithms (Kerrigan et al., 2015). This is also the motivation to use primal decomposition as opposed to dual decomposition for the scenario MPC problem.

The paper is organized as follows. The framework of scenario MPC is introduced in section 2. The decomposition algorithm is presented in section 3. The proposed methodology is verified using a case study in section 4 before concluding the paper in section 5.

2. SCENARIO MPC

Consider a discrete-time nonlinear system of the form,

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

where, $\mathbf{x}_k \in \mathbb{R}^{n_x}$ denotes the state vector at time step k, $\mathbf{u}_k \in \mathbb{R}^{n_u}$ is the vector of control inputs and $\mathbf{d}_k \in \mathbb{R}^{n_d}$ represents the uncertain parameters and disturbances. Let us assume that the uncertainty belongs to a known distribution such that $\mathbf{d}_k \in \mathcal{U}\ \forall k$.

If the model (1) is *perfect*, the predicted state trajectory is given by $\mathbf{x}_{[k,k+N]}$ for the open-loop implementation of the corresponding input trajectory $\mathbf{u}_{[k,k+N-1]}$ over the prediction horizon $[k, k+N]$. However, in the presence of plant-model mismatch, $\mathbf{u}_{[k,k+N-1]}$ must be associated

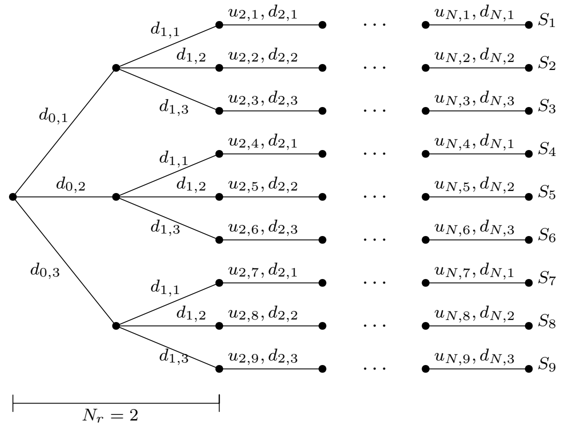

Fig. 1. Scenario Tree for $M = 3$ and $N_r = 2$.

with a cone of state trajectories $\{\mathbf{x}_{[k,k+N]}\}_{\mathcal{U}}$ depending on the realization of the uncertain variables (Guay et al., 2015). Optimizing over a single control trajectory (open-loop optimization) disregards feedback. In other words, it disregards the fact that new information will be available in the future and the control trajectory will be re-optimized. It may be prudent to optimize over different control policies rather than a single control trajectory, see Mayne (2014) and Mayne (2015). In other words, the optimization problem should compute a cone of possible control trajectories $\{\mathbf{u}_{[k,k+N-1]}\}_{\mathcal{U}}$ instead of a single control trajectory. A simple approach to solve this problem is to discretize the uncertainty space and represent the cone of trajectories as discrete scenarios. This is the basic principle behind scenario MPC. Scenario MPC (also known as multistage MPC or feedback min-max MPC) is thus a closed-loop optimization approach, where the evolution of the uncertainty is explicitly taken into account by modelling a tree of discrete scenarios as described by Scokaert and Mayne (1998). By doing so, we can considerably reduce the conservativeness of the solution compared to min-max methods that optimize over a single control trajectory (Lucia et al., 2013a).

To formulate the scenario MPC mathematically, the discrete-time nonlinear system (1) reads as,

$$\mathbf{x}_{k+1,j} = \mathbf{f}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \mathbf{d}_{k,j}) \tag{2}$$

where, the subscript $(\cdot)_{k,j}$ represents the j^{th} scenario at time step k.

The first step to building a scenario tree is to discretize the uncertainty space $\mathcal{U}$ to get M discrete realizations. A common practice is to consider a combination of nominal and extreme values to cover the overall uncertainty space, which has been shown to give good results in many different application examples, see Lucia et al. (2013a), Krishnamoorthy et al. (2017) and the references therein.

From the discrete realizations of the uncertainty, a scenario tree is generated as shown in Fig.1. Each scenario is defined as the path from root node to the leaf node. The number of scenarios resulting from the branching at each time step leads to exponential growth of the problem. A simple strategy to curb this is to stop the branching after a certain period of time N_r (known as robust horizon) as justified in Lucia et al. (2013a). The total number of scenarios is then given by $S = M^{N_r}$.

The resulting optimization problem is then written as,

$$\min_{\mathbf{x}_{k,j},\mathbf{u}_{k,j}} \sum_{j=1}^{S}\left[\omega_j \sum_{k=1}^{N}\mathbf{J}(\mathbf{x}_{k,j},\mathbf{u}_{k,j})\right] \tag{3a}$$

s.t

$$\mathbf{x}_{k+1,j} = \mathbf{f}(\mathbf{x}_{k,j},\mathbf{u}_{k,j},\mathbf{d}_{k,j}) \tag{3b}$$

$$\mathbf{g}(\mathbf{x}_{k,j},\mathbf{u}_{k,j}) \leq 0 \tag{3c}$$

$$\sum_{j=1}^{S}\bar{\mathbf{E}}_j\mathbf{u}_j = 0 \qquad \forall j \in \{1,\cdots,S\} \tag{3d}$$

where ω is the probability or weight for each scenario, $\mathbf{J}(\mathbf{x}_{k,j},\mathbf{u}_{k,j})$ is the cost function, $\mathbf{f}(\mathbf{x}_{k,j},\mathbf{u}_{k,j},\mathbf{d}_{k,j})$ is the system model, $\mathbf{g}(\mathbf{x}_{k,j},\mathbf{u}_{k,j})$ represents the nonlinear constraints. The constraints in (3d) are known as *non-anticipativity* or *causality* constraints which impose the fact that the future control inputs cannot anticipate the realization of the uncertainty. This 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 input for the j^{th} scenario, i.e. $\mathbf{u}_j = \left[\mathbf{u}_{0,j}^T \cdots \mathbf{u}_{N-1,j}^T\right]^T \in \mathbb{R}^{n_u N}$. To explain the notation of $\bar{\mathbf{E}}$, we first introduce the notation:

$$p = n_u \sum_{j=1}^{S-1} n_{c,(j,j+1)} \tag{4}$$

where $n_{c,(j,j+1)}$ represents the number of common nodes for two consecutive scenarios j and $j+1$ in the scenario tree (Klintberg et al., 2016). The matrices $\bar{\mathbf{E}}_j \in \mathbb{R}^{p\times n_u N}$ can then be given as,

$$\bar{\mathbf{E}} = \left[\begin{array}{c|c|c|c|c} E_{1,2} & -E_{1,2} & & & \\ & E_{2,3} & -E_{2,3} & & \\ & & \ddots & \ddots & \\ & & & E_{S-1,S} & -E_{S-1,S} \end{array}\right] = \left[\,\bar{\mathbf{E}}_1 \middle| \bar{\mathbf{E}}_2 \middle| \cdots \middle| \bar{\mathbf{E}}_S\,\right] \tag{5}$$

where

$$E_{j,j+1} = \left[\begin{array}{ccc|c} I_{n_u} & & & \\ & \ddots & & \mathbf{0} \\ & & I_{n_u} & \end{array}\right] \in \mathbb{R}^{n_u n_{c,(j,j+1)}\times n_u N} \tag{6}$$

and $\mathbf{0} \in \mathbb{R}^{n_u n_{c,(j,j+1)}\times n_u (N-N_r)}$ is a zero matrix. Using such a chain structure for the non-anticipativity constraints results in sparse structures, which can be an added advantage (Klintberg et al., 2016).

3. SCENARIO DECOMPOSITION

As described above, the different scenarios are independent except for the non-anticipativity constraints, which couple the different scenarios together. To this end, the different scenarios are easily separable. Different decomposition strategies exists that facilitates efficient solutions of such large scale optimization problems by decomposing them into smaller subproblems. This way the different subproblems can be parallelized. A master problem is then employed to co-ordinate the coupling constraints, (Bertsekas, 1999).

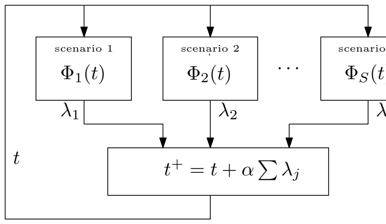

Fig. 2. Block diagram showing the structure information flow between the subproblems and the master problem.

3.1 Lagrangian Decomposition

In Lagrangian decomposition, the dual variables λ corresponding to the non-anticipativity constraints are used to define the Langrange function,

$$\mathcal{L}(\mathbf{x},\mathbf{u},\lambda) = \sum_{j=1}^{S}\left[\omega_j \sum_{k=1}^{N}\mathbf{J}(\mathbf{x}_{k,j},\mathbf{u}_{k,j})\right] + \lambda^T\sum_{j=1}^{S}\bar{\mathbf{E}}_j\mathbf{u}_j \tag{7}$$

Since $\mathcal{L}(\mathbf{x},\mathbf{u},\lambda)$ is separable in $\mathbf{x}$ and $\mathbf{u}$, each scenario can be solved independently by relaxing the non-anticipativity constraints as shown (Klintberg et al., 2016),

$$\mathcal{L}_j(\mathbf{x}_j,\mathbf{u}_j,\lambda) = \omega_j\sum_{k=1}^{N}\mathbf{J}(\mathbf{x}_{k,j},\mathbf{u}_{k,j}) + \lambda^T\bar{\mathbf{E}}_j\mathbf{u}_j \tag{8}$$

Note that (7) and (8) are also subject to the system model (3b) and nonlinear constraints (3c) for all $j \in \{1,\ldots,S\}$. The master problem iterates on λ and the non-anticipativity constraints become feasible only upon convergence of λ. Different forms of augmented Lagrangian decomposition methods have also been introduced in Martí et al. (2015), where an additional quadratic penalty term is added to (7) to improve the convergence properties. However, this makes the problem not separable in $\mathbf{x}$ and $\mathbf{u}$. The subproblems must then be solved sequentially using the Alternating Directions Method of Multipliers (ADMM) approach (Boyd et al., 2011). Solving the subproblems sequentially can then make the computation time longer for problems with large number of scenarios.

The relaxation of the non-anticipativity constraints in Lagrangian decomposition poses a challenge for real time implementation. In an MPC framework, the optimization problem is solved to compute the optimal control trajectory and the first control move in implemented in the plant in a receding horizon fashion. In scenario MPC, the non-anticipativity constraints ensure that the first control input is the same for all the scenarios. However, if λ fails to converge within the required sampling time, infeasibility of the non-anticipativity constraints would mean that the first control move provided by the different scenario subproblems may be different. This is not acceptable for closed-loop implementation. We therefore, provide an alternative approach using primal decomposition framework which always produces a primal feasible point with monotone-decreasing objective value at each iteration.

3.2 Primal Decomposition

Primal decomposition iterates directly on the shared variables (Bertsekas, 1999). This ensures that the non-anticipativity constraints are always feasible at any point

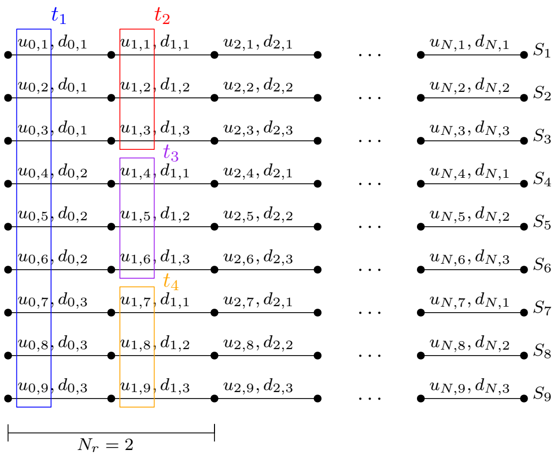

Fig. 3. Parallalized representation of the scenario tree for an uncertainty with 3 discrete realizations (M = 3) and a robust horizon of 2 samples ($N_r = 2$). The non-anticipativity constraints that couple the different scenarios are marked in different coloured boxes.

in time. Thus the first control move provided by all the scenario subproblems will be the same, enabling closed-loop implementation.

The subproblem for each scenario can be written by introducing a new auxiliary variable t_l,

$$\mathbf{\Phi}_j(t_l) = \min_{\mathbf{x}_{k,j},\mathbf{u}_{k,j}} \omega_j \sum_{k=1}^{N} \mathbf{J}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}) \tag{9a}$$

s.t

$$\mathbf{x}_{k+1,j} = \mathbf{f}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \mathbf{d}_j) \tag{9b}$$

$$\mathbf{g}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}) \leq 0 \tag{9c}$$

$$\bar{\mathbf{E}}_j \mathbf{u}_j = \bar{\mathbf{t}}_j \tag{9d}$$

where $\bar{\mathbf{t}}$ has a similar structure to $\bar{\mathbf{E}}_j$ as shown below,

$$\bar{\mathbf{t}} = \left[\begin{array}{c|c|c|c} t_{1,2} & -t_{1,2} & & \\ & t_{2,3} & -t_{2,3} & \\ & & \ddots & \ddots \\ & & & t_{S-1,S} & -t_{S-1,S} \end{array}\right] = [\,\bar{\mathbf{t}}_1 | \bar{\mathbf{t}}_2 | \cdots | \bar{\mathbf{t}}_S\,] \tag{10}$$

$t_{j,j+1} \in \mathbb{R}^{n_u n_{c,(j,j+1)}}$ is composed of auxiliary variables $t_l \in \mathbb{R}^{n_u}$. The index l can be given by the expression,

$$l \in \{1, \ldots, \sum_{m=1}^{N_r} M^{m-1}\} \tag{11}$$

The master problem is then written as,

$$\min_{t_l} \sum_{j=1}^{S} \Phi_j(t_l) \tag{12}$$

which simplifies to updating each t_l using the corresponding lagrange multipliers from the different subproblems as shown in Fig.2.

The generation of $t_{j,j+1}$ and the master problem update is illustrated using an example with $M = 3$ and $n_R = 2$. The corresponding scenario tree is shown in Fig.1 and the decomposed tree is shown in Fig.3. For such a tree, $l = 4$ and $t_l = \{t_1^T, \cdots, t_4^T\}$. Table 1 shows the $t_{j,j+1}$ or each scenario pair.

Each t_l is then updated in the master problem as shown,

$$t_1^+ = t_1 + \alpha_1(\lambda_{1,1} + \cdots + \lambda_{1,9}) \tag{13}$$

$$t_2^+ = t_2 + \alpha_2(\lambda_{2,1} + \cdots + \lambda_{2,3}) \tag{14}$$

$$t_3^+ = t_3 + \alpha_3(\lambda_{2,4} + \cdots + \lambda_{2,6}) \tag{15}$$

$$t_4^+ = t_4 + \alpha_4(\lambda_{2,6} + \cdots + \lambda_{2,9}) \tag{16}$$

where the subscripts of $\lambda_{k,j}$ represents the lagrange multiplier at sample instant k for the j^{th} scenario and α is a suitable step length. A simple stopping criteria for the iterations between the master problem and the subproblems could be when the change in t between two consecutive iterations is less than some small user-defined tolerance ϵ.

By introducing the auxiliary variables t_l, the first control input for all the scenarios is $u_{1,j} = t_1$. The master problem iterates to drive t_1 to the optimal input. In the case, where the master problem does not converge to the optimum within the required sampling time, the non-anticipativity constraints are still feasible, thus enabling closed-loop implementation. By warm-starting t_l in the subsequent time steps, the optimization problem is expected to eventually converge to the true optimum.

4. ILLUSTRATIVE EXAMPLE

4.1 Process description

The primal decomposition approach proposed above is implemented on an oil and gas production optimization problem. We consider a gas lifted well network consisting of 2 wells producing to a common manifold and a riser as shown in Fig.4. More detailed description of the system can be found in Krishnamoorthy et al. (2016) and the references therein.

The objective of the optimization problem is to find the optimum gas lift injection rates for the two wells such that the the profits from the oil production is maximized and the cost of gas compression is minimized. The gas-oil-ratio GOR_i for each well $i \in \{1, \cdots, n_w\}$, is assumed to be uncertain. The nominal value GOR_{0_i} and the variance σ_i are assumed to be known a-priori.

$$\min_{w_{gl}} \sum_{k=1}^{N} \left[-\$_o \sum_{i=1}^{n_w} w_{po,i_k} + \$_{gl} \sum_{i=1}^{n_w} w_{gl,i_k} \right] \tag{17a}$$

s.t.

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, GOR_{i_k}) \tag{17b}$$

$$GOR_i \in \{GOR_{0_i} \pm \sigma_i\} \tag{17c}$$

$$\forall i \in \{1, \cdots, n_w\}, k \in \{1, \cdots, N\}$$

where w_{po} is the oil production rate from each well, w_{gl} is the gas lift injection rate for each well, $n_w = 2$ is the number of wells, $\$_o$ and $\$_{gl}$ are the value of produced

Table 1. Construction of $t_{j,j+1}$ for the scenario tree in Fig.3.

$(j, j+1)$	$n_{c,(j,j+1)}$	$t_{j,j+1}$
(1,2)	2	$[t_1^T, t_2^T]^T$
(2,3)	2	$[t_1^T, t_2^T]^T$
(3,4)	1	$[t_1^T]$
(4,5)	2	$[t_1^T, t_3^t]^T$
(5,6)	2	$[t_1^T, t_3^t]^T$
(6,7)	1	$[t_1^T]$
(7,8)	2	$[t_1^T, t_4^t]^T$
(8,9)	2	$[t_1^T, t_4^t]^T$

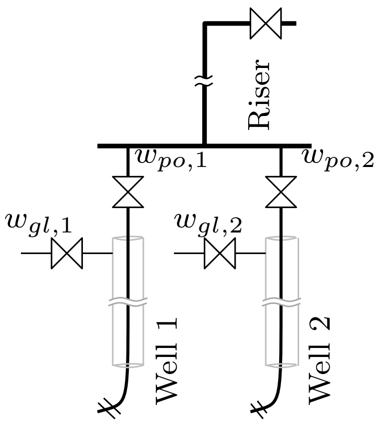

Fig. 4. Schematic of a gas lifted well network with 2 wells producing to a common riser manifold.

Table 2. The discrete realizations of GOR used in the optimizer

GOR well 1	0.08	0.10	0.12
GOR well 2	0.10	0.12	0.14

oil and cost of gas compression respectively. The system constraints are enforced in (17b).

The continuous time differential equations are discretized into (17b) using a third order direct collocation scheme in CasADi v3.0.1 (Andersson, 2013) using the MATLAB R2017a programming environment. The NLP problem is then solved using IPOPT version 3.12.2 running with mumps linear solver.

The dynamic optimization problem was solved with a prediction horizon of $N = 15$ and a sampling time of $T_s = 5min$. A robust horizon of $n_R = 2$ was chosen. M =3 discrete realizations of the uncertain parameter GOR chosen are shown in Table.2. For the scenario decomposition approach, the step length was fixed at $\alpha = [0.0001, 0.0002, 0.0002, 0.0002]^T$. The stopping criteria was defined as when the change in t between two consecutive iterations is less than $\epsilon = 0.001$.

4.2 Results and Discussion

In this section, the performance of the centralized approach and the distributed approach using primal decomposition as proposed above is compared using the case study described above. For the comparison of scenario MPC with nominal and worst case MPC for this problem, the reader is referred to Krishnamoorthy et al. (2017).

In the first simulation, we compare the centralized solution with the decomposed solution. The true realization of GOR for the cases is as shown in Fig.5. The total produced oil for the centralized and decomposed case are shown in top left subplot in Fig.5. The error between the centralized and decomposed solution is shown in top right subplot. The control input (gas lift injection rates for wells 1 and 2) for centralized and decomposed solution is plotted in the middle left subplot and the corresponding error is plotted in the middle right subplot. The number of iterations required for the scenario decomposition to converge at each time step is plotted in the bottom right subplot. From the simulation results, it can be seen that the primal decomposition approach provides similar solution as the centralized approach. Warm starting the problem at subsequent time steps reduced the number of iterations required to converge in the subsequent time steps. The average computation time for each subproblem was around 1s, as opposed to 11s for the centralized problem.

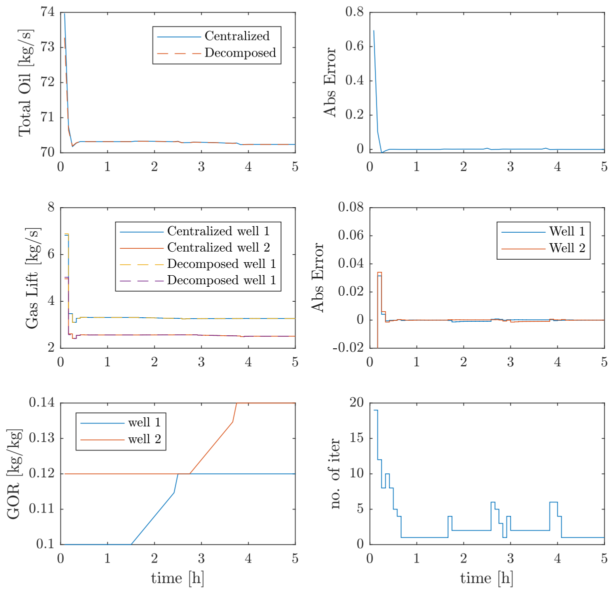

Fig. 5. Comparison of centralized approach and decomposition approach.

As mentioned earlier, the main advantage of primal decomposition over dual decomposition methods is when the master problem does not converge within a required sample time. This will lead to violation of the non-anticipativity constraints in dual decomposition, thus leading to closed-loop implementation issues. However, primal decomposition always ensures the feasibility of non-anticipativity constraints. From the results in Fig.5, it was seen that the number of iterations varied between 1 and 19 to converge. To emulate the case where the master problem has to be terminated before it converges fully, the number of iterations is capped at 5. The simulation setup is the same as the previous case. In the case of dual decomposition, prematurely stopping the iterations as done in this simulation will result in an infeasible solution, which causes implementation issues.

The results are shown in Fig.6. It can be clearly seen that the error between the centralized and decomposed approach is much larger during the first hour compared to the results in Fig.5. It can also be seen that the error becomes smaller over time, clearly showing the benefits of warm starting the master problem. The number of iterations required is also reduced to 1 when the change in GOR is constant for a period. This shows that if the the disturbance is not varying too much, the primal decomposition is able to converge to the true optimal solution despite terminating the master problem prematurely. A close look of the first 1 hour of simulation comparing the simulation with the number or iteration uncapped (Fig.5) and capped (Fig.6) is shown in Fig.7.

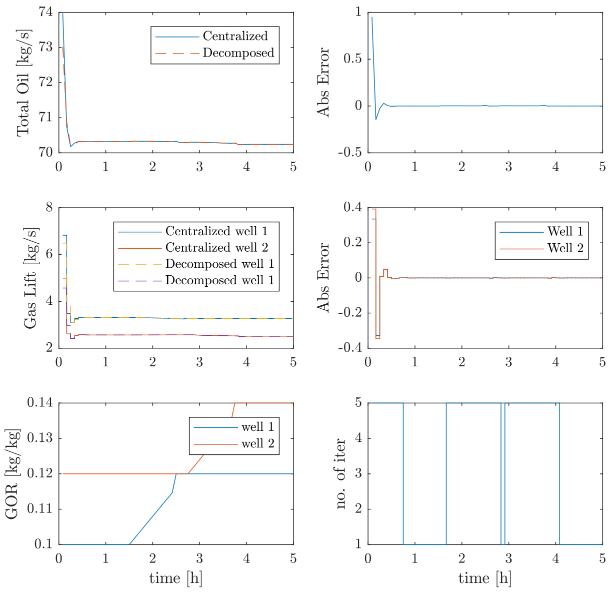

Fig. 6. Comparison of centralized approach and decomposition approach with the maximum number of iterations capped at 5.

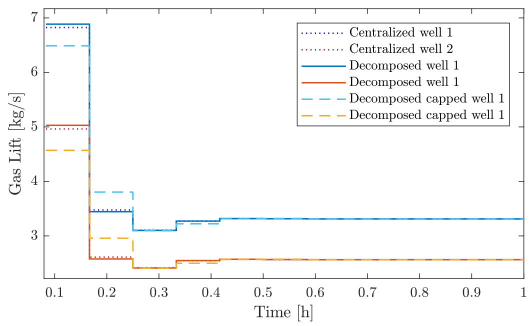

Fig. 7. Closer look at the first one hour of simulation to compare the centralized, decomposed, and decomposed with max iterations capped at 5.

5. CONCLUSION

In this paper, we presented an alternative approach to scenario decomposition using primal decomposition. The primal decomposition approach always ensures the feasibility of the non-anticipativity constraints, hence enabling closed-loop implementation, unlike dual decomposition methods. Warm-starting the master problem eventually leads to convergence over time. Primal decomposition approach may thus be an useful way to decompose scenario MPC for applications with higher sampling rates. The proposed method was tested on a gas lift optimization case study. The simulation results clearly demonstrates the benefit of primal decomposition approach for scenario decomposition. Simulation results show that the primal decomposition eventually converges to the solution of the centralized counterpart despite being terminated prematurely.

6. ACKNOWLEDGEMENTS

Fruitful discussions with Prof. Eduardo Camponogara, Dr.Thiago Lima and Marco Aguiar from the Federal University of Santa Catarina are gratefully acknowledged.

REFERENCES

Andersson, J. (2013). *A General-Purpose Software Framework for Dynamic Optimization*. PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering and Optimization in Engineering Center.

Bertsekas, D.P. (1999). *Nonlinear programming*. Athena scientific Belmont.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1), 1–122.

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

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

Kerrigan, E.C., Constantinides, G.A., Suardi, A., Picciau, A., and Khusainov, B. (2015). Computer architectures to close the loop in real-time optimization. In *Decision and Control (CDC), 2015 IEEE 54th Conference on*, 4597–4611. IEEE.

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

Krishnamoorthy, D., Foss, B., and Skogestad, S. (2017). Gas lift optimization under uncertainty. *Computer Aided Chemical Engineering*, 40, 1753–1758.

Krishnamoorthy, D., Foss, B., and Skogestad, S. (2016). Real time optimization under uncertainty - applied to gas lifted wells. *Processes*, 4(4). doi:10.3390/pr4040052.

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

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

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

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

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

Scokaert, P. and Mayne, D. (1998). Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic control*, 43(8), 1136–1142.