Multistage Model Predictive Control with Online Scenario Tree Update using Recursive Bayesian Weighting

Dinesh Krishnamoorthy, Sigurd Skogestad and Johannes Jäschke

Abstract— **This work deals with a nonlinear multistage model predictive control (MPC) formulation, where the future propagation of the uncertainty in the prediction horizon is represented via a discrete scenario tree. The scenario tree is often generated using finite realizations of the uncertainty sampled from an uncertainty set or a probability distribution function. Once the scenarios are chosen, the scenario tree is often kept fixed for all the iterations. In this paper, we propose to update the different discrete realizations of the uncertainty in the scenario tree using a recursive Bayesian weighting approach. We show that by gradually shrinking the uncertainty set, we can further reduce the conservativeness of the closed-loop solution. The effectiveness of the proposed method is demonstrated using an oil and gas production optimization case study.**

I. INTRODUCTION

Model predictive control is a popular optimal control method in the process industry due to its ability to handle multivariable constrained systems. However the performance of MPC is strongly affected by the quality of the prediction model used by the MPC. Models are almost always subject to uncertainties due to imperfect knowledge of the system or model simplification. In the presence of constraints, additional robustification must be introduced in order to ensure robust constraint satisfaction despite the uncertainty.

To this end, there has been several developments in min-max approaches [1], where the optimal input trajectory is computed by minimizing the cost of the worst case realization of the uncertainty. Although this ensures robust constraint satisfaction, this often leads to overly conservative and hence suboptimal solutions. This is because the optimization is performed for the worst-case scenario in an open-loop fashion without any notion of feedback.

In a recent review paper [2], the author argues that effective handling of uncertainty requires feedback models and hence the control trajectory computed by solving an open loop optimization problem is not optimal. Multi-stage scenario-based MPC, also known as feedback min-max MPC, is one such closed-loop optimization approach introduced in [3] and later extended to nonlinear systems in [4]. Here, the future evolution of the uncertainty in the prediction horizon is represented by a discrete scenario tree. Different control trajectories are then computed for the different scenarios. In other words, the multistage MPC formulation explicitly takes into account the fact that new information will become available in the future and new optimal control input will be re-computed. This can be envisaged as a player's decision making process in an evolutionary strategic game. Instead of preparing a single sequence of optimal moves, we compute several backup moves depending on the future evolution of the uncertainty throughout the game. This concept is commonly known as *recourse* in the stochastic programming literature and is an important property in optimal decision making under uncertainty.

The multistage scenario-based MPC formulation has been shown to be less conservative than traditional min-max approaches for various applications, see for e.g. [4],[5], [6], [7] to name a few. Nevertheless, the solution provided by the multistage MPC will be conservative in order to ensure robust constraint satisfaction for all the scenarios considered in the scenario tree. A common approach to choosing the discrete realizations of the uncertainty in the scenario tree is to use a combination of the maximum, minimum and nominal values of the uncertain parameters. If the assumed range of the uncertain parameter is large, the resulting span of the scenario tree is also large, and consequently the solution provided will be conservative, albeit less conservative than traditional min-max MPC.

For problems with constant but unknown parameters, i.e. time-invariant uncertain parameters, it may be desirable to approach the problem from an adaptive framework rather than a robust framework[8]. Typical adaptive control frameworks involve the use of parameter estimators that adapts the uncertain parameters online such that it converges asymptotically to the true system. However, developments in adaptive MPC have been rather limited. Parameter estimation algorithms also often requires the uncertain parameters to be observable from the measurements, which may not always be the case.

In this paper, we propose to adapt the uncertainty characteristics (i.e. the span of the uncertain parameters) instead of adapting the parameters directly. The idea of updating the uncertainty characteristics instead of using a parameter estimator itself is not entirely new. Similar ideas of adaptive robust approaches were also explored in [8], [9], [10], where the uncertainty set containing all possible values of the uncertain parameters are estimated instead of adapting the parameters directly. The different works used different approches to update the uncertinty sets, such as ensemble Kalman filter (EnKF) or set-based guranteed parameter estimation etc. In this paper, we propose an alternative appraoch

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

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

to updating the uncertainty set that does not requre the use of a parameter estimation algorithm. Instead, we propose to use the recursive Baysian probability to update the scenario tree.

Using the process measurements and the model predictions from the different scenarios, we compute and assign a Bayesian weight for the different scenarios. The Bayesian weights are then used to shrink the span of the scenario tree by updating the scenarios with a very low weight. In other words, by computing Bayesian weights for the different discrete uncertainty realizations, we gradually eliminate values from the uncertainty set that do not explain the observed measurements with sufficient likelihood. This results in shrinking the span of the scenario tree over time and hence further reduce the conservativeness from the standard multistage scenario-based MPC.

Recursive Bayseian weighting approach such as the one used in this work were also used in multiple model predictive control (MMPC) formulations in works such as [11] and [12] to interpolate between different models from a model bank. We apply a similar Bayesian weighting scheme and instead of interpolating between the different scenarios, we update the scenarios with very low weights. This is an intuitive approach to updating the scenario tree online based on the available measurements and model predictions.

The reminder of the paper is organized as follows. The multistage scenario-based robust MPC framework is introduced in Section 2. The proposed online scenario tree adaptation based on recursive Bayesian weighting approach is described in Section 3. The proposed method is demonstrated using an oil and gas production optimization problem in section 4 before concluding the paper in Section 5.

II. Background

Consider a discrete-time nonlinear system parameterized by a vector of *time-invariant* uncertain parameters $\mathbf{p} \in \mathbb{R}^{n_p}$,

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

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \tag{2}$$

where $\mathbf{x}_k \in \mathbb{R}^{n_x}$ and $\mathbf{u}_k \in \mathbb{R}^{n_u}$ denotes the state and input vectors, respectively. The system model is represented by $\mathbf{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_x}$. The vector of available measurements is denoted by $\mathbf{y} \in \mathbb{R}^{n_y}$ given by the measurement model $\mathbf{h} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_y}$.

The objective is to minimize a cost function $J : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}$ subject to the nonlinear inequality constraints $\mathbf{g} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_c}$ over a prediction horizon of length N. The optimal control problem can then be written as,

$$\min_{\mathbf{x}_k, \mathbf{u}_k} \quad \sum_{k=0}^{N-1} J(\mathbf{x}_k, \mathbf{u}_k) \tag{3a}$$

s.t.

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

$$\mathbf{g}(\mathbf{x}_k, \mathbf{u_k}) \leq 0 \tag{3c}$$

$$\mathbf{x}_0 = \hat{\boldsymbol{x}} \tag{3d}$$

$$\mathbf{p} \in \mathcal{U}, \quad \forall k \in \{0, \dots, N-1\}$$

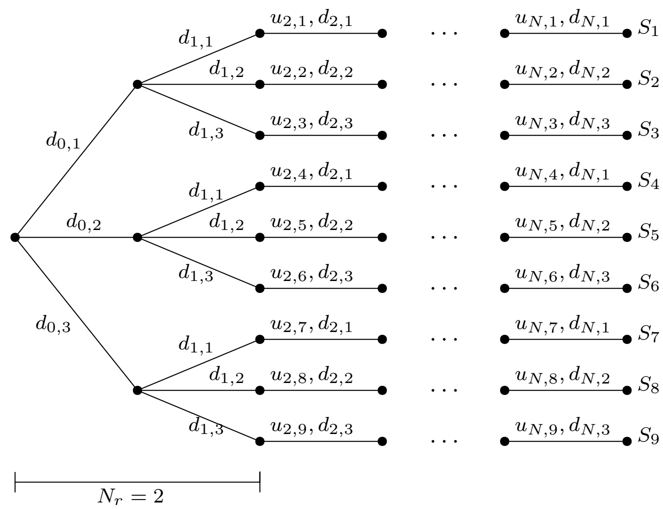


Fig. 1: Schematic representation of the scenario tree with $M = 3$ discrete realizations of the uncertainty and a robust horizon of $N_r = 2$, leading to $S = 9$ scenarios.

where $\mathcal{U} \subset \mathbb{R}^{n_p}$ denotes the bounded uncertainty characteristics, either in the form of uncertainty set or probability distribution function. Initial conditions are enforced in (3d), where $\hat{\boldsymbol{x}}$ denotes the current state estimates.

If the model was *perfect* and $\mathbf{p}$ was known accurately, then for an optimal input trajectory $\mathbf{u}^p_{[t,t+N]}$, the predicted state trajectory is given by $\mathbf{x}^p_{[t,t+N]}$. However, in the presence of uncertainty, an optimal input trajectory $\mathbf{u}^p_{[t,t+N]}$ would give rise to a cone of state trajectories $\{\mathbf{x}^p_{[t,t+N]}\}_{\mathcal{U}}$ depending on the value of the uncertain parameter $\mathbf{p} \in \mathcal{U}$. Optimizing over a single control trajectory $\mathbf{u}^p_{[t,t+N]}$ ignores the fact that new information will be made available at the next time step and a new optimal input trajectory will be re-computed. In other words, the optimization is performed in an open-loop fashion (although the implementation may be in closed-loop, if the optimal control problem is re-solved at each sampling time with only the first control input move implemented on the process). Closed-loop optimization, on the other hand, involves computing a cone of possible control trajectories $\{\mathbf{u}^p_{[t,t+N]}\}_{\mathcal{U}}$ instead of a single control trajectory $\mathbf{u}^p_{[t,t+N]}$, thereby introducing *recourse* action.

A simple approach to closed-loop optimization is by discretizing the uncertainty characteristics and solving the sampled average approximate problem as explained in [3] and [4]. Therefore the first step to designing a multistage scenario-based MPC is to select the discrete realizations of the uncertainty from the uncertainty set. In order to ensure robust constraint satisfaction for any realization of the uncertainty from the uncertainty set, a combination of maximum, minimum and nominal values of the different uncertain parameters are often chosen as the scenarios.

To this end, M discrete realizations of the uncertain parameters are sampled from the uncertainty set $\mathcal{U}$ to generate a scenario tree as shown in Fig.1. In order to prevent the exponential growth of the problem, the scenario branching is often terminated after a certain number of samples N_r in the prediction known as robust horizon, as justified in [4]. This results in $S = M^{N_r}$ number of scenarios. The resulting scenario optimization problem can be written as,

$$\min_{\mathbf{x}_{k,j},\mathbf{u}_{k,j}} \quad \sum_{j=1}^{S} \omega_j \sum_{k=1}^{N} J(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}) \tag{4a}$$

s.t.

$$\mathbf{x}_{k+1,j} = \mathbf{f}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \mathbf{p}_j) \tag{4b}$$

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

$$\mathbf{x}_0 = \hat{\boldsymbol{x}} \tag{4d}$$

$$\sum_{j=1}^{S} \bar{\mathbf{E}}_j \mathbf{u}_j = 0 \tag{4e}$$

$$\forall j \in \{1, \cdots, S\} \; \forall k \in \{0, \ldots, N-1\}$$

where ω_j represents the weight for the different scenarios. The constraints in (4e) represents the non-anticipativity or causality constraints which enforce the fact that the control inputs cannot anticipate the future realization of the uncertainty. In other words, the states that branch from the same parent node must have the same control inputs. Here $\mathbf{u}_j$ represents the sequence of optimal control input for the j^{th} scenario, i.e. $\mathbf{u}_j = \left[\mathbf{u}_{0,j}^T \cdots \mathbf{u}_{N-1,j}^T\right]^T \in \mathbb{R}^{n_u N}$. The reader is referred to [13] and [14] for detailed description of the non-anticipativity constraints and the structure of $\bar{\mathbf{E}}_j$ used in (4e).

III. Multistage MPC with Online Scenario Update

As mentioned earlier, often a combination of maximum, minimum and nominal values of the different parameters are chosen as the different uncertainty realizations in the scenario tree. If the uncertainty parameter range is rather large, then the scenario tree has a large span resulting in a conservative solution. A common assumption in most works considering multistage scenario MPC is that, once the scenarios are selected, the scenario tree remains fixed. In the case of *time-invariant* uncertain parameters, an adaptive framework may be preferable that enables the controller to improve its performance over time by employing some adaptive mechanism to eliminate/update scenarios that do not explain the observations with sufficient likelihood.

The proposed multistage model predictive control method with online scenario tree update is based on adapting the scenario tree online using the available measurements $\mathbf{y}^m \in \mathbb{R}^{n_y}$. A recursive Bayesian weighting scheme is used to assign weights to the different scenarios. The scenarios that do not explain the observations with sufficient likelihood (represented by very low weights) are then updated online to reduce the span of the scenario tree.

A. The recursive Bayesian weighting scheme

In this paper, we use a probabilistic weighting scheme which assigns weights to each scenario, which is a value ranging from 0 to 1. This is based on the conditional probability of the j^{th} scenario being the true realization of the uncertainty given the past history of residuals and probabilities of all the scenarios. The recursive Bayes theorem for the j^{th} scenario at time step k is then given by,

$$P_{k,j} = \frac{e^{-0.5\boldsymbol{\epsilon}_{k,j}^T K \boldsymbol{\epsilon}_{k,j}} P_{k-1,j}}{\sum_{m=1}^{S} e^{-0.5\boldsymbol{\epsilon}_{k,m}^T K \boldsymbol{\epsilon}_{k,m}} P_{k-1,m}} \tag{5}$$

where the residual $\boldsymbol{\epsilon}_{k,j} \in \mathbb{R}^{n_y}$ for the j^{th} scenario at the current time step k is computed using the observations $\mathbf{y}^m$ and the model predictions (2) for the j^{th} scenario,

$$\boldsymbol{\epsilon}_{k,j} = \mathbf{y}_k^m - \mathbf{h}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}) \tag{6}$$

Here, K is a weighting matrix, which is typically chosen to be diagonal and can be seen as the inverse of the residual covariance. Hence, large values of K often leads to faster convergence towards a scenario and smaller values of K lead to a more averaging approach to the scenarios. Using the recursive probabilities (5), the weight for the j^{th} scenario at time step k are then computed as shown below,

$$W_{k,j} = \frac{P_{k,j}}{\sum_{m=1}^{S} P_{k,m}} \tag{7}$$

B. Online scenario tree update

Using the Bayesian weights, we can update the scenario tree online. We find the scenario corresponding to the smallest weight represented by $\underline{j}$ and find the scenario corresponding to the largest Bayesian weight represented by $\bar{j}$. If the Bayesian weight for the $\underline{j}^{\text{th}}$ scenario becomes lower than a user defined threshold δ, then the least likely $\underline{j}^{\text{th}}$ scenario is updated by moving it towards the most likely $\bar{j}^{\text{th}}$ scenario. For example, if the $\underline{j}^{\text{th}}$ scenario has a very low weight $W_{k,\underline{j}} < \delta$ relative to the $\bar{j}^{\text{th}}$ scenario, then the $\underline{j}^{\text{th}}$ scenario is updated in the direction of the most likely scenario using a user defined step length $\alpha < 1$ as shown below,

$$\mathbf{p}_{\underline{j}} \leftarrow \mathbf{p}_{\underline{j}} + \alpha(\mathbf{p}_{\bar{j}} - \mathbf{p}_{\underline{j}}) \tag{8}$$

The step length α must be sufficiently small (typically in the range of 0.1) to retain the originally envisioned robustness properties. Hence, at the next time step $k+1$, the $\underline{j}^{\text{th}}$ scenario in the scenario tree is updated according to (8) and the scenario MPC problem (4) is solved using the updated scenario tree. Note that, the threshold δ must be chosen such that the likelihood of the $\underline{j}^{\text{th}}$ scenario explaining the observations must be sufficiently low. This is to avoid updating a scenario which only has a marginally lower Bayesian weight relative to the other scenarios. Since we have used a recursive weighting scheme, it is also crucial that the weights are reset after each time the scenarios are updated. This is because the new weight of the updated scenario must not be based on the probability of the old scenario realization. A sketch of the proposed multistage MPC with online scenario tree update scheme is described in Algorithm 1. The algorithm presented here to compute the recursive Baysiean weights is computationally inexpensive and is known to reject poor models exponentially fast [11].

Algorithm 1 Multistage MPC framework with online scenario tree update

Define tolerance $\delta > 0$, $\alpha < 1$, initial probability for each scenario $P_{0,j} = \frac{1}{S}, \quad \forall j \in \{1, \dots, S\}$.

Input: At each time step, observations $\mathbf{y}_k^m$ and model predictions $\hat{\mathbf{y}}_{k,j}$

for $j = 1, 2, \dots, S$ **do**

$\boldsymbol{\epsilon}_{k,j} \leftarrow \mathbf{y}_k^m - \hat{\mathbf{y}}_{k,j}$

$P_{k,j} \leftarrow \frac{e^{-0.5\boldsymbol{\epsilon}_{k,j}^T K \boldsymbol{\epsilon}_{k,j}} P_{k-1,j}}{\sum_{m=1}^{S} e^{-0.5\boldsymbol{\epsilon}_{k,m}^T K \boldsymbol{\epsilon}_{k,m}} P_{k-1,m}}$

$W_{k,j} \leftarrow \frac{P_{k,j}}{\sum_{m=1}^{S} P_{k,m}}$

end for

Find scenario corresponding to smallest weight $\underline{j} = \arg\min_j \quad W_{k,j}$

Find scenario corresponding to largest weight $\bar{j} = \arg\max_j \quad W_{k,j}$

if $W_{k,\underline{j}} < \delta$ **then**

Update $\underline{j}^{th}$ scenario $\mathbf{p}_{\underline{j}} \leftarrow \mathbf{p}_{\underline{j}} + \alpha(\mathbf{p}_{\bar{j}} - \mathbf{p}_{\underline{j}})$

Reset probability $P_{k,j} \leftarrow \frac{1}{S}, \quad \forall j \in \{1, \dots, S\}$

end if

$[\mathbf{x}_{k,j}^*, \mathbf{u}_{k,j}^*] \leftarrow$ Solution of Scenario MPC problem (4)

Output: $\mathbf{u}_{k,j}^*$

The computed Bayesian weights, in addition to scenario pruning, can also be used to weight the different scenarios in the cost function by setting $\omega_j = W_{k,j}$ in (4). By doing so, we give more weight on the scenarios that explains the observations with a higher likelihood.

IV. Case Study

In this section, we demonstrate the proposed scenario tree update mechanism using a gas lift optimization case study.

A. Process description

We consider an oil and gas production network consisting $n_w = 2$ gas lifted wells as shown in Fig.2. The objective is to maximize the total oil production from the network while maintaining the total gas production within the processing capacity constraints. This is expressed as,

$$\min_{w_{gl,i}} \; -c_o \sum_{i=1}^{n_w} w_{po,i} + c_{gl} \sum_{i=1}^{n_w} w_{gl,i} \tag{9a}$$

s.t.

$$\sum_{i=1}^{n_w} w_{pg,i} \leq w_{pg}^{max} \tag{9b}$$

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}) \tag{9c}$$

$$\mathbf{p} \in \mathcal{U} \tag{9d}$$

$$\forall i \in \{1, \dots, n_w\}$$

where $w_{gl,i}$ is the gas lift injection rate for each well and is the manipulated variable ($n_u = 2$), $w_{po,i}$ and $w_{pg,i}$ are the oil and gas production rates from each well respectively, w_{pg}^{max} is the total gas processing capacity. c_o and c_{gl} are economic terms that represents the value of oil and cost of gas compression respectively. (9c) represents the nonlinear dynamic model of the gas lifted wells. The reader is referred to [5] for more detailed description of the gas lifted well models.

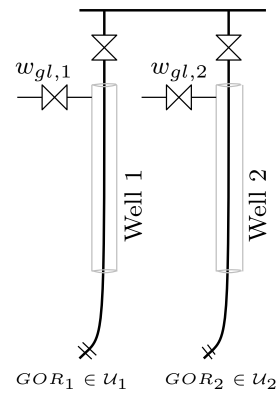


Fig. 2: Schematic representation of two gas lifted wells.

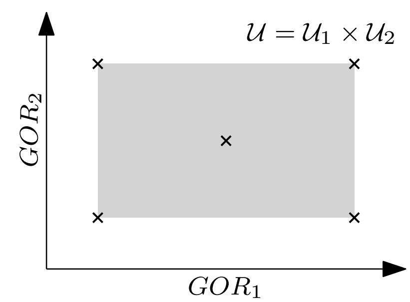


Fig. 3: Uncertainty subspace showing the 5 discrete scenarios used to generate the scenario tree.

The gas-oil-ratio GOR is a reservoir property that denotes the ratio of oil and gas entering each well from the reservoir, which is an uncertain parameter (i.e. $\mathbf{p} = GOR$). $\mathcal{U}$ denotes the uncertainty characteristics to which the uncertain parameter is known to belong. Since we have two wells, we have two uncertain parameters ($n_p = 2$), namely, the gas-oil ratio for each well. In this simulation example, we consider the uncertainty in $GOR_i \in \mathcal{U}_i$ to be equally distributed with $GOR_1 \in [0.05, 0.15]$kg/kg and $GOR_2 \in [0.11, 0.13]$kg/kg. Hence the uncertainty set $\mathcal{U} = \mathcal{U}_1 \times \mathcal{U}_2$ is given as a box uncertainty set. For the multistage scenario MPC, $M = 5$ discrete realizations of the uncertainty are considered that corresponds to the combination of minimum, maximum and nominal values of the gas-oil-ratio of the two wells. This is schematically represented in Fig.3, where the discrete scenarios are marked by an '$\times$'. Multistage scenario MPC was then applied with a robust horizon of $N_r = 1$ leading to $S = 5$ discrete scenarios, as described in [15].

The multistage scenario MPC is setup with a sampling time of 5min and with a prediction horizon of 2hours using `CasADi v3.1.0` [16] with `MATLAB` programming environment. The resulting nonlinear programming (NLP)

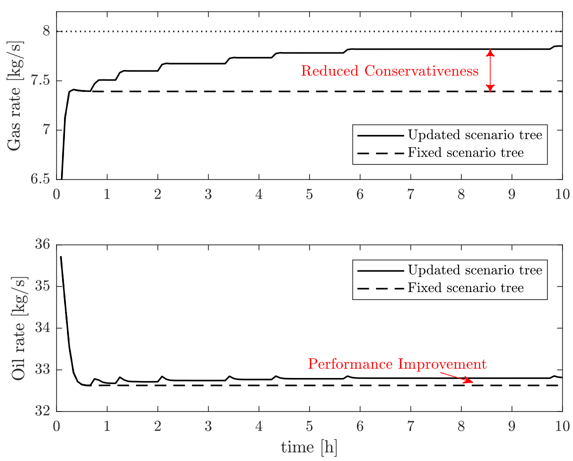


Fig. 4: Simulation results comparing the proposed method with online update of scenario tree with fixed scenario tree multistage MPC

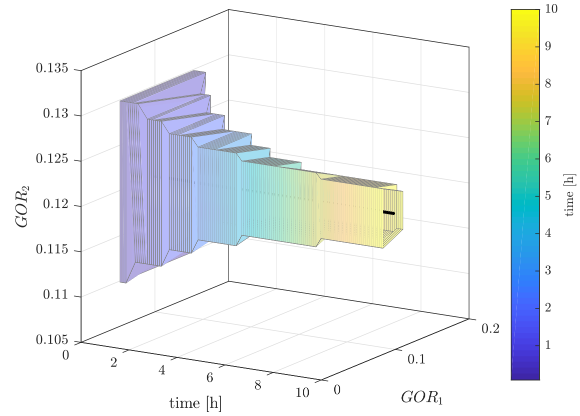
problem is solved using `IPOPT v.3.12.2` running with a `MUMPS` linear solver. The simulations were performed on a 2.6GHz workstation with 16GB memory. The plant model was implemented using the `IDAS` integrator. It is worth noting that, given the timescale of the optimization problem, the changes in the reservoir properties are very slow and can be considered constant for the optimization problem. Hence, the uncertain parameter GOR, which is a reservoir property, can be assumed to be a time invariant parameter for the production optimization problem, with a given initial uncertainty characteristics $\mathcal{U}$.

B. Simulation results

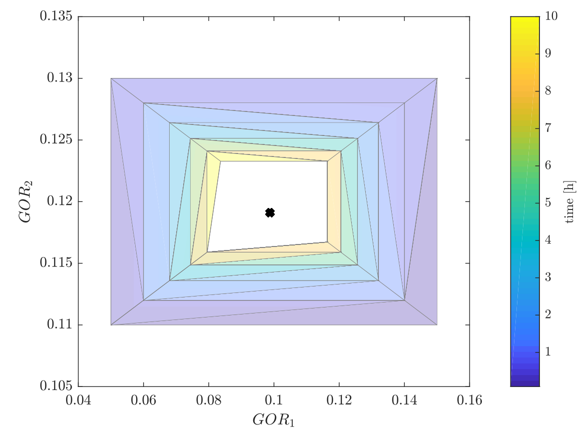
In this section, we apply the proposed multistage MPC formulation on the gas lift optimization case study, where the scenarios are updated online using the recursive Bayesian weighting scheme, with a suitable step length $\alpha = 0.2$. The results from the standard multistage MPC with a fixed scenario tree was used to benchmark the performance of the proposed scheme. To update the scenario tree, pressure measurements and flow measurements through the wellhead choke are used to compute the Bayesian weight for the different scenarios.

In the first simulation, we assume the true realization of the gas-oil-ratio in the plant is $GOR = [0.0975, 0.1201]^\mathsf{T}$ and the maximum gas capacity limit is considered to be $w_{pg}^{max} = 8$kg/s. Fig.4 shows the closed-loop simulation results for the multistage MPC with updated scenario tree (solid lines) compared to the standard multistage MPC with fixed scenario tree (dashed lines). The first subplot shows the total gas production rate and the second plot shows the total oil production rate.

In the proposed multistage MPC scheme, the scenarios with very low weights are updated as described in Algorithm 1. By doing so, the span of the uncertainty subspace covered by the scenarios gradually reduces. Consequently,



(a)



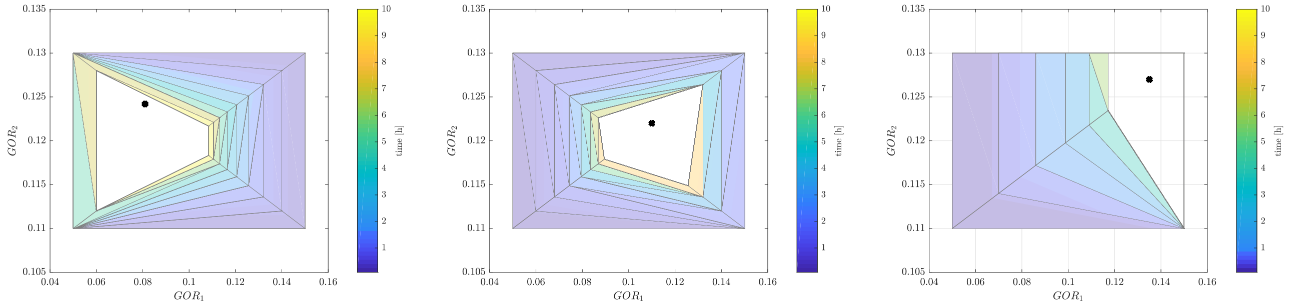
(b)

Fig. 5: Uncertainty space spanned by the different scenarios updated online, initiated with a box comprised of a combination of minimum and maximum GOR values. (a) 3-D view of the uncertainty subspace over time. (b) corresponding 2-D view. The true realization is marked by '$\times$'

the conservativeness also reduces gradually. This can be clearly seen in the plot in Fig.4, where the total gas produced is utilizing more of the available capacity to increase the total oil production compared to the standard multistage MPC with a fixed scenario tree. Fig.5 shows the evolution of the discrete scenarios in the scenario subspace. It can be seen that the span of the uncertainty subspace is gradually reduced compared to the initial box uncertainty.

We then test the proposed approach for 30 different realizations of the uncertainty randomly selected from $\mathcal{U}$ to be the true realization in the plant, as shown in Fig.7 (bottom subplot). To evaluate the performance, we plot the integrated objective, which is the oil production rate integrated over a period of 10 hours for each simulation run and compare it with the multistage MPC with fixed scenario tree, see Fig.7 (top subplot).

It can be clearly seen that by updating the scenario tree and gradually shrinking the uncertainty space covered by the



(a) (b) (c)

Fig. 6: 2-D view of the uncertainty subspace for three different uncertainty realizations for (a) run number 8, (b) run number 18, (c) run number 12. The true realization of GOR is shown as a solid black '$\times$'

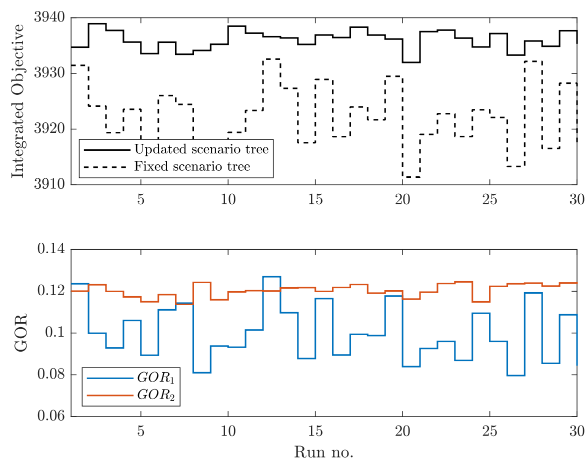


Fig. 7: Monte Carlo Simulations results with different realizations of the uncertain parameters.

scenario tree, the conservativeness can be further reduced, leading to less conservative operation. The uncertainty space covered by the updated scenario tree for run numbers 8, 12 and 18 are shown in Fig.6, to portray how the initial box uncertainty set is updated for different true realizations of the uncertain parameter.

V. Conclusion

When the time-invariant uncertain parameters cannot be estimated directly, methods that update the uncertainty characteristics may be useful in reducing the conservativeness. In this paper, we presented one such adaptive robust multistage MPC framework, where the scenario tree was updated online by computing recursive Bayesian weights for the different scenarios as summarised in Algorithm 1. It is important to note that this paper considers only the classs of systems with time-invariant parameters, where the idea is to narrow down the uncertainty using available measurements.

This is a simple, yet practical approach to updating the scenario trees based on the observed closed-loop performance. Using the gas lift optimization case study, we showed that by updating the scenario tree online, the conservativeness can be further reduced, leading to increased profits.

References

[1] P. J. Campo and M. Morari, "Robust model predictive control," in *American Control Conference, 1987*. IEEE, 1987, pp. 1021–1026.

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

[3] 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.

[4] 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.

[5] D. Krishnamoorthy, B. Foss, and S. Skogestad, "Real time optimization under uncertainty - applied to gas lifted wells," *Processes*, vol. 4, no. 4, 2016.

[6] A. Verheyleweghen and J. Jäschke, "Framework for combined diagnostics, prognostics and optimal operation of a subsea gas compression system," *IFAC-PapersOnLine (IFAC World Congress)*, vol. 50, no. 1, pp. 15 916–15 921, 2017.

[7] 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.

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

[9] S. Thangavel, S. Lucia, R. Paulen, and S. Engell, "Dual robust nonlinear model predictive control: A multi-stage approach," *Journal of Process Control*, vol. 72, pp. 39–51, 2018.

[10] S. Lucia and R. Paulen, "Robust nonlinear model predictive control with reduction of uncertainty via robust optimal experiment design," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 1904–1909, 2014.

[11] B. Aufderheide and B. W. Bequette, "Extension of dynamic matrix control to multiple models," *Computers & chemical engineering*, vol. 27, no. 8-9, pp. 1079–1096, 2003.

[12] N. N. Nandola and S. Bhartiya, "A multiple model approach for predictive control of nonlinear hybrid systems," *Journal of process control*, vol. 18, no. 2, pp. 131–148, 2008.

[13] D. Krishnamoorthy, B. Foss, and S. Skogestad, "A distributed algorithm for scenario-based model predictive control using primal decomposition." *IFAC-PapersOnLine (IFAC ADCHEM)*, vol. 51, no. 18, pp. 351–356, 2018.

[14] ——, "A primal decomposition algorithm for distributed multistage scenario model predictive control." *Journal of Process Control*, vol. I(n-Press), 2019.

[15] ——, "Gas lift optimization under uncertainty," *Computer Aided Chemical Engineering*, vol. 40, pp. 1753–1758, 2017.

[16] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, In Press, 2018.