

Application of Data-Driven Economic NMPC on a Gas Lifted Well Network ^{*}

Joakim Rostrup Andersen ^{*} Lars Imsland ^{*}

^{} Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway (e-mail: {joakim.r.andersen,lars.imsland}@ntnu.no).*

Abstract: The Daily Production Optimization (DPO) problem is the task of maximizing production of hydrocarbons subject to operational constraints. Handling of uncertainty in model structure and parameters is of high importance to the usefulness of the solution. Ignoring these challenges will, most likely, render the solution either infeasible or the solution will not be an optimum of the plant. We suggest to apply a data-driven methodology to use state- and output-measurements from the plant to iteratively update the Optimal Control Problem (OCP) which are used to control the plant. The goal of the method is to tune the OCP such that the solution will go towards an optimum of the plant as the parameters are being updated. A Reinforcement Learning updating technique is used to update the optimization formulation.

Keywords: nonlinear process control, reinforcement learning control, model predictive and optimization-based control

1. INTRODUCTION

The oil and gas industry produce value by drilling wells to extract hydrocarbons which are trapped in subsurface reservoirs. The Daily Production Optimization (DPO) activity within this industry aims at optimizing production from these wells to maximize revenue, while obeying a set of production constraints. There are many different sources giving rise to constraints, the most typical being limitations in how much gas and water, typically extracted along with oil, that can be processed. If the downhole pressure in the wells are not high enough to lift the fluids to the topside at an acceptable flowrate, then artificial lift can be used. One commonly used artificial lift method is the injection of gas into the wellbore to lower the density of the fluid mixture. Since a production facility typically may have many wells, and a limited amount of available gas, this also gives rise to a constrained optimization problem, namely the allocation of available lifting gas. The controllable elements in production optimization are typically chokes (valves) to control flows, and the settings of compressors and pumps.

Mathematical programming is well suited to tackle the DPO problem (Foss et al., 2018). Kosmidis et al. (2004) formulated the DPO problem as a mixed integer nonlinear programming problem, where the integers were used to piece wise linearize nonlinear functions. Kosmidis et al. (2005) used integers to also decide the routing of the streams. Other works looking at the DPO as a mixed integer (non)linear programming problem are Misener et al. (2009); Codas and Camponogara (2012); Silva and Camponogara (2014); Grimstad et al. (2016).

Modelling of the multiphase reservoir inflow, wells, risers and topside facilities are a complex task, with limited information available of subsurface conditions. Thus, the models used in DPO are typically uncertain, which may lead to poor prediction capabilities. This uncertainty is a serious challenge for the DPO problem (Foss et al., 2018). Nonetheless, most works in production optimization simply ignores this issue altogether (Krishnamoorthy et al., 2016). The quality and usefulness of the solution are heavily affected by such negligence. The solution could be infeasible for the real system, or it could be suboptimal.

Uncertainty in the DPO problem was first explicitly handled by Bieker et al. (2007) according to Foss et al. (2018).

Dynamic scenario-based optimization was used to deal with parametric uncertainty by Krishnamoorthy et al. (2016) for production optimization of gas lifted wells. They reduced the conservativeness of the solution compared to the classical worst-case optimization, while still being robustly feasible.

To avoid the need of steady-state measurements, Krishnamoorthy et al. (2018) suggested to use transient measurements to perform model updating, whilst only using a steady-state model in the optimization. They demonstrated the hybrid method on a gas lifted well network simulation case study.

The modifier adaptation optimization method was applied to a similar setup by Matias et al. (2018), also considering parametric uncertainty. At convergence, the plant and model optimum coincided. Modifier adaptation has the advantage that both model structure and model parameter mismatch can be handled. However, the applied method needs a steady-state detection phase.

^{*} This research is a part of BRU21 – NTNU Research and Innovation Program on Digital and Automation Solutions for the Oil and Gas Industry (www.ntnu.edu/bru21).

The contribution of this paper is the introduction of a methodology into the field of DPO. The method is able to compensate for both error in the model structure and uncertainty in parameters. Steady-state detection is not applicable as dynamic models are used. Similar to the modifier adaptation method, the goal is to use the data to optimally control the plant, and not necessarily to correctly model the system dynamics. The data-driven methodology was proposed by Gros and Zanon (2020). The method combines Economic Nonlinear Model Predictive Control (ENMPC) with Reinforcement Learning (RL). The ENMPC is used to control the system, whereas the RL is used to tune the ENMPC such that it more optimally controls the system.

2. THEORY

In this Section, the relevant background theory will be provided. It gives an introduction to the relevant parts of Reinforcement Learning (RL). The newly suggested function approximator (Gros and Zanon, 2020) is put into perspective of the other parts, and an updating scheme for the function is provided. For a deeper understanding of RL, we refer to Sutton and Barto (2018).

2.1 Reinforcement Learning

Reinforcement Learning is a kind of machine learning, where an *agent* is supposed to learn how to operate a system to maximize a numerical reward signal. The agent is not explicitly told what the reward will be for taking an action, but it must instead try the action and learn from it. In Reinforcement Learning, there are three main components. The first component is the *reward* which is used as a metric to evaluate the goodness of the immediate return of the applied action in the current state. The second component is the *value function*, which gives the value of being in a state. It is the discounted sum of all future rewards given that we follow the *policy*. Finally, we have the *policy* which provides the next action to apply given a state. There is also a component known as the *action-value function*, which is the same as the value function except that it provides the value of a state given that we apply a specified action at the first step.

Given a state s , and an action a , we have the policy $\pi_\theta(s)$, value function $V_\theta(s)$ and action-value function $Q_\theta(s, a)$, parameterized by the parameter vector θ . The relationship between them is given as follows:

$$V_\theta(s) = \min_a Q_\theta(s, a), \quad (1a)$$

$$\pi_\theta(s) = \arg \min_a Q_\theta(s, a). \quad (1b)$$

Let s' denote the next state, and $r(s, a)$ denote the reward, then (1a) can be rewritten as the Bellman Equation:

$$V_\theta(s) = \min_a r(s, a) + \gamma V_\theta(s'), \quad (2)$$

where γ is a discount factor.

Finding the optimal action-value function for all possible combinations of states and actions is in general not possible. However, a process is typically working in only a small subset of such combinations. The aim is to create an approximation of the (action-) value function in the operating region.

2.2 Action-value approximation

In this paper, we will use a parametric Economic Nonlinear Model Predictive Control (ENMPC) as the function approximator for the action-value function. This approximator was proposed by Gros and Zanon (2020), and extended with system identification capabilities by Martinsen et al. (2020). A parametric ENMPC is presented below.

$$Q_\theta(s, a) = \min_{x, u, \sigma} \sum_{k=0}^{N-1} [\gamma^k (L_\theta(x_k, u_k) + \omega^\top \sigma_k)] + \gamma^N L_\theta^f(x_N) + \lambda_0(x_0) \quad (3a)$$

$$s.t. \quad x_{k+1} = f_\theta(x_k, u_k) \text{ for } k \in [0, \dots, N-1] \quad (3b)$$

$$h_\theta(x_k, u_k) \leq \sigma_k \text{ for } k \in [0, \dots, N-1] \quad (3c)$$

$$x_0 = s \quad (3d)$$

where $L_\theta(\cdot)$ is an estimated reward function, ω is used to penalize constraint violation given by the slack variable σ , $f_\theta(\cdot)$ is the dynamic model, N is the control horizon, and x_0 is the initial state. Finally, $L_\theta^f(\cdot)$ is the terminal cost that should capture all future rewards, and $\lambda_0(x_0)$ is an initial cost which does not impact the solution of the optimization problem, but may help the RL. For the rest of the article, we will use s, a to denote real states and actions, whereas x, u will denote predicted states and inputs.

This choice of function approximator is actually rather intuitive. An MPC gives you the next action to apply given current state, just like the policy. It also provides the objective function value given current state, just like the value-function. If we impose an additional constraint: $u_0 = a$, then we get the same behaviour as an action-value function.

2.3 Goal of Reinforcement Learning

The overall goal in Reinforcement Learning (RL) is to find a policy $\pi_\theta(\cdot)$ that minimizes the following expected value:

$$\mathbb{E} \sum_{i=0}^{\infty} \gamma^i \bar{L}(s_i, \pi_\theta(s_i)), \quad (4)$$

where \bar{L} is given as:

$$\bar{L}(s_i, a_i) = L(s_i, a_i) + \hat{\omega}^\top \max(0, h(s_i, a_i)), \quad (5)$$

where the first term is the experienced reward, and the latter is the penalization of any unwanted behaviour due to that experience. For example, in our case study $L(s_i, a_i)$ is the negative value of the produced fluids, whereas the latter term represents any gas that must be flared due to excessive gas production. The weight $\hat{\omega}$ is used to tell the RL how critical it is to not violate the constraints $h(\cdot)$. It should be noted that this does not allow for hard constraints to be included since this would indicate a value of $+\infty$ in $\bar{L}(\cdot)$, which most RL methods cannot handle (Martinsen et al., 2020). Moreover, having hard constraints in any MPC scheme may lead to an infeasible optimization problem. One could also argue that if one has state/output constraints that never can be violated, then either one must control the system extremely conservatively, or one would need a perfect model of the system and its constraints. And if one has the perfect model, then RL is superfluous either way.

2.4 Q-learning

To minimize (4) we will apply the concept of Q-learning (Watkins, 1989). The tuning of the action-value function, $Q_\theta(s, a)$, is driven by the minimization of what is known as the temporal-difference error δ_t :

$$\delta_t = y_t - Q_\theta(s_t, a_t), \quad (6)$$

where y_t is the *fixed target value* given by $y_t = \bar{L}(s_t, a_t) + \gamma V_\theta(s_{t+1})$. In the parameter updating step, it is assumed that y_t is independent of θ . This assumption makes the updating a bootstrapping strategy which means caution should be used when selecting the step size.

A batch of samples will be used each time the parameters are updated to help avoid overfitting. A Gauss-Newton method will be used to minimize the sum of squares:

$$\min_{\theta} \sum_{t=1}^{n_b} \delta_t^2, \quad (7)$$

where n_b is the number of elements in the batch. The update law with learning rate, or step size, α , is given by:

$$\theta \leftarrow \theta + \alpha (J_Q^T J_Q)^{-1} J_Q^T \delta, \quad (8)$$

where

$$J_Q = \begin{bmatrix} \nabla_{\theta} Q_{\theta}(s_{t,1}, a_{t,1}) \\ \nabla_{\theta} Q_{\theta}(s_{t,2}, a_{t,2}) \\ \vdots \\ \nabla_{\theta} Q_{\theta}(s_{t,n_b}, a_{t,n_b}) \end{bmatrix}, \quad \delta = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_{n_b} \end{bmatrix}. \quad (9)$$

Inverting the approximate Hessian $H = J_Q^T J_Q$ will lead to problems if the matrix is singular, or badly conditioned. To circumvent this issue, let \tilde{H}^\dagger denote the ‘‘pseudo-inverse’’ of the symmetric H :

$$\tilde{H}^\dagger = \mathcal{F}(\mathcal{F}^T H \mathcal{F})^{-1} \mathcal{F}^T, \quad (10)$$

where \mathcal{F} is chosen as the ‘‘near’’ fullspace of H , with the modification that the singular value of a direction has to be strictly greater than σ_{\min} to be included in the fullspace. The new update-law then becomes:

$$\theta \leftarrow \theta + \alpha \tilde{H}^\dagger J_Q^T \delta. \quad (11)$$

This updating law does not guarantee that a global optimum of the parameters for the nonlinear Q_θ function is found. However, most applications of nonlinear function approximators in RL will suffer from this limitation (Martinsen et al., 2020). Nonetheless, as long as the Q_θ function is fitted to the operating range of interest, a local optimum is sufficient. If the operating area changes, then we could apply online updating to try to fit the Q_θ function to the current operating range.

3. SIMULATION STUDY

In this study, we will apply the above theory on a hydrocarbon production system consisting of two gas lifted wells, see Fig. 1, which produce to a common manifold with a fixed pressure of 50 bar. The goal is to distribute the available lift gas between the two wells such that the oil production is maximized. The processing facility also has a limitation on how much gas it can handle.

3.1 The model of a gas lifted well

The utilized gas lifted well model is explained in great details in Binder (2012). All constants may be found in that reference, we used well 2 and 4.

This model is based on mass balance of the different phases in the annulus and the tubing. This kind of model has been developed and studied in several papers, e.g., see Jahanshahi (2013), and Imsland (2002). The version used in this paper takes into account oil, water and gas phases, but flow friction is neglected.

The states of the model are the mass of gas in the annulus m_{ga} , mass of gas in the tubing m_{gt} , and mass of liquid in the tubing m_{lt} . The ordinary differential equations, which are based upon mass balance, for well j is given below:

$$\dot{m}_{ga,j} = w_{gl,j}(u_{gl,j}) - w_{gi,j}(\cdot), \quad (12a)$$

$$\dot{m}_{gt,j} = w_{gr,j}(\cdot) + w_{gi,j}(\cdot) - w_{gp,j}(u_{pc,j}), \quad (12b)$$

$$\dot{m}_{lt,j} = w_{lr,j}(\cdot) - w_{lp,j}(u_{pc,j}), \quad (12c)$$

where $w_{gl,j}$ is the injected gas into the annulus, $w_{gi,j}$ is the gas going from the annulus into the tubing, $w_{gr,j}$ and $w_{lr,j}$ are the gas and liquid coming from the reservoir into the tubing, finally, $w_{gp,j}$ and $w_{lp,j}$ is the gas and liquid produced by the well. For each well, there are two manipulated variables: $u_{gl,j}$ and $u_{pc,j}$. The first is $u_{gl,j} = w_{gl,j}$ which is the amount of gas in kg/s injected into the annulus, and the second is $u_{pc,j} \in [0, 1]$ which is the production choke opening (cf. Fig. 1). In (12) we have indicated which flows directly depend upon the manipulated variables.

The model contains two one-way chokes to make sure that fluids from the tubing may not enter the annulus and to make sure that the well does not drain fluids from the manifold. E.g., the flow through the gas injection valve is modelled as:

$$w_{gi,j} = C_{iv,j} \sqrt{\rho_{gi,j} \max(0, p_{ai,j} - p_{ti,j})}, \quad (13)$$

where $C_{iv,j}$ is the valve specific constant for the injection valve, $\rho_{gi,j}$ is the density of the gas in the annulus at the injection valve, and $p_{ai,j}$ and $p_{ti,j}$ are the pressures at each side of the valve. The Optimal Control Problem (OCP) is implemented using CasAdi (Andersson et al., 2019), which allows to use max-functions in the formulation. To avoid the issue with an unbounded gradient of the square root in (13) when the argument is close to zero, we used a smooth approximation when $p_{ai,j} - p_{ti,j}$ becomes small.

If one would like to apply a constraint on, or penalize, how fast the control inputs may change, then one could augment the state space with two more states for each well:

$$\dot{u}_{gl,j} = v_{gl,j}, \quad (14a)$$

$$\dot{u}_{pc,j} = v_{pc,j}, \quad (14b)$$

where the v 's represent the rate of change of the controls and will be the new manipulated variables in the Model Predictive Control (MPC). The state vector x contains five elements:

$$[m_{ga,j}, m_{gt,j}, m_{lt,j}, u_{gl,j}, u_{pc,j}],$$

and the control vector has two elements, $[v_{gl,j}, v_{pc,j}]$, for each well at each time step.

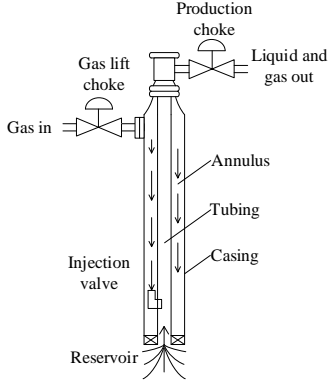


Fig. 1. Schematic of the gas lifted well model.

3.2 The RL function approximator

Based on the theory outlined in the previous section, we use the function approximator $Q_\theta(a, s)$ implicitly defined by the solution of:

$$\min_{x, v, \sigma} \lambda_0 + \sum_{k=0}^{N-1} [\gamma^k (w_{\theta, op}(x_k, v_k) + \omega \sigma_{k+1})] + \gamma^N \frac{\gamma}{1 - \gamma} w_{\theta, op}(x_{N-1}, v_{N-1}) \quad (15a)$$

$$s.t. \quad x_{k+1} = f_\theta(x_k, v_k), \text{ for } k \in [0, \dots, N-1] \quad (15b)$$

$$w_{\theta, gp}(x_k) + \lambda_1 \leq \Gamma + \sigma_k, \text{ for } k \in [1, \dots, N] \quad (15c)$$

$$\underline{x} \leq x_k \leq \bar{x}, \text{ for } k \in [1, \dots, N] \quad (15d)$$

$$\underline{v} \leq v_k \leq \bar{v}, \text{ for } k \in [1, \dots, N] \quad (15e)$$

$$x_N \text{ is a steady-state} \quad (15f)$$

$$x_0 = s \quad (15g)$$

where $w_{\theta, op}(x_k, v_k)$ is the integrated oil production from both wells from step k to $k+1$, $w_{\theta, gp}(x_k)$ is the instantaneous gas production at step k , and $f_\theta(x_k, v_k)$ is defined by (12). The gas handling constraint is enforced by (15c), where $\Gamma = 3.8\text{kg/s}$ is the maximum gas that should be produced at a time instant, and λ_0 and λ_1 are two tunable bias parameters. The control horizon was set to $N = 100$ steps, over a simulation time $T = 3600\text{s}$ (1 hour), giving a time step of 36s . The penalty factor was set $\omega = 1000$, and the discount factor was set to $\gamma = 0.98$.

We impose (15f) to make sure the system reaches a steady-state. We do this in the example by requiring $x_N = x_{N-j}$, for $j \in [1, \dots, 5]$. The steady-state allows us to use the property of a geometric series to efficiently express the discounted terminal cost:

$$\sum_{k=N}^{\infty} \gamma^k = \gamma^N \frac{\gamma}{1 - \gamma}, \quad \text{for } |\gamma| < 1, \quad (16)$$

multiplied by the oil produced at the last interval. Note that constraint (15f) could lead to infeasibility issues, if that was encountered the simulation time T and control horizon N could be extended.

The lower bounds, \underline{x} , for the masses and production chokes were set to zero, and for the gas injection it was set to 1.2 and 0.1 kg/s for well 2 and 4, respectively. The upper bounds, \bar{x} for all states were set to ∞ except for the production chokes that had an upper bound of 1 . The upper and lower bounds for all the v 's were set to $\pm 0.25/(T/N)$ to limit aggressive behaviour.

For this case study, we chose four tunable parameters in the function approximator (15). Two of them are directly related to the optimization formulation: λ_0 and λ_1 . For each well, the valve specific constant for the injection valve, see Fig. 1, is selected as a parameter that the RL can tune. This means that the parameter vector θ has the elements $C_{iv,2}$, $C_{iv,4}$, λ_0 , and λ_1 . This selection of parameters is not unique and other choices could give better, or worse, performance.

3.3 Implementation details

The optimal control problem (OCP) was implemented using CasADi (Andersson et al., 2019) version 3.5.0 with IPOPT version 3.12.3 (Wächter and Biegler, 2006) as the nonlinear program (NLP) solver. IPOPT was ran with default parameters except that maximum iterations was increased to 5000. Direct Collocation, with Legendre collocation points and a polynomial order of degree of 2, was used as a direct transcription method, see e.g., Biegler (2010) for more information on the method.

For testing, a plant replacement model was implemented using the same model, but with somewhat different parameters, as can be seen in Table 1. The plant was integrated with the explicit Runge-Kutta 4 method. Two integration steps were used for each control step. The penalty factor in (5) was set to $\hat{\omega} = 20000$, the h was taken as the instantaneous excessive gas production at $t+1$, and the L was set to the accumulated oil production from t to $t+1$.

For the RL, we used a moving window of 50 elements. The batch was chosen to always include the newest point, in addition to 7 uniformly randomly selected from the window. The step size, or learning rate, was set to $\alpha = 0.01$. The threshold for counting a singular value as zero was set to $\sigma_{\min} = 0.1$.

Scaling was applied to the state vector, the control vector, the objective function and the constraints in the NLP to obtain the same order of magnitudes. When performing the update law (11), we only performed scaling on the tunable parameters such that the gradients of the Lagrangian of the NLP with respect to the parameters were approximately of the same magnitude.

Sometimes when performing the Gauss-Newton step, we experienced that the evaluation of the Q_θ function was unsuccessful. When this happened, we ignored that sample in the batch update and continued. There were 59, 7 and 1 batches where 1, 2 and 3 evaluations failed, respectively.

Table 1. Real and guessed values of the parameters.

Parameter	Real value	Guessed value
r_{gor} well 1	0.07	0.06
r_{wc} well 1	0.7	0.8
r_{gor} well 2	0.09	0.1
r_{wc} well 2	0.5	0.6
C both wells	0.8	0.7
C_{pc} both wells	0.0016	0.00176
C_{iv} both wells	0.00016	0.000144

4. RESULTS AND DISCUSSION

In this Section, we present the results obtained in the simulation study. The method is compared to the ‘‘perfect’’ case

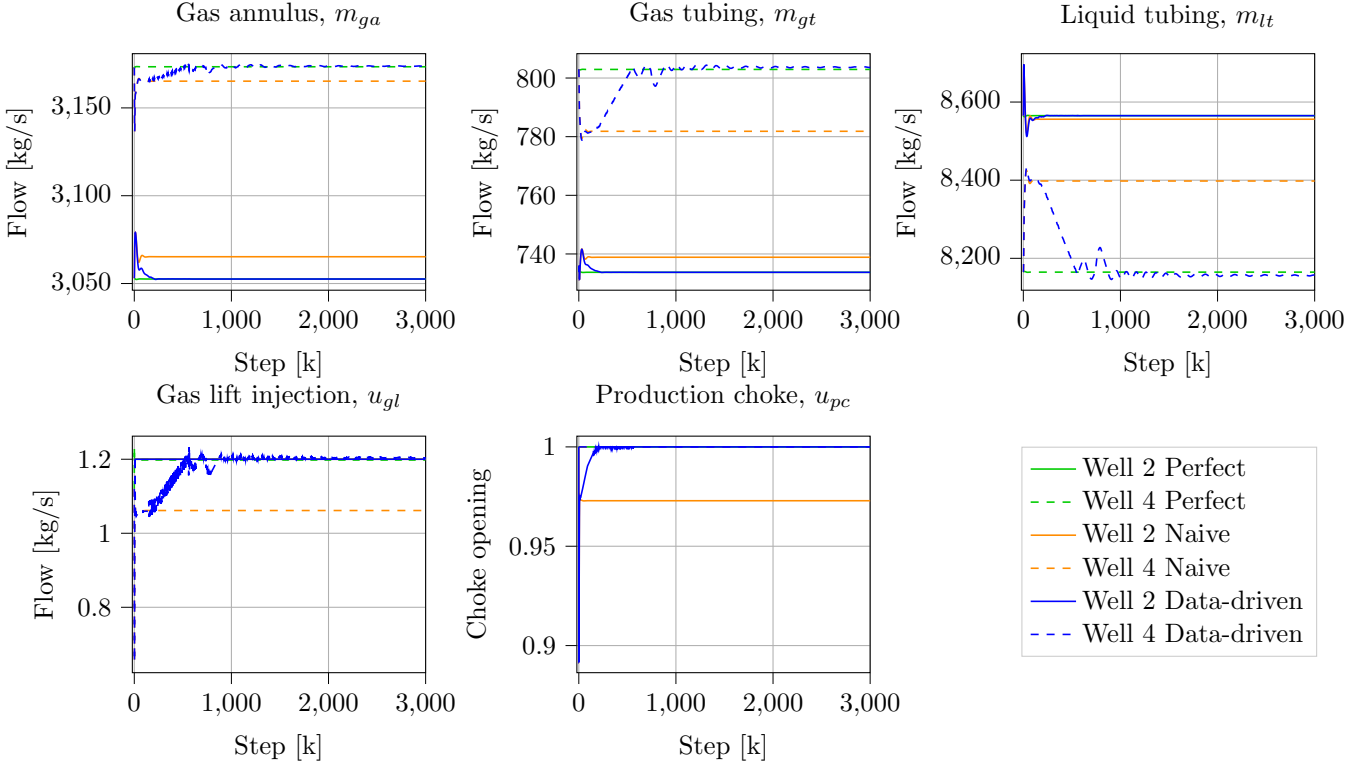


Fig. 2. The states, including the controls, for the different strategies.

where we have full knowledge of the model parameters, and also to a “naive” approach where an ENMPC was applied with wrong parameters, see Table 1, and nothing was done to improve the control model.

In Fig. 2, we see that as the RL keeps updating the parameter vector θ , the MPC keeps suggesting another input. The data-driven ENMPC slowly goes towards the trajectories obtained by the perfect ENMPC.

In this specific application, two fully open production chokes and an active gas processing constraint are necessary for optimal operation. In Fig. 2, we can see that the chokes are fully open for both the ideal and data-driven approaches, but not for the naive one. In Fig. 3, we can see that the gas processing constraint is reached at the beginning by the ideal approach, and it is never reached by the naive approach. Further, we see that the constraint will eventually be reached also by the data-driven ENMPC. However, it does take some time for the RL to tune the parameters to some values that give a close to optimal control. It takes approximately 550 iterations before the constraint is active. The slow convergence is due to a small learning rate, or step size.

If we increased the step size, then the constraint might be violated by a greater amount, leading to a large penalty-term. With a too small constraint violation penalty, $\hat{\omega}$, then the calculated optimal control input could be one that keeps violating the constraints. On the other hand, if it is too big, combined with a large step size, it could lead to an updating step where the parameters are moved far away from the optimal values.

In Fig. 3, we can see that data-driven ENMPC does slightly violate the constraint. However, the amount is in

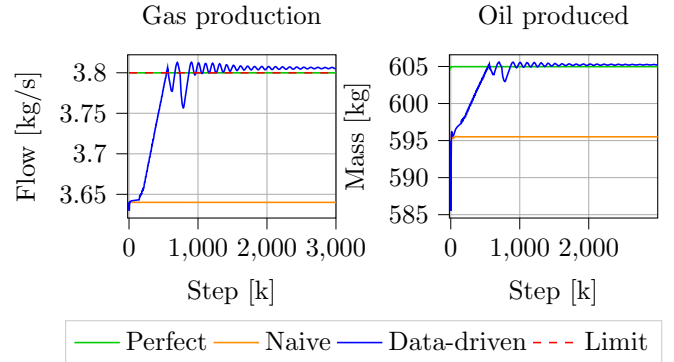


Fig. 3. The gas production and produced oil for the different strategies.

the range of 6 grams of gas a second which most likely is unnoticeable due to non-perfect sensors. Further, it is reasonable to assume that operators would use a Γ that is lower than the actual limitation of the plant. Increasing the $\hat{\omega}$ slightly and decreasing the step size would, most likely, decrease the gap between the produced gas and the limit Γ .

The integrated produced oil at each time interval is shown in Fig. 3. The figure shows that the data-driven ENMPC converges towards the same production as is achieved by the ENMPC with full knowledge of the system.

The evolution of the four tunable parameters can be seen in Fig. 4. The final values of the two valve specific constants are not the same as those in the “Real” column of Tab. 1. This is as expected, because the goal is to tune the Q_θ function such that we optimally control the system,

and not to do system identification. If we set the two bias parameters to zero, and let the RL tune all the parameters in Tab. 1, then one may expect the parameters to converge to the real values. This could potentially also remove the small constraint violation in the gas processing capabilities.

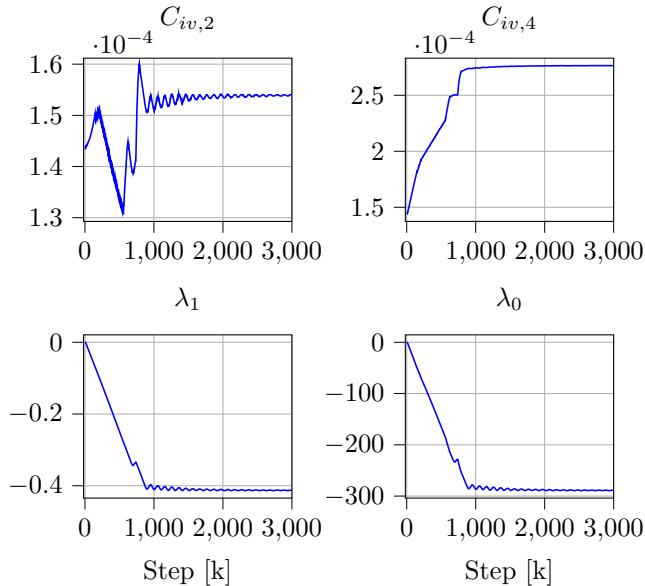


Fig. 4. The evolution of the tunable parameters.

5. CONCLUSION

We have studied a data-driven ENMPC method proposed by Gros and Zanon (2020), and applied it to a simulation study of two gas lifted oil wells. The results were promising in the sense that the data-driven ENMPC was able to tune the parameters to a set of values that gave a near-optimal operation of the process even though there were more incorrectly guessed parameters than tunable ones. It should be noted that the approach is a state-feedback approach, and relies on measurements of states that are not commonly available. In practice, this method must therefore be accompanied by a state estimation scheme.

ACKNOWLEDGEMENTS

We would like to thank Andreas Bell Martinsen for sharing his valuable insight and knowledge on the methodology.

REFERENCES

- Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1), 1–36.
- Biegler, L.T. (2010). *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. SIAM.
- Bieker, H.P., Slupphaug, O., Johansen, T.A., et al. (2007). Well management under uncertain gas or water oil ratios. In *Digital Energy Conference and Exhibition*. Society of Petroleum Engineers.
- Binder, B.J.T. (2012). *Production optimization in a cluster of gas-lift wells*. Master’s thesis, Institutt for teknisk kybernetikk, NTNU.
- Codas, A. and Camponogara, E. (2012). Mixed-integer linear optimization for optimal lift-gas allocation with well-separator routing. *European Journal of Operational Research*, 217(1), 222 – 231.
- Foss, B., Knudsen, B.R., and Grimstad, B. (2018). Petroleum production optimization – a static or dynamic problem? *Computers & Chemical Engineering*, 114, 245 – 253. FOCAPO/CPC 2017.
- Grimstad, B., Foss, B., Heddle, R., and Woodman, M. (2016). Global optimization of multiphase flow networks using spline surrogate models. *Computers & Chemical Engineering*, 84, 237 – 254.
- Gros, S. and Zanon, M. (2020). Data-driven economic nmmpc using reinforcement learning. *IEEE Transactions on Automatic Control*, 65(2), 636–648.
- Imsland, L. (2002). *Topics in nonlinear control.: Output Feedback Stabilization and Control of Positive Systems*. Phd thesis, Norwegian University of Science and Technology (NTNU), Trondheim, Norway. 112, 0809-103X.
- Jahanshahi, E. (2013). *Control Solutions for Multiphase Flow: Linear and nonlinear approaches to anti-slug control*. Phd thesis, Norwegian University of Science and Technology (NTNU), Trondheim, Norway. 271, 1503-8181.
- Kosmidis, V.D., Perkins, J.D., and Pistikopoulos, E.N. (2004). Optimization of well oil rate allocations in petroleum fields. *Industrial & Engineering Chemistry Research*, 43(14), 3513–3527.
- Kosmidis, V.D., Perkins, J.D., and Pistikopoulos, E.N. (2005). A mixed integer optimization formulation for the well scheduling problem on petroleum fields. *Computers & Chemical Engineering*, 29(7), 1523 – 1541.
- Krishnamoorthy, D., Foss, B., and Skogestad, S. (2016). Real-time optimization under uncertainty applied to a gas lifted well network. *Processes*, 4(4), 52.
- Krishnamoorthy, D., Foss, B., and Skogestad, S. (2018). Steady-state real-time optimization using transient measurements. *Computers & Chemical Engineering*, 115, 34 – 45.
- Martinsen, A.B., Lekkas, A.M., and Gros, S. (2020). Combining system identification with reinforcement learning-based mpc. *arXiv preprint arXiv:2004.03265*.
- Matias, J.O., Le Roux, G.A., and Jäschke, J. (2018). Modifier adaptation for real-time optimization of a gas lifted well network. *IFAC-PapersOnLine*, 51(8), 31–36.
- Misener, R., Gounaris, C.E., and Floudas, C.A. (2009). Global optimization of gas lifting operations: A comparative study of piecewise linear formulations. *Industrial & Engineering Chemistry Research*, 48(13), 6098–6104.
- Silva, T.L. and Camponogara, E. (2014). A computational analysis of multidimensional piecewise-linear models with applications to oil production optimization. *European Journal of Operational Research*, 232(3), 630–642.
- Sutton, R.S. and Barto, A.G. (2018). *Reinforcement learning: An introduction*. MIT press.
- Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1), 25–57.
- Watkins, C.J.C.H. (1989). *Learning from delayed Rewards*. Phd thesis, King’s College.