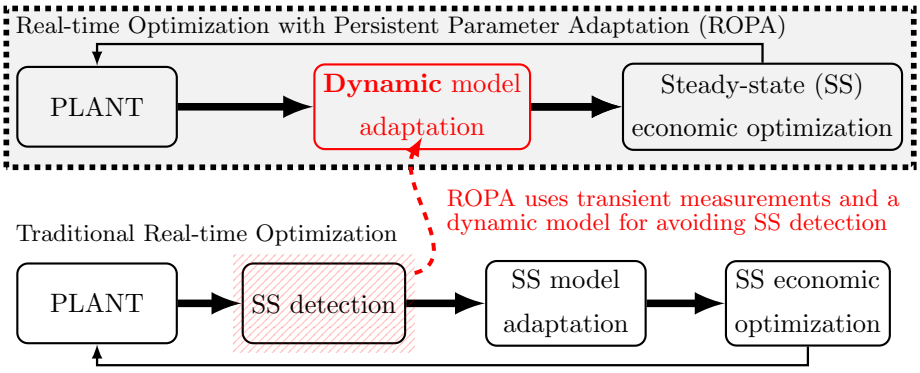
Graphical Abstract

Steady-state Real-time Optimization Using Transient Measurements on an Experimental Rig

José Matias, Julio P. C. Oliveira, Galo A.C. Le Roux, Johannes Jäschke



Highlights

Steady-state Real-time Optimization Using Transient Measurements on an Experimental Rig

José Matias, Julio P. C. Oliveira, Galo A.C. Le Roux, Johannes Jäschke

- RTO with persistent parameter adaptation (ROPA- also known as hybrid RTO), traditional, and dynamic RTO (DRTO) are implemented on lab scale plant;
- ROPA avoided SS detection and showed economic performance comparable to DRTO.
- Guidelines for the practical implementation of ROPA
- An analysis of ROPA convergence to the steady-state optimum is carried out

Steady-state Real-time Optimization Using Transient Measurements on an Experimental Rig*

José Matias[a], Julio P. C. Oliveira[a,b], Galo A.C. Le Roux[b] and Johannes Jäschke[a,*]

[a]Chemical Engineering Department, Norwegian University of Science and Technology, Sem Sælandsvei 4, Kjemiblokk 5, Trondheim, Norway
[b]Chemical Engineering Department, University of São Paulo, Av. Prof. Lineu Prestes, 580, Conjunto das Químicas/bloco 18, São Paulo, Brazil

ARTICLE INFO

Keywords:
Real-time optimization
Online production optimization
Practical Implementation
Oil & gas

Abstract

Real-time optimization with persistent parameter adaptation (ROPA) is an RTO approach, where the steady-state model parameters are updated dynamically using transient measurements. Consequently, we avoid waiting for a steady-state before triggering the optimization cycle, and the steady-state economic optimization can be scheduled at any desired rate. The steady-state wait has been recognized as a fundamental limitation of the traditional RTO approach. In this paper, we implement ROPA on an experimental rig that emulates a subsea oil well network. For comparison, we also implement traditional and dynamic RTO. The experimental results confirm the *in-silico* findings that ROPA's performance is similar to dynamic RTO's performance with a much lower computational cost. Additionally, we present some guidelines for ROPA's practical implementation and a theoretical analysis of ROPA's convergence properties.

1. Introduction

Real-time Optimization (RTO) is a production optimization technique that aims at improving plant economic performance in real-time. In the traditional steady-state RTO (SSRTO), which was originally proposed by Chen and Joseph (1987), a rigorous steady-state model is adapted to the current plant state. Next, the updated model is used for computing the optimal operating point, which is then implemented in the plant (see Figure 1a). Cutler and Perry (1983) found that combining SSRTO with advanced process control can increase plant profit up to 10 %. Despite its potential benefits, SSRTO is still not widely used in practice (Darby et al., 2011).

Multiple challenges and technical issues are associated with this reluctant acceptance (e.g. corrupted information coming from sensors, plant-model mismatch, interface between RTO and advanced control, etc.). Among them, the need to wait for a steady-state (SS) before triggering the optimization cycle has been recognized as a fundamental limitation (Friedman, 1995). This drawback comes from the fact a *steady-state* model of the plant is used for finding the optimal operating strategy. Since it uses a static model, the system *must* be at steady-state for a reliable update of the model parameters. Otherwise, the computed operating conditions are likely to be sub-optimal and potentially hazardous to the plant (Engell, 2007).

Figure 2 illustrates this limitation, showing the schematic response of the traditional RTO to a ramp disturbance. At t_0, the disturbance enters the system. The economic optimization executes only at $t_0 + \Delta t_{SSD}$. This delay is caused by a combination of the time required for the SSD procedure to detect steady-state and the process settling time, which can be considerably long for persistent disturbance as the one shown in Figure 2. As an additional practical limitation, identifying if the data comes from a stationary or transient period is challenging (Menezes, 2016). Practitioners tend to be conservative and only few data periods are identified as steady-state (Câmara et al., 2016). As a result, RTO is too seldom executed, decreasing its economic benefits.

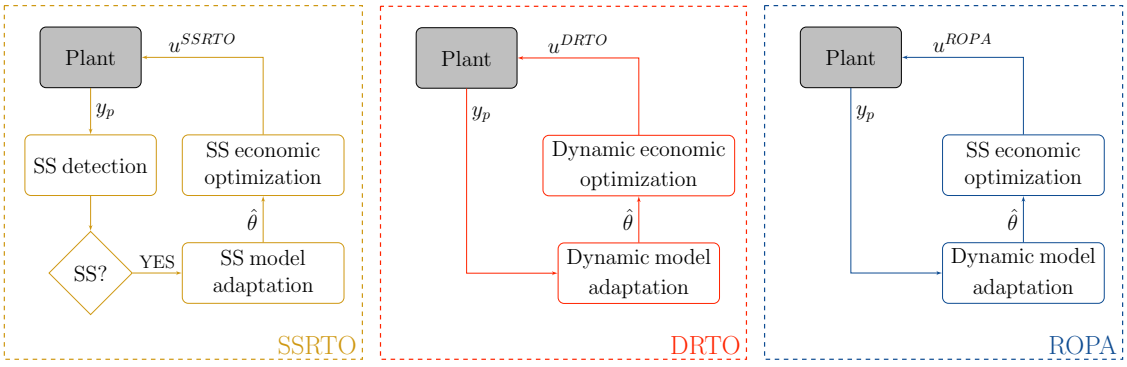
The issue of steady-state wait for updating the steady-state model can be addressed by using a dynamic real-time optimization (DRTO) approach (Figure 1b). Although DRTO is conceptually similar to SSRTO (i.e. a model adaptation step followed by an economic optimization step), it relies fully on a dynamic model of the plant. Thus, it is possible to use transient measurements in the adaptation step, avoiding the steady-state wait. In Figure 2, we schematically compare DRTO and SSRTO. Since we do not have to wait for a new steady-state, DRTO drives the plant to the new optimum immediately after the disturbance starts, reaching the new optimum faster than SSRTO. Despite being conceptually attractive, only a few real-world applications were reported in the literature (for example, Rawlings et al. (2018)).

Alternatively, a hybrid approach (Figure 1c) can be used, where we use transient measurements and a dynamic model of the system for the adaptation step (like in DRTO), whereas the economic optimization is performed using a steady-state model (like in SSRTO). As a consequence, the steady-state economic optimization can be scheduled at any arbitrary rate avoiding the need to wait for a steady-state. This scheme was independently proposed by Valluru et al. (2015), by Krishnamoorthy et al. (2018) using the name Hybrid RTO (HRTO), and by Matias and Le Roux (2018), who called it Real-time Optimization with Persistent Parameter Adaptation (ROPA). We will use the latter nomenclature in this paper.

In this paper, we implement the three approaches mentioned above on an experimental rig that emulates a subsea oil well network and compare the results. The main con-

*The authors acknowledge financial support from the Norwegian Research Council/Intpart, SUBPRO, grant number: 237893. Galo A. C. Le Roux acknowledges CNPq, the National Council for Scientific and Technological Development for his productivity grant 312049/2018-8.

*Corresponding author
johannes.jaschke@ntnu.no (J. Jäschke)
ORCID(s): 0000-0002-1094-6145 (J. Matias)



(a) Steady-state RTO (b) Dynamic RTO (c) RTO with persistent parameter adaptation

Figure 1: Block diagram comparing the three approaches. Here, u are the computed inputs (system manipulated variables); $\hat{\theta}$ the estimated parameters, and y_p the plant measurements.

tribution of this paper is to confirm the previous *in-silico* findings: since ROPA avoids the steady-state wait, the optimization frequency increases, improving the overall economic performance to levels similar to DRTO with a much lower computational effort. Along with using experimental evidence to support ROPA's capabilities, we also perform a theoretical analysis to illustrate ROPA's convergence to the steady-state optimum. Finally, we also propose some guidelines to its practical implementation. Our goal is to support practitioners with important model design decisions and also provide help for choosing values for ROPA's tuning parameters.

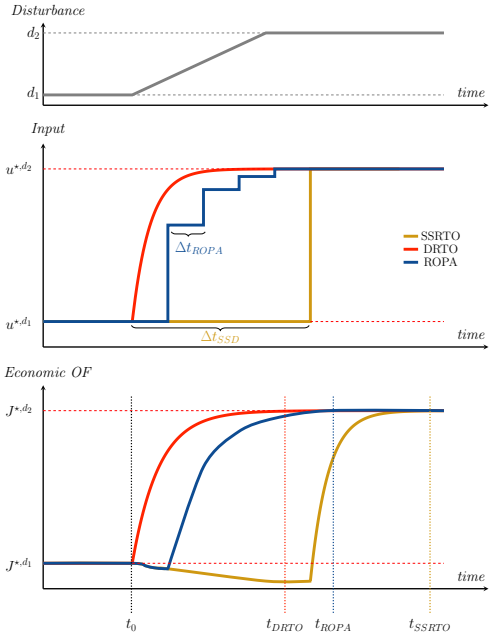


Figure 2: Idealized response of the three approaches to a ramp disturbance ($d_1 \rightarrow d_2$). The computed inputs $u^\star$ and the resulting economic index J are shown. t_{SSRTO}, t_{DRTO} and t_{ROPA} represent the time the approaches take to drive the plant to the new steady-state optimum $J^{\star,d_2}$. Δt_{SSD} is the steady-state wait time of the traditional RTO and Δt_{ROPA} is ROPA's execution rate.

2. Real-time optimization with persistent parameter adaptation - ROPA

ROPA works on the assumption that the SS predictions of a sufficiently accurate dynamic model are a good indication of the future plant steady-state. Then, by continuously estimating the model parameters, we capture the effect of the current disturbances in the plant steady-state that is yet to be achieved, i.e. current disturbances are measured and can be accounted for before their effect is fully realized in the system. Consequently, if we continuously adapt the value of inputs u based on this future steady-state, ROPA can drive the system to the desired stationary optimum.

Note that, since SSD is not necessary, ROPA can trigger the model adaptation and update the inputs u every Δt_{ROPA}, which is the approach's sampling time (Figure 2). It must be chosen such that relevant process dynamics and disturbances are captured. For a discussion on how to tune this parameter, see Section 6.3. As a consequence of frequently updating the input u, ROPA reaches the new optimal steady-state point $J^{\star,d_2}$ faster than SSRTO. In the *worst-case* scenario, its performance is *at least as* good as SSRTO. When compared to DRTO, the input sequence computed by ROPA is sub-optimal during the transients because they are not optimized.

2.1. Using a SS model instead of a dynamic for economic optimization

Despite having sub-optimal transients, there are advantages of running a steady-state instead of a dynamic economic optimization. As mentioned, one of the main challenges with DRTO is the large amount of computation required. Both the dynamic model adaptation and dynamic economic optimization problems need to be solved every sampling time, which may result in a computation delay and subsequently optimization performance degradation as well as system instabilities (Findeisen and Allgöwer, 2004).

On the other hand, solving only the dynamic model adaptation problem (which, in addition, can be carried out by a recursive method) is much more attractive from a computation cost point of view. Especially, if this alternative leads to sim-

Table 1
List of Acronyms

Acronym	*Full name*
DAE	Differential-algebraic equations
DRTO	Dynamic RTO
EKF	Extended Kalman filter
HRTO	Hybrid RTO
MHE	Moving Horizon Estimator
MILP	Mixed-integer Linear Programming
MINLP	Mixed-integer Nonlinear Programming
MPC	Model Predictive Control
ODE	Ordinary differential equation
ROPA	RTO with persistent excitation
RTE	Real-time Evolution
RTO	Real-time Optimization
SS	Steady-state
SSD	Steady-state Detector
SSRTO	Steady-state RTO
UKF	Unscented Kalman Filter

ilar economic performances. For example, Krishnamoorthy et al. (2018) applied ROPA and DRTO on a simulated subsea well network and showed that both approaches achieved very similar economic performance while ROPA had much lower computational requirements.

Another important advantage of using steady-state models is related to solving mixed-integer (non)linear programming (MILP/MINLP) optimization problems. For example, in subsea oil well networks, the well flow can be routed to one of different flowlines (Foss et al., 2018). If the routing becomes a decision variable of a dynamic economic optimization, the problem needs to be formulated as a dynamic MILP/MINLP, which is much more challenging to solve than the steady-state version.

2.2. ROPA building blocks

According to Figure 1, the two main building blocks of ROPA are *Dynamic model adaptation* and SS *economic optimization*, which are described in the next sections.

2.2.1. Dynamic model adaptation

In SSRTO, the model adaptation is typically formulated as a parameter estimation problem (Darby et al., 2011), in which the model parameters (or a subset of them) are updated such that the difference between model predictions and averaged plant measurements is minimized (see Section 3 for more details). In contrast, there are different strategies based on different paradigms for dynamic model adaptation. For instance: optimization-based methods, like moving horizon estimator (Rao and Rawlings, 2000), and recursive methods, such as recursive least squares estimation, extended Kalman filter (EKF), and unscented Kalman Filter (UKF). In this paper we will use the extended Kalman filter for adapting the model parameters.

Extended Kalman filter (EKF): this is the most common method used for nonlinear dynamic state/parameter adaptation in practice (Schneider and Georgakis, 2013). For implementing it, we model the system as:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)\,|\boldsymbol{\theta}(t)) + \boldsymbol{\upsilon}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t)|\boldsymbol{\theta}(t)) + \boldsymbol{\omega}(t) \end{aligned} \tag{1}$$

where, $\boldsymbol{x}$ are the model states, $\boldsymbol{u}$ the set of inputs (manipulated variables), $\boldsymbol{y}$ the model predictions, and $\boldsymbol{\theta}$ the model parameters. The variables $\boldsymbol{\upsilon}$ and $\boldsymbol{\omega}$ are the process and measurement noise, both are assumed to be uncorrelated zero-mean Gaussian random processes. The function $\boldsymbol{f}$ is the system dynamic model[1], and $\boldsymbol{h}$ is a function mapping $\boldsymbol{x}$ and $\boldsymbol{u}$ to the model outputs $\boldsymbol{y}$.

EKF is implemented by first using Equation 1 for evolving the current state estimates in time. Next, $\boldsymbol{f}$ and $\boldsymbol{h}$ are linearized using an appropriate method. Then, since the nonlinear model in Equation 1 is now approximated by a linear model, the estimates and their covariance matrix can be updated using the standard Kalman filter equations (Walter and Pronzato, 1997).

Since in ROPA context, we are also interested in estimating the parameters, we need to adapt our model in Equation 1. First, we assume that the parameter "dynamics" follow a Gaussian random walk model. Then, we extend $\boldsymbol{f}$ with the parameter "dynamics" as well as the state vector to include $\boldsymbol{\theta}$. As a result, we can use EKF's framework for obtaining an estimation of the parameter $\hat{\boldsymbol{\theta}}$. For more details, refer to e.g. Walter and Pronzato (1997).

2.2.2. Steady-state economic optimization

The economic optimization is executed repeatedly after the model is adapted at every Δt_{ROPA}. In this step, new inputs $\boldsymbol{u}$ are calculated such that an economic criterion J is optimized. The function J is typically chosen as profit (e.g. product value – feed costs – variable costs). The optimized inputs are then implemented in the plant. The SS economic optimization problem can be posed as:

$$\begin{aligned} \boldsymbol{u}^{\star}, \boldsymbol{x}^{\star} = \quad & \arg\max_{\boldsymbol{x},\boldsymbol{u}} \quad J(\boldsymbol{y}, \boldsymbol{u}|\hat{\boldsymbol{\theta}}) \\ \text{s.t.} \quad & 0 = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}\,|\hat{\boldsymbol{\theta}}) \\ & \boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}|\hat{\boldsymbol{\theta}}) \\ & \boldsymbol{g}(\boldsymbol{y}, \boldsymbol{u}) \leq 0 \end{aligned} \tag{2}$$

where, $\boldsymbol{x}$ are the model states, $\boldsymbol{u}$ the set of inputs (manipulated variables), $\boldsymbol{g}$ is the set of operational constraints and $\hat{\boldsymbol{\theta}}$, the current parameter estimates from the dynamic model adaptation step.

Note that we assume that parameters values are estimated such that the process model is a proper representation of the plant, i.e. no plant-model mismatch. Therefore, we do not consider the noise $\boldsymbol{\upsilon}$ and $\boldsymbol{\omega}$ models in the economic optimization (2). If that does not apply, production optimization techniques that cope with plant-model mismatch (e.g. Marchetti et al. (2016)), robust optimization schemes (e.g. Golshan

[1] For simplicity, we represent the model as a system of ordinary differential equations (ODE) but it could be easily adapted to a system of differential-algebraic equations (DAE).

et al. (2008)), or online model updated schemes (e.g. Matias and Jäschke (2021)) can be used.

To avoid upsetting the plant by implementing large input changes between the sample times, it is common to filter computed input sequence. Thus, it is ensured that the plant gradually is moved to a new operation point, which is desirable for practical reasons. That is, the input implemented at RTO iteration $k+1$ is given by

$$u(k+1) = (1 - K_u)\boldsymbol{u}^\star(k) + K_u\boldsymbol{u}^\star(k+1) \tag{3}$$

where $\boldsymbol{u}^\star(k)$ is the optimal input computed at sample time k. In a similar manner, the estimated parameter $\hat{\theta}$ can be filtered with a parameter K_θ to realize a smoother operation.

2.3. ROPA convergence analysis

Despite updating a SS model with parameters estimated from transient measurements, we show here that ROPA always drives the process to its steady-state optimum under some assumptions. Its convergence is demonstrated based on the SSRTO Point-Wise Stability test (Forbes and Marlin, 1996). We start the analysis by defining two stability concepts.

Definition 2.1 (Lyapunov and asymptotic stability (Wiggins, 2003))**.** *An equilibrium point $\boldsymbol{u}_{SS}$ of a system of recursive algebraic equations $\mathcal{F}(\boldsymbol{u}_k, k) \mapsto \boldsymbol{u}_{k+1}$ is said to be Lyapunov stable, if:*

$\forall \epsilon > 0$, $\exists \delta > 0$ *such that, if* $\|u_0 - u_{SS}\| < \delta$, *then* $\forall k \geq 0$, $\|u_k - u_{SS}\| < \epsilon$

The point $\boldsymbol{u}_{SS}$ is said to be asymptotically stable if:

it is Lyapunov stable and $\exists \kappa > 0$ *such that, if* $\|u_0 - u_{SS}\| < \kappa$, *then* $\lim_{k\to\infty} \|u_k - u_{SS}\| = 0$

Note that we have given the definition above in terms of $\boldsymbol{u}$, and not $\boldsymbol{x}$, because the states $\boldsymbol{x}$ can be formally eliminated using the model equations. Now, consider ROPA diagram of Figure 1c. At steady state, each block can be represented as a static non-linear map:

$$\boldsymbol{u}^\star = \mathcal{E}(\boldsymbol{\theta}), \quad \boldsymbol{\theta} = \mathcal{A}(\boldsymbol{y}), \quad \boldsymbol{y} = \mathcal{P}(\boldsymbol{u}, t) \tag{4}$$

where, $\mathcal{E}$ maps the parameter value $\boldsymbol{\theta}$ to the optimal input $\boldsymbol{u}$ (SS economic optimization); $\mathcal{A}$ is the dynamic model adaptation block; and $\mathcal{P}$ accounts for the dynamic plant, which maps the inputs and the sampling time into the process measurements $\boldsymbol{y}$. For arbitrarily small deviations of their inputs and assuming that the appropriate derivatives exist, the maps above can be represented by their first-order approximations around the current values:

$$\begin{aligned}
\delta\boldsymbol{u}^\star &= \left.\frac{d\boldsymbol{u}^\star}{d\boldsymbol{\theta}}\right|_{\hat{\theta}_k} \delta\boldsymbol{\theta} \\
\delta\boldsymbol{\theta} &= \left.\frac{d\boldsymbol{\theta}}{d\boldsymbol{y}}\right|_{y_{p,k}} \delta\boldsymbol{y} \\
\delta\boldsymbol{y} &= \left.\frac{d\boldsymbol{y}}{d\boldsymbol{u}}\right|_{\boldsymbol{u}^\star_{ROPA,k}} \delta\boldsymbol{u} + \underbrace{\left.\frac{d\boldsymbol{y}}{dt}\right|_{t_k} \cancel{\delta t}^{\Delta t_{ROPA}}}_{\approx 0}
\end{aligned} \tag{5}$$

Note that, ROPA execution period Δt_{ROPA} is chosen such that the most significant dynamic effects (related to the SS economic optimization time scale) are negligible. As in Matias and Le Roux (2018), we assume that the model is structurally identifiable (i.e. $\boldsymbol{\theta} \mapsto \boldsymbol{y}$ is one-to-one), and $\mathcal{E}$ has an unique solution for a given value of the parameters (see the conditions in Fang et al. (2009)). If the assumptions hold, we can rearrange the three separate systems of linear equations above into a single recursive system:

$$\delta\boldsymbol{u}^\star(k+1) = \left.\left(\frac{d\boldsymbol{u}^\star}{d\boldsymbol{\theta}}\frac{d\boldsymbol{\theta}}{d\boldsymbol{y}}\frac{d\boldsymbol{y}}{d\boldsymbol{u}}\right)\right|_{\hat{\theta}_k, \boldsymbol{y}_{p,k}, \boldsymbol{u}^\star_{ROPA,k}} \delta\boldsymbol{u}^\star(k) \tag{6}$$

where, k is ROPA iteration index.

Theorem 2.1 (ROPA convergence)**.** *If the plant gradients (5) are boundend, the steady-state input $\boldsymbol{u}^\star_{SS}$ is an asymptotically stable equilibrium point of the system in Equation (6).*

Proof. All gradients in (5) are bounded. The first two terms in (6) can be made arbitrarily small by decreasing the filter parameters K_u and K_θ. As the last term inside the parenthesis of (6) is also bounded, the filter parameters K_u and K_θ can be selected such that $\forall k$:

$$\left\| K_u \left.\frac{d\boldsymbol{u}^\star}{d\boldsymbol{\theta}}\right|_{\hat{\theta}_k} K_\theta \left.\frac{d\boldsymbol{\theta}}{d\boldsymbol{y}}\right|_{y_{p,k}} \left.\frac{d\boldsymbol{y}}{d\boldsymbol{u}}\right|_{\boldsymbol{u}^\star_{ROPA,k}} \right\| < 1. \tag{7}$$

Hence, all the eigenvalues of the system in Equation (6) can be placed inside the unit circle. As a consequence, we know that $\lim_{k\to\infty} \delta\boldsymbol{u}^\star(k) = 0$ and the system is asymptotically stable. Therefore, if proper values of K_u and K_θ are chosen, ROPA is able to stabilize the system at a given steady-state value $\boldsymbol{u}^\star_{SS}$ for arbitrarily small (differential) changes in $\boldsymbol{u}$. □

Furthermore, if we assume that the solution of Equation (2) satisfy the first and second-order conditions of optimality at the plant optimum (Ganesh and Biegler, 1987), $\boldsymbol{u}^\star_{SS}$ will be a local optimum of the plant.

2.4. Related work

The use of transient measurements to speed up static production optimization has been investigated in the literature

since the 70s. The first studies in this area (e.g. Sawaragi et al. (1971); Bamberger and Isermann (1978)) used transient measurements to estimate the parameters of a simplified dynamic input-output model of the process online. Next, the steady-state gradients of the system were calculated using the updated model. Then, a gradient search algorithm was employed to solve the optimization problem. Despite being successfully applied to some pilot plants (for example, cooling water circulation of a thermal power plant (Bamberger and Isermann, 1978), a distillation column (Sawaragi et al., 1971), and a fixed bed reactor (Lee and Lee, 1985)), the proposed methods did not handle constraints.

Since the optimum operation of chemical processes often lies at the constraints (Bhattacharya and Joseph, 1982), new developments were suggested such as approximating the nonlinear constrained optimization through Successive Linear Programming (Bhattacharya and Joseph, 1982), and obtaining an unconstrained problem by rearranging the system of equations (McFarlane and Bacon, 1989). Even though the constraints were included in these methods, the optimality and feasibility of the optimization problem solution depends on the accuracy of the gradient estimates. Inaccuracies in the dynamic model parameters can lead to oscillatory behavior of the objective function (Bhattacharya and Joseph, 1982).

For obtaining more informative measurements, excitation signals (e.g. pseudo random binary sequences) were added to the inputs continuously (Lee and Lee, 1985), or at least during an initial identification phase (Garcia and Morari, 1984). However, extra perturbations for identification purposes may not be desired from an operational point of view. Moreover, even if the accuracy problem is solved, steepest descent algorithms converge slowly to the optimum for most problems (Golden and Ydstie, 1989).

In the 80s, the two-step RTO became a central technology for production optimization. Although the combination of rigorous models with optimization algorithms improved the convergence speed, the bottleneck shifted to the steady-state wait. All the problems associated with this issue were already discussed in details in the introduction of this paper. The first reported attempts to speed up production optimization in the context of steady-state RTO were based on the idea of a "*step in the right direction should be better than to wait until the process has settled to a new steady state*" (Engell, 2007).

In Besl et al. (1998), the authors updated the model to the plant condition when the plant reached steady-state; however, the production optimization algorithm was schedule at a fixed rate, independently whether or not the plant was at steady-state. Similarly, Prior and Lopez (1999) updated individual sections asynchronously when they reached steady-state and ran the optimization at a fixed rate using using the latest filtered parameter updates. In an interesting advance, Sequeira et al. (2002) proposed a method called Real-time Evolution (RTE). In RTE, steady-state information is used for data reconciliation and model updating. Then, small adjustments in the inputs are computed with a given frequency and within a small neighborhood around current operating conditions. This optimum search mechanism is similar to EVOP (Box, 1957) but based on the steady state model instead of plant perturbations.

Even though the approaches above reported significant benefits, they can only be applied to quasi-stationary systems since they neglect dynamics. Also, the infrequent model update can decrease predictive capability of the steady-state models. This shortcoming was partially addressed by Rodrigues et al. (2018). The authors proposed an approach to speed up SSRTO implementations by estimating the current output change rate and its future steady-state effect using a pure data-based strategy. The disadvantages are: the system dynamics needs to be characterized by two different timescales (fast and slow states); the unknown part of the dynamics must depend only on the fast states; and the overall performance is very sensitivity to noise.

Since the use of this data-driven strategy restricts the amount of noise admissible for obtaining accurate steady-state estimates (Rodrigues et al., 2018), the natural next step is to apply a dynamic model to estimate the system dynamics evolution, which is the strategy used in ROPA. The main challenge becomes obtaining an accurate dynamic model for parameter estimation. This issue was addressed in Delou et al. (2021), where the authors use a Hammerstein model structure to represent the system dynamics. This structure exploits the static model with added dynamics by a linear autoregressive model (ARX) identified from plant data or experiment.

Various other approaches have been proposed in the context of ROPA. Valluru and Patwardhan (2019) and Santos et al. (2021) proposed using the dynamic estimator to updated the model predictive control (MPC) model in order to avoid mismatch between RTO and MPC layers. Shamaki and Odloak (2020) combined ROPA with a infinite horizon zone control MPC. Curvelo et al. (2021) studied ROPA's performance when applied to systems with different transient patterns, such as long time delays and non-minimum phase behavior. ROPA has also been extended to a plant-wide optimization context using an asynchronous parameter adaptation strategy (Matias and Le Roux, 2020). Here, only a subset of the plant-wide steady-state model parameters is adapted dynamically. This strategy allows the *plant-wide* optimization cycle to be triggered much more frequently. From an economic point of view, it is advantageous to solve the plant-wide economic problems since the trade-offs between plant units are taken into account (Friedman, 1995). Moreover, the complete model is solved and no decomposition techniques are required.

Despite being an active research topic, the only experimental validation of the method was proposed by Matias et al. (2021). Here, the authors focused on the improvement related to avoiding the steady-state detection rather than the loss of performance connected to not optimizing the transients, which is one of the main contributions of this paper.

Note that the trend of combining transient measurements with static optimization can also be seen in different pro-

duction optimization strategies, such as Modifier Adaptation (François and Bonvin, 2013; Rodríguez-Blanco et al., 2017; Gao and Engell, 2017), Extremum-seeking control (Guay et al., 2005; Krishnamoorthy et al., 2019);, and feedback-based RTO methods (Dirza et al., 2021).

3. SSRTO and DRTO implementation

The main goal of the paper is to compare the performance of ROPA with the state-of-the-art production optimization approaches (steady-state and dynamic RTO) in a physical system. In the next subsections, we briefly present both approaches and discuss their main characteristics. For a more extensive review of SSRTO, refer to (Darby et al., 2011). For DRTO, see e.g. (Srinivasan et al., 2003). Despite being focused on batch processes, this paper has a complete description of DRTO building blocks and solution methods.

3.1. Steady-state RTO

3.1.1. SS Detection

The first step of the SSRTO implementation is the steady-state detection (Figure 1a). There are many different SSD procedures available in the literature. Typically, they compare intervals of measurements using statistical properties, such as hypothesis tests using F-statistic (Alekman, 1994), Student t-test (Kelly and Hedengren, 2013), and R-statistic combined with first-order filters for the measurements (Cao and Rhinehart, 1995). Although the methods rely on statistical theory, they strongly depend on tuning inputs, such as filter gains and tolerances (Câmara et al., 2016). Therefore, it is common that different procedures applied to the same data set yield different results (Menezes, 2016). Independently of the chosen procedure, high-frequency or persistent disturbances can hinder the start of the optimization cycle. Also, if the system to be optimized is composed of several units, SSRTO is triggered only if *all* units are at SS, which significantly decreases the number of optimization runs (Matias and Le Roux, 2020). In our implementation, this step is carried out using a Student t-test described in details in Appendix A.

3.1.2. Model adaptation and economic optimization

After detecting the steady-state, the SS model adaptation step is triggered (see Figure 1). The goal is to find the values of $\boldsymbol{\theta}$ that minimize the difference between the SS model predictions $\boldsymbol{y}$ and the current plant measurements $\boldsymbol{y_p}$. The model adaptation problem is posed as:

$$\begin{aligned} \hat{\boldsymbol{\theta}},\ \hat{\boldsymbol{x}} = \quad & \underset{\boldsymbol{x},\boldsymbol{\theta}}{\arg\min} \quad ||\boldsymbol{y_p} - \boldsymbol{y}||_V \\ \text{s.t.} \quad & 0 = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u_p} \,|\boldsymbol{\theta}) \\ & \boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u_p}) \\ & \boldsymbol{\theta} \in \boldsymbol{\Theta} \end{aligned} \tag{8}$$

where, $\boldsymbol{\Theta}$ is the allowable parameter constraint set; and $||\cdot||$ is a norm weighted by a matrix V, which is usually chosen as the identity matrix or inverse of the covariance of the measurements $\boldsymbol{y_p}$. $\boldsymbol{u_p}$ are the inputs currently on the plant. After adapting the parameters, the SS economic optimization uses the updated steady-state model to find $\boldsymbol{u}^{SSRTO}$. This step is performed by solving the optimization problem given by Equation (2) (as in ROPA).

3.2. Dynamic RTO

The dynamic parameter estimation step is carried out in the same way as for ROPA. We also use EKF in the dynamic RTO implementation, but any other dynamic estimator is applicable.

3.2.1. Dynamic Economic Optimization

The dynamic economic optimization problem is defined on a prediction horizon N_p. It is stated as follows:

$$\begin{aligned} \boldsymbol{u}^{\star}(t) = & \underset{\boldsymbol{x}(t),\boldsymbol{u}(t)}{\arg\max} \int_{t_0}^{t_0+N_p} \boldsymbol{J}(\boldsymbol{y}(t),\boldsymbol{u}(t)) + \dot{\boldsymbol{u}}(t)^T \boldsymbol{R}\, \dot{\boldsymbol{u}}(t) dt \\ \text{s.t.} \quad & \text{on } t \in [t_0, t_0 + N_p] \\ & \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t) \,|\hat{\boldsymbol{\theta}}(t)), \qquad \boldsymbol{x}(t_0) = \hat{\boldsymbol{x}} \\ & \boldsymbol{y}(t) = \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \\ & \boldsymbol{g}(\boldsymbol{y}(t), \boldsymbol{u}(t)) \leq 0 \\ & |\dot{\boldsymbol{u}}(t)| \leq \dot{\boldsymbol{u}}_{\max} \end{aligned} \tag{9}$$

All the symbols were previously defined, except the input movement $\dot{\boldsymbol{u}}$, where the dot symbol represents the time derivative. $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{\theta}}$ are the current estimates of the states and parameters, and $\boldsymbol{u_p}$ the current implemented inputs. For the implementation, the system is discretized in time using, for example, orthogonal collocation on finite elements, and the input signal $u(t)$ is assumed to be piecewise constant on these elements (Biegler, 2007). Then, the solution $\boldsymbol{u}^{\star}(t)$ can be represented by a finite sequence $\boldsymbol{U}^{\star} = [\boldsymbol{u_0}^{\star}, \boldsymbol{u_1}^{\star}, \ldots]$. The problem is then solved repetitively at each sampling time of the system, and only the first control move in $\boldsymbol{U}^{\star}$ is implemented.

4. Case study: Subsea oil well network

In subsea oil production, the goal is to extract oil and gas trapped in subsea geological structures, which is achieved by drilling several wells in these hydrocarbon reservoirs. The production of the well network is led to processing facilities on sea level by long vertical pipelines, known as risers. These facilities are responsible for separating the reservoir outflow fluids, typically gas, oil and water.

Typically, the reservoir pressure drives the fluids from below the seafloor to the top facilities. If this pressure is not large enough, artificial lifting methods, such as electrical submersible pumps, subsea boosting stations, and gas lifting, need to be applied. The latter is commonly used since it has a robust design and relatively low-cost (Amara, 2017).

In gas lifted systems, the excess gas that is produced is compressed and injected back in the well (Hernandez, 2016). As a result, fluid bulk density is reduced, decreasing the hydrostatic bulk pressure on the reservoir and increasing the

production. Since the reservoirs can be located several kilometers below sea level, such a decrease has significant positive effect on the system productivity. However, if the gas injection flowrate becomes too large, the effect of frictional pressure drop in the pipelines dominates, decreasing the gain of injecting more gas to the system. The gas lift effect is illustrated by Figure 3.

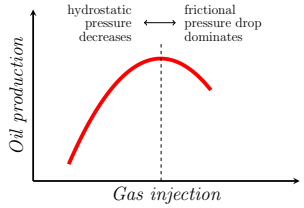


Figure 3: Gas lift effect on well's oil production. The dashed line indicates the turning point, in which the frictional pressure drop effect starts to dominate the hydrostatic pressure.

On a daily basis, operators and engineers are responsible for deciding the gas lift injection flowrates such that the system operates as efficiently as possible, maximizing the revenue (Foss et al., 2018). This decision process can be automated and improved by the usage of production optimization approaches, such as SSRTO, ROPA, and DRTO.

4.1. Experimental rig

To study how the different approaches perform in a subsea oil well network, a small-scale experimental rig was built. For simplification purposes, the setup uses water and air as working fluids instead of oil and gas. Since the change in hydrostatic pressure induced by gas injection is the phenomenon of interest, the denser the liquid, the larger the gas lift effect. Thus, by changing the fluids we not only enhance the gas lift effect but also have a cheaper and more environmentally-friendly setup. Moreover, water and air have already been used for replacing oil and gas in experimental rigs that represent subsea oil wells (see, e.g. Jahanshahi et al. (2017)).

A simplified flowsheet of the rig is shown in Figure 4. The system is divided into three sections:

Reservoir: this section is composed of a 200 L stainless steel tank, a centrifugal pump, and three control valves (CV101, CV102, and CV103). We can freely manipulate the openings of these valves, which are chosen for representing different reservoir behaviors. With this setup, the reservoir produces only liquid and its outflow ranges from 2 L min^{-1} to 15 L min^{-1}. Flow meters (FI101, FI102, and FI103) are located before the reservoir valves. A controller regulates the pump rotation such that the its outlet pressure (PI104) remains at a given setpoint, which is kept constant at 0.3 barg in the experiments;

Wells: three parallel flexible hoses with 2 cm inner diameters and length of 1.5 m represent the wells. Approximately 10 cm after the reservoir valves, air is injected by three air flow controllers (FIC104, FIC105, and FIC106) within the range of 1 sL min^{-1} to 5 sL min^{-1};

Risers: this section is composed of three vertical pipelines, orthogonal to the well section, with 2 cm inner diameters and 2.2 m high. We measure the pressures on top of the risers (PI101, PI102, and PI103). After the sensors, we have three manual valves which are kept open during the experiments. The air is vented out to the atmosphere. For environmental purposes, the liquid is recirculated to the reservoir water tank.

5. Experimental Setup

The optimal operation point of the system is achieved by maximizing the "oil" revenue, while accounting for gas availability constraints and bounds on the gas lift flowrates. The economic objective function J is chosen as:

$$J = 20\, Q_{l,1} + 10\, Q_{l,2} + 30\, Q_{l,3} \tag{10}$$

where, Q_l are the liquid flowrates of wells 1, 2, and 3. For illustration purposes, we assumed that the wells have different valued hydrocarbons, which are reflected by different weights in J. The constraint set g in Equation (2) and (9) is composed by the gas lift injection Q_g lower and upper bounds ($Q_{g,min} = 1$ sL min^{-1} and $Q_{g,max} = 5$ sL min^{-1}) and the gas availability constraint ($Q_{g,1} + Q_{g,2} + Q_{g,3} \leq 7.5$ sL min^{-1}).

To study ROPA's performance in comparison with the other approaches, we run experiments of 20 min and change the opening of the valves CV101, CV102 and CV103 according to Figure 5, while keeping the pump outlet pressure P_{pump} constant. Since the holdup inside the pipes is small, the system response to changes in the gas flowrates is fast (<1 min). The reservoir valves are used to emulate slow time-scale and persistent reservoir disturbances. This is similar to what happens in practice with a significant time scale separation between the wells and reservoir (Foss et al., 2018). This disturbance scenario emulates the well's depletion (i.e. declining oil production over time), where larger valve openings indicate larger reservoir outflows.

The initial value of the inputs is specified as $Q_{g,1} = Q_{g,2} = Q_{g,3} = 2.5$ sL min^{-1}. During the experiment, we want to find the best gas lift distribution among the three wells. For carrying out this task, we implement the three production optimization approaches and a naive strategy, in which the inputs are fixed as 2.5 sL min^{-1} during the entire experiment. This specific distribution is obtained by dividing the maximum gas availability equally by the three wells. Since this is the natural choice when no information is available about the system, this is used as a baseline for the performance comparison. The procedures for implementing ROPA, SSRTO and DRTO are presented in Algorithms 1, 2, and 3, respectively. The codes used in the experimental rig are available in our Github page[2]. In Appendix B, we show the tuning parameters for the three approaches.

Note that we use first order input filters in the implementation ROPA (Algorithm 1/Step 10) and SSRTO (Algo-

[2] https://github.com/Process-Optimization-and-Control/ProductionOptRig

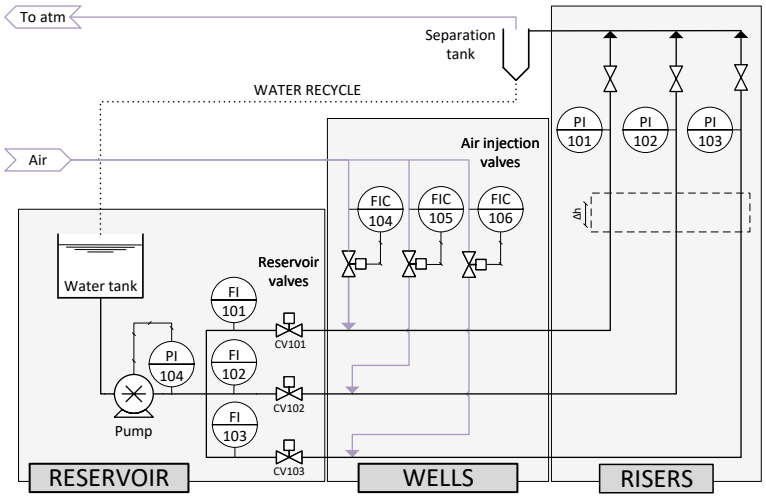


Figure 4: Experiment schematic. The system measurements $\boldsymbol{y}_p$ are the well top pressures (PI101, PI102 and PI103), the pump outlet pressure (PI104), the liquid flowrates (FI101, FI102, and FI103), and the gas flowrates (FI104, FI105, and FI106). Three PI controllers are used for controlling the gas flowrates, whose setpoints are the system inputs $\boldsymbol{u}$ computed by the optimization layer. The reservoir valve openings (CV101, CV102, and CV103) are the system disturbances. They change during the experiments for representing different reservoir behaviors, while the pump outlet pressure is kept constant by a PI controller.

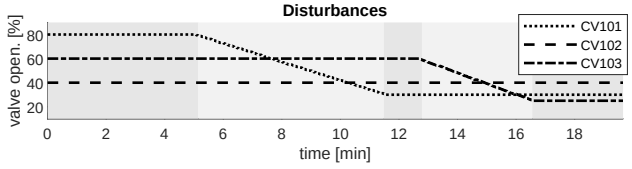


Figure 5: Disturbance profile. Changes in CV101 and CV103 openings emulate a declining oil production in wells 1 and 3.

Algorithm 1 ROPA

1: Get plant measurements $\boldsymbol{y}_{p,k}$ and inputs $\boldsymbol{u}_{p,k}$, where k indicates the current time instant

Dynamic model adaptation (EKF)

2: Define an extended state with $\hat{\boldsymbol{x}}_k^e = [\hat{\boldsymbol{x}}_k,\ \hat{\boldsymbol{\theta}}_k]^T$

3: Use random walk for parameter evolution characterization: $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \boldsymbol{\upsilon}_k, \quad \boldsymbol{\upsilon} \sim \mathcal{N}(\mathbf{0}, Q_\theta)$

4: Obtain the affine version of the dynamic model $\boldsymbol{f}(\cdot,\cdot|\cdot)$ using the sensitivity equations and extend it with the parameter evolution model from Step 3

5: Update extended state estimate covariance matrix and the estimator gain

6: Compute the extended state estimate based on the current prediction error

7: Obtain parameter estimates $\hat{\boldsymbol{\theta}}_{k+1}$ based on current measurements

Steady-state economic optimization

8: Update steady-state model $0 = \boldsymbol{f}(\cdot,\cdot|\hat{\boldsymbol{\theta}}_{\boldsymbol{k}})$

9: Compute $\boldsymbol{u}^\star_{k+1}$ using Equation (2)

10: Apply input filter $\boldsymbol{u}_{k+1} = \boldsymbol{u}_{p,k} + K_u(\boldsymbol{u}^\star_{k+1} - \boldsymbol{u}_{p,k})$

11: Implement $\boldsymbol{u}_{k+1}$

Algorithm 2 SSRTO

1: Get plant measurements $\boldsymbol{y}_{p,k}$ and inputs $\boldsymbol{u}_{p,k}$

2: Steady-state detection of $\boldsymbol{y}_{p,k}$

If Steady-state is $True$

Steady-state model adaptation

3: Compute parameter estimates $\hat{\boldsymbol{\theta}}_{k+1}$ using Equation (8)

Steady-state economic optimization

4: Update steady-state model $0 = \boldsymbol{f}(\cdot,\cdot|\hat{\boldsymbol{\theta}}_{\boldsymbol{k}})$

5: Compute $\boldsymbol{u}^\star_{k+1}$ using Equation (2)

6: Apply input filter $\boldsymbol{u}_{k+1} = \boldsymbol{u}_{p,k} + K_u(\boldsymbol{u}^\star_{k+1} - \boldsymbol{u}_{p,k})$

7: Implement $\boldsymbol{u}_{k+1}$

Else

8: Do nothing

Algorithm 3 DRTO

1: Get plant measurements $\boldsymbol{y}_{p,k}$ and inputs $\boldsymbol{u}_k$

Dynamic model adaptation (EKF)

2: Implement Step 2 to 7 from Algorithm 1

Dynamic economic optimization

3: Update dynamic model $\boldsymbol{f}(\cdot,\cdot|\hat{\boldsymbol{\theta}}_{\boldsymbol{k}})$

4: Compute $\boldsymbol{U}^\star_{k+1}$ using Equation (9)

5: Extract $\boldsymbol{u}_{k+1}$ from $\boldsymbol{U}^\star_{k+1}$

6: Implement $\boldsymbol{u}_{k+1}$

rithm 2/Step 6) as well as input regularization in the objective function of the dynamic economic optimization (Equation 9). Despite affecting the economic performance, penalizing large input variations is necessary for practical applications. For example, since the wells in our problem are identical, small deviations in their models, which can result from slightly different updates in their parameters, may lead

to significant changes in the optimal gas lift flowrate distribution. Potentially, these values can go from the minimum to the maximum limits in one iteration. This type of decision can easily raise credibility issues on the plant engineers and operators concerning the production optimization approaches' performance. Thus, they need to be avoided and input filters/regulation are an interesting alternative.

6. ROPA Implementation

Before showing the main results regarding the comparison of the three approaches, we present some recommendations for ROPA's practical implementations[3]. They are mostly based on lessons learned during the implementation on the experimental rig and can be used as guidelines for other systems. They are focused on three aspects that should be defined before the online deployment: *Process modeling*, *online parameter and state estimation*, and ROPA *execution period tuning*.

6.1. Process modeling

The first step for implementing ROPA is obtaining the dynamic and steady-state models. Their complexity should be carefully taken into account. The models should reflect the main economic trade-offs and accurately predict the system operating constraints. The more precise the model, the better the production optimization performance (Câmara et al., 2016). However, if the model is too complex, the model adaptation task becomes more challenging and little value is then added over a simplified version (Darby et al., 2011). Additionally, model maintenance also becomes harder.

6.1.1. Experimental rig modeling

The goal of the production optimization approaches is to determine the optimal injection rate (i.e. the *setpoints* of flow controllers FIC104, FIC105, and FIC106) such that J is maximized, while considering maximum gas availability constraints and gas injection bounds. Thus, when modeling the system, we should take the following into account:

(a) Since the system dynamics are mainly related to the reservoir, the wells' dynamics do not have to be considered in detail. Therefore, the momentum balance can be simplified, i.e. the pressure dynamic does not need to be taken into account given that its time scale is much faster than the reservoir time scale;

(b) The phenomenon of interest is gas lift. Hence, both the hydrostatic pressure and friction pressure loss need to be computed. However, since the pressure difference between the bottom and top of the riser is not significant, the gas cooking-off effect (Hernandez, 2016) does not need to be considered. Thus, the pressure spatial variance is replaced by two lumped pressures, one at the bottom and another one at top of the riser;

(c) The main system constraint is gas availability. Since the gas flow rate is measured, the model precision with relation to this constraint is not an issue.

[3] In Appendix A, we also briefly discuss the implementations of SSRTO and DRTO

Given the considerations above, the following dynamic model is derived. It is based on Krishnamoorthy et al. (2018). We only show the equations for one well, but the extension to three wells is straightforward. The static model is obtained by setting the time derivatives to zero. A model diagram is shown in Figure 6.

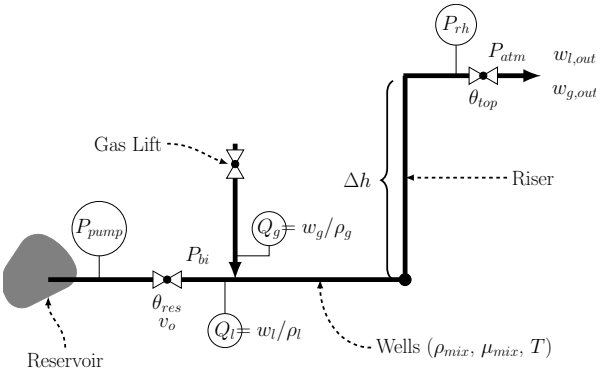


Figure 6: Diagram of a single-well model. P_{rh}, P_{pump} and Q_l are measured. Q_g is controlled by the gas flowrate controller and the reservoir valve opening v_o is assumed as a measured disturbance.

Differential equations. The liquid and gas mass balances are represented by:

$$\dot{m}_g = w_g - w_{g,out} \tag{11a}$$

$$\dot{m}_l = w_l - w_{l,out} \tag{11b}$$

where, m_l and m_g are the liquid and gas mass holdups inside the wells and riser. The dot symbol represents the time derivative. w_g is the gas injection mass flowrate and w_l is the liquid flowrate coming from the reservoir. $w_{g,out}$ and $w_{l,out}$ are the outlet production rate of gas and liquid.

Algebraic equations: The reservoir outflow is obtained by the following relationship:

$$w_l = v_o \theta_{res} \sqrt{\rho_l (P_{pump} - P_{bi})} \tag{12}$$

where, ρ_l is the liquid density, θ_{res} is the reservoir valve flow coefficient, and v_o the valve opening. The pump outlet pressure P_{pump} is measured and the pressure before the injection point P_{bi} is computed taking into account the hydrostatic pressure and the pressure drop due to friction (as a simplification, we use the Darcy-Weisbach expression for laminar flow in cylindrical pipes). Thus, P_{bi} becomes:

$$P_{bi} = P_{rh} + \rho_{mix} g \Delta h + \frac{128 \mu_{mix} (w_g + w_l) L}{\pi \rho_{mix} D^4} \tag{13}$$

where, P_{rh} is the pressure at the riser head, which is measured. Δh, L and D are the height, length (well + riser), and diameter of the pipes. g is the gravitational acceleration. μ_{mix} is the mixture (liquid + gas) viscosity. In the experimental setup, the mixture viscosity is approximated as the liquid viscosity. The mixture (liquid + gas) density ρ_{mix}

is obtained by:

$$\rho_{mix} = \frac{m_{total}}{V_{total}} = \frac{m_g + m_l}{V_{total}} \tag{14}$$

which is complemented by an equation indicating that the summation of the gas V_g and liquid V_l volumetric holdups is equal to the total system volume:

$$V_{total} = V_g + V_l = \frac{m_l}{\rho_l} + \frac{m_g}{\rho_g} \tag{15}$$

The liquid density ρ_l is assumed constant, whereas the gas density ρ_g is computed using the ideal gas law:

$$\rho_g = \frac{P_{bi} M_g}{RT} \tag{16}$$

where, M_g is the air molecular weight, R the gas universal constant, and T the room temperature. The total outlet flowrate is obtained by the following relationship:

$$w_{total} = w_{g,out} + w_{l,out} = \theta_{top}\sqrt{\rho_{mix}(P_{rh} - P_{atm})} \tag{17}$$

where, P_{atm} is the atmospheric pressure, and θ_{top} is the top valve flow coefficient. We make an additional assumption that the proportion between liquid and total outlet flowrate is the same as the liquid fraction in the mixture α_l, i.e.:

$$\alpha_l = \frac{m_l}{m_{total}} = \frac{w_{l,out}}{w_{total}} \tag{18}$$

6.2. Online estimation of parameters and states

In order to adapt the model to the current plant condition, its states and parameters can be updated. However, in order to obtain reliable updates of both, we need to ensure that the states are observable and the model parameters are identifiable from the system outputs. Both observability and structural identifiability are properties connected to the model equations, meaning that they can be determined by the system dynamics and output alone. For more details about these properties, refer to Walter and Pronzato (1997) and Ljung (1999).

Typically, structural unidentifiability is connected to unobservability, in a sense that the wrong estimates of the parameters may lead to wrong predictions of the states. However, one does not imply the other (Villaverde, 2019). Hence, it is preferable to carry out both analysis (either simultaneously or separately) for decreasing the possibility of drawing false conclusions from the model of interest. This becomes even more significant when the model is used for production optimization, since a poor model adaptation can have a significant negative impact on the overall economic performance (Quelhas et al., 2013).

For our system, we decided to use a two-tiered approach, where we first choose a set of model parameters that can be properly identified at stationary periods (i.e. they do not require any type of specific excitation to be properly estimated) by performing a steady-state identifiability test. Next, since we consider the system parameters as additional states with trivial dynamics (Gaussian random walk model, as discussed in Section 2.2.1), the identifiability and observability analysis are performed together using an observability framework.

6.2.1. Tier 1: Steady-state parameter identifiability

Given that our goal is to implement the estimator in an experimental rig, we should account not only for the model structure but also for limitations regarding experimental data quality/availability when choosing the estimable parameter set. Therefore, we run a practical identifiability analysis in this step. This analysis is related to its structural counterpart; however, the focus shifts to assess the effect of noisy measurements into the parameter estimation performance.

To run the practical identifiability analysis, we need to collect experimental data that properly represent operational situations. The goal is to ensure that the inputs that are typically applied to the plant are able to excite the system enough and, consequently, guarantee that the parameter estimates converge to their true value. Since the input moves generated by the optimization may not fulfill this requirement, using data from previous experimental runs may affect the analysis outcomes. For example, when using a given input sequence, the analysis may indicate that parameters can be uniquely identified; however, if the data sequence is slightly changed, the parameters would not be identifiable anymore.

As an alternative, the practical identifiability analysis can be carried out using stationary data and the steady-state version of the dynamic model. Then, the results of the analysis will not be linked to any specific excitation pattern to ensure convergence. That is, as the steady-state model is identifiable, the extended dynamic model is also identifiable (Schei, 2008). Hence, there is no need for persistence of excitation as there is in general system identification to guarantee parameter identifiability.

Experimental rig model parametrization

A good starting point for selecting the adjustable parameters is to choose the ones that can be updated such that the most significant process disturbances are represented in the model. For example, since our main disturbances are associated with the reservoir valves (CV101, CV102, and CV103), it is interesting to estimate parameters associated with the behavior of these valves. Additionally, by selecting the number of estimable parameters to be smaller or equal to the number of measured variables, zero steady-state deviations in the predicted outputs can typically be achieved (Schei, 2008).

In the system of interest, we have 6 available measurements, one liquid flowrate and one pressure for each well (see Figure 4). By analyzing the model structure, the parameter candidates are: for each well, the reservoir θ_{res} and top valve θ_{top} constants, the mixture viscosity μ_{mix}, and the liquid fraction α_l. Note that, if we choose μ_{mix} and α_l, we need to exclude some of the model simplifying assumption, namely: mixture viscosity equals to the liquid viscosity, and the liquid ratio of the outlet flowrate is equal to the liquid fraction.

Based on the available candidates and the number of measurements, we chose different sets containing 6 parameters. To decrease the number of possible combinations, we considered that parameters representing the same variables (e.g. top valve constants) should be estimated simultaneously for

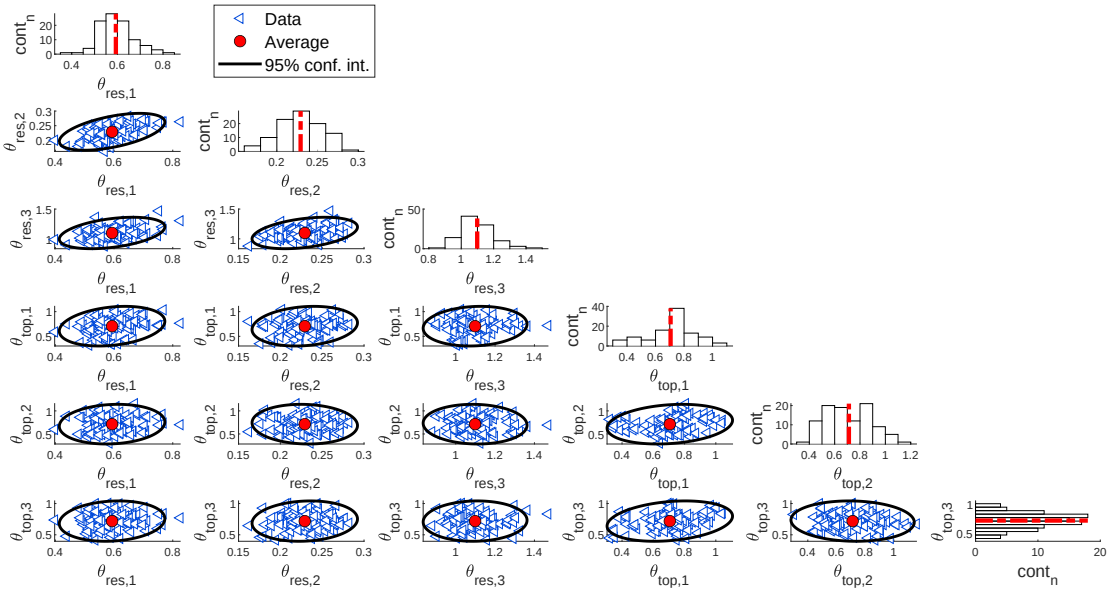


Figure 7: 100 independent steady-state model adaptation runs using historical data. The histograms of the individual parameters are plotted, with a red line indicating the mean. Also, we show the 2-dimensional distribution, analyzing two parameters at a time. In these plots we also the average (red dot) and 95% confidence interval (black line).

the three wells, i.e. if we add the valve constant for reservoir valve at well 1 ($\theta_{res,1}$) to the set, we also must include $\theta_{res,2}$ and $\theta_{res,3}$. Next, we estimated the parameters of the set 100 times based on on 100 different SS data points from our database. The sample size is arbitrarily chosen, but we consider it as a large enough sample to draw conclusions about the underlying parameter distribution. The following estimable parameter was chosen:

$$\boldsymbol{\theta} = [\theta_{res,1}, \theta_{res,2}, \theta_{res,3}, \theta_{top,1}, \theta_{top,2}, \theta_{top,3}]^T. \quad (19)$$

The distribution of these estimates of $\boldsymbol{\theta}$ is shown in Figure 7. We see that the computed confidence regions of the estimated parameters are bounded and small compared to their magnitudes. Also, none of the estimates lies on the constraints (which are not indicated in the plots). When α_l was included in $\boldsymbol{\theta}$ and we ran the same test (see Appendix C), a considerable part of the estimates of α_l was at the constraints. Although these bounds force the parameters estimates to adequate physical ranges, such a pattern would have been a clear indication of an ill-conditioned model adaptation problem and/or improper definition of the bounds. As a consequence, the adapted parameter values would lose their physical and statistical meaning and the model updating step may become useless.

Another characteristic of ill-conditioned model adaptation problems is parameter estimates with high correlations (i.e. main axes of the confidence interval ellipsoid in Figure 7 are not perpendicular to the plot axes). A correlation pattern can be noted among the reservoir valve parameters ($\theta_{res,1}$, $\theta_{res,2}$, and $\theta_{res,3}$), which is mainly caused by the large influence of the pump outlet pressure in the estimated value. Since the correlation is not significant (the inclination angle is relatively small), the parameters can be individually estimated. Usually, this is the desired case; however, it may be advantageous to pose an ill-determined estimation problem for flexibility of the estimation problem (cf. Chapter 7 in (Bard, 1974)).

Remark. *If this analysis is carried out, the tuning step of the* EKF *becomes much simpler, since a good initial guess for the estimate covariance matrices (both the model states and parameters) is available.*

6.2.2. *Tier 2: Combined observability and identifiability analysis*

After choosing $\boldsymbol{\theta}$, we need to check if the dynamic extended state (system states plus parameters) of the nonlinear model in Equation (1) is observable.

One alternative is to use results from differential geometry based as in Hermann and Krener (1977). In this test, a local nonlinear observability matrix $\mathcal{O}$, which is defined properly in the next section, is constructed using Lie Derivatives. Then, a necessary condition for local observability of a nonlinear ODE model is that the rank of $\mathcal{O}$ is equal to the states dimension. If this condition is met, it means that for every state x_0 there exists a neighbourhood where it can be distinguished from any other state x (Villaverde, 2019), i.e. it is possible to determine x_0 from output measurements in finite time. Despite the fact that the original test considers only states, it can be easily extended for parameters by considering $\boldsymbol{\theta}$ as additional states with trivial dynamics, i.e. $\dot{\boldsymbol{\theta}} = \mathbf{0}$.

Note that this approach may not be adequate if the model is complex (Alexander et al., 2020); however, in our case, it is tractable to construct $\mathcal{O}$ based on the system equations.

Testing The experimental rig model

In order to perform the analysis, we use the model of a single well. Since the models for the three wells are identi-

cal, this simplification does not affect the results. Next, we transform the DAE model described in Section 6.1.1 into an ODE model as in Equation 1. In our case this step is straightforward because the DAE system is index one. Finally, the model is rearranged in an extended form, where the system dynamics are augmented by the trivial parameter dynamics. Consequently, the parameters can be considered as pseudo-state variables and added to the state vector ($\boldsymbol{x}_e = [\boldsymbol{x}, \boldsymbol{\theta}]^T$). The model then becomes:

$$\begin{aligned} \dot{\boldsymbol{x}}_e(t) &= \boldsymbol{f}_e(\boldsymbol{x}_e(t), \boldsymbol{u}(t)) \\ \boldsymbol{y}(t) &= \boldsymbol{h}_e(\boldsymbol{x}_e(t), \boldsymbol{u}(t)) \end{aligned} \tag{20}$$

For this nonlinear system of equations, we can reconstruct the extended states $\boldsymbol{x}_e$ from the outputs y by computing the derivatives $\dot{\boldsymbol{y}}$, $\ddot{\boldsymbol{y}}$, and so on, which are performed by taking Lie derivatives of the output function $\boldsymbol{h}_e$ at a given time instant:

$$L_f \boldsymbol{h}_e(\boldsymbol{x}_e, \boldsymbol{u}) = \frac{\partial \boldsymbol{h}_e(\boldsymbol{x}_e, \boldsymbol{u})}{\partial \boldsymbol{x}_e} \boldsymbol{f}_e(\boldsymbol{x}_e, \boldsymbol{u}) \tag{21}$$

The time-dependent notation is dropped here for simplicity. The higher-order Lie derivatives can then be recursively calculated as:

$$L_f^i \boldsymbol{h}_e(\boldsymbol{x}_e, \boldsymbol{u}) = \frac{\partial L_f^{i-1} \boldsymbol{h}_e(\boldsymbol{x}_e, \boldsymbol{u})}{\partial \boldsymbol{x}_e} \boldsymbol{f}_e(\boldsymbol{x}_e, \boldsymbol{u}) \tag{22}$$

Finally, we obtain the nonlinear observability identifiability matrix by stacking $n_{x_e} - 1$ Lie derivatives, where n_{x_e} is the dimension of the extended state:

$$\mathcal{O} = \begin{bmatrix} \frac{\partial \boldsymbol{h}_e(\boldsymbol{x}_e, \boldsymbol{u})}{\partial \boldsymbol{x}_e} \\ \frac{\partial L_f \boldsymbol{h}_e(\boldsymbol{x}_e, \boldsymbol{u})}{\partial \boldsymbol{x}_e} \\ \frac{\partial L_f^2 \boldsymbol{h}_e(\boldsymbol{x}_e, \boldsymbol{u})}{\partial \boldsymbol{x}_e} \\ \vdots \\ \frac{\partial L_f^{n_{x_e}-1} \boldsymbol{h}_e(\boldsymbol{x}_e, \boldsymbol{u})}{\partial \boldsymbol{x}_e} \end{bmatrix} \tag{23}$$

If the rank($\mathcal{O}$) $= n_{x_e}$, the extended state vector can be uniquely obtained from the knowledge of the measurements. Note that, even if this *local* analysis is successful, it only indicates that two adjacent states are distinguishable (Villaverde et al., 2016); thus, distant states may be unobservable. To mitigate that, we run the analysis at several operation points, which are located along the trajectory of a previous production optimization experiment based on steady-state RTO. The results can be seen in Figure 8.

Since the dimension of the extended state for a single well is 4 ($\boldsymbol{x}_e = [m_g, m_l, \theta_{res}, \theta_{top}]^T$) at every point, the analysis indicates that the system is locally observable along the whole trajectory. Thus, we consider that the extended system is observable for practical purposes and we can proceed to the online implementation.

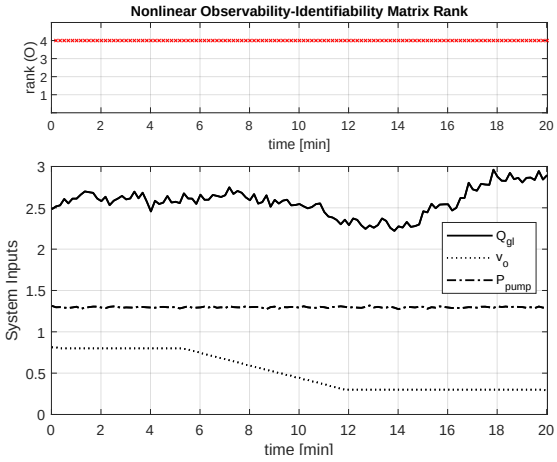


Figure 8: Nonlinear observability-identifiability matrix rank along the trajectory of a previous production optimization experiment based on steady-state RTO.

6.2.3. Data cleansing

So far, we assume a good instrumentation design and maintenance, which can provide unbiased measurements with low noise levels. However, in an industrial implementation, it is very likely that plant data is subjected to errors because of miscalibration, faulty sensors or just random events. Consequently, the collected data will not represent the underlying statistical distribution of the measurements (Albuquerque and Biegler, 1996). In this case, a data reconciliation needs to be carried out. In SSRTO, good results have been achieved when the parameter estimation algorithm is combined with data reconciliation procedures (Prata et al., 2009). For dynamic estimation, moving horizon estimators have been used as methods of state estimation, data reconciliation for nonlinear models (see, for example Rawlings and Bakshi (2006)).

6.3. ROPA execution period tuning

The final step for ROPA implementation is determining its execution period Δt_{ROPA}. For this decision, we should consider the system response time. As in any RTO implementation, we assume that there is a control layer (either advanced or PIs controllers) implementing the set points determined by ROPA. Therefore, we need to take into account the dynamic effects of the system and the controller when determining Δt_{ROPA}.

If Δt_{ROPA} is excessively small, the system becomes sensitive to rapid changes in the disturbance and noise. In this case, the inputs computed by ROPA can oscillate significantly before converging to the optimum and, in the worst-case scenario, they may not be physically realizable. On the other hand, if Δt_{ROPA} is too large (on the order of the plant's settling time), ROPA performance approaches the SSRTO performance. Based on the authors' experience, Δt_{ROPA} can be initially chosen as half of the difference between the change in the controller setpoints and the smallest plant time constant, in which only the RTO-relevant dynamics are considered.

6.3.1. Choosing Δt_{ROPA} for the rig implementation

For determining the system response time in the experimental rig, we apply a step-change in the setpoint of the gas lift flowrate controller in one of the wells and analyze the effect on the liquid flowrate. In Figure 9, we show the normalized profiles of Q_g^{SP}, Q_g and Q_l. For reference, we indicate the time of the setpoint step change (0 s); PI controller response (4 s); and system response (approx. 21 s). The results show that the difference between the changes in the setpoint and the plant response is around 20 s. Thus, according to the rule of thumb, choosing $\Delta t_{ROPA} = 10$ s is a good tuning.

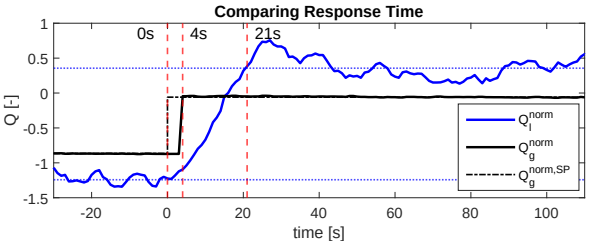


Figure 9: Step response. The initial and final mean values of Q_l are also indicated. The values were normalized for facilitating the visualization. We step the gas injection setpoint at 0 s. The control takes 4 s to track the new setpoint, while the system settles at a new steady-state after approximately 20 s.

7. Experimental results and discussion

The results comparing the three approaches are presented in Figure 10. As shown previously, we defined Δt_{ROPA} as 10 s. For comparison reasons, DRTO is also executed every Δt_{ROPA} and the steady-state detection step of SSRTO is schedule at the same rate. If ROPA and DRTO present similar results, it is an indication that there are no major advantages in optimizing the transients in the system of interest. If ROPA and SSRTO performances are the same, this means that we are waiting too long to re-optimize (i.e. Δt_{ROPA} is of the same order of magnitude than the plant settling time). In this case, we lose the potential benefits of implementing ROPA due to improper tuning.

In order to mitigate the effect of noise and unmeasured disturbances in the analysis, we run two independent experiments for each approach and present the average profiles of the computed inputs and profit. On the other hand, for the estimated parameters, we show the results of a single run. The reason is that SSRTO is triggered in different time instants during different experiments due to the SSD procedure. Therefore, showing the average at a given time instant is not relevant. We show the profiles of the estimated parameter of two independent SSRTO runs in Appendix A.

In Figures 10a and 10b, we present the values of the estimated reservoir valve and top valve coefficients. Since in ROPA and DRTO, we estimate the parameter every execution time (i.e. every 10 s), we use a continuous line. In turn, for SSRTO, we use markers to indicate not only the estimated values but also when it was executed. The approaches show a consistent θ_{res} estimation profile with low variability of both steady-state and dynamic model adaptation steps. This is a direct consequence of the proper choice of the model parametrization, which means that the parameter values are not greatly affected by measurement noise. Moreover, the reservoir parameter profiles reflect the disturbances shown in Figure 5, which is in line with the guidelines that indicate that the chosen parameters should reflect the most significant process disturbances.

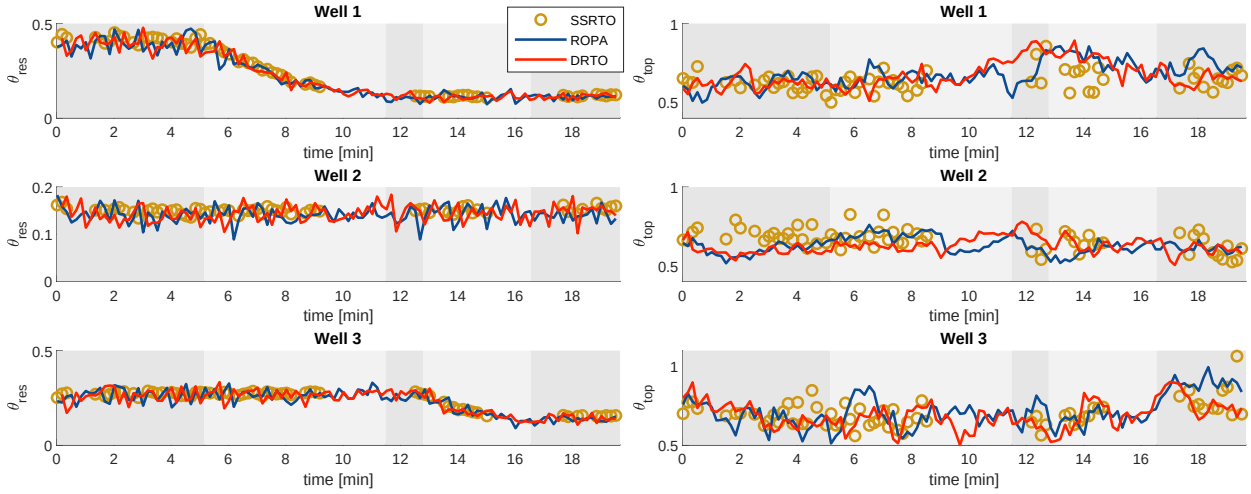
In turn, the top valve coefficient estimates oscillate more and the consistency between the steady-state and dynamic estimation is smaller. In our case, EKF was tuned based on a visual comparison between the predicted liquid flowrate and the measured one (not shown here). Since the sensitivity of the liquid flowrate Q_l prediction and the reservoir valve coefficient is larger, our tuning prioritized θ_{res} over θ_{top}. Despite the acceptable results, we believe that the tuning can still be improved to decrease the noisy θ_{top} estimates.

Figure 10c shows the average profile of the manipulated variable Q_{gl}. Similarly to the parameters results, the manipulated variables profiles are consistent among the three approaches. Moreover, the computed inputs are also consistent with the profiles inferred from engineering insight. In an exploratory experiment, we determined that lower reservoir valve openings are connected to larger input/output gains (for example, adding one unit of gas lift flowrate when $v_o = 40\,\%$ increases the liquid flowrate more than when $v_o = 80\,\%$). Therefore, between 0 min to 4 min, we expected the approaches to increase production in wells 1 and 3 due to the larger weights in the objective function. However, the fact that well 1 has a larger input/output gain should be considered. At the end of the experiment, after 17 min, all three wells have similar input/output gains. Thus, we expect the approaches to distribute the available gas more evenly among the three wells, taking only the different weights in J into account.

The profiles in Figure 10c confirm that the methods follow the expected behavior. In the beginning, well 3 is prioritised, but its value is not kept at the maximum ($Q_{g,\max} = 5\,\mathrm{sL\,min^{-1}}$). Since the first well has a larger input/output gain, the methods choose to balance the gas injection between wells 1 and 3. In turn, the gas injection in well 2 is kept at the minimum ($Q_{g,\min} = 1\,\mathrm{sL\,min^{-1}}$). At the end, the gas is more evenly distributed but still with $Q_{g,3} > Q_{g,1} > Q_{g,2}$, which follows the weights on J.

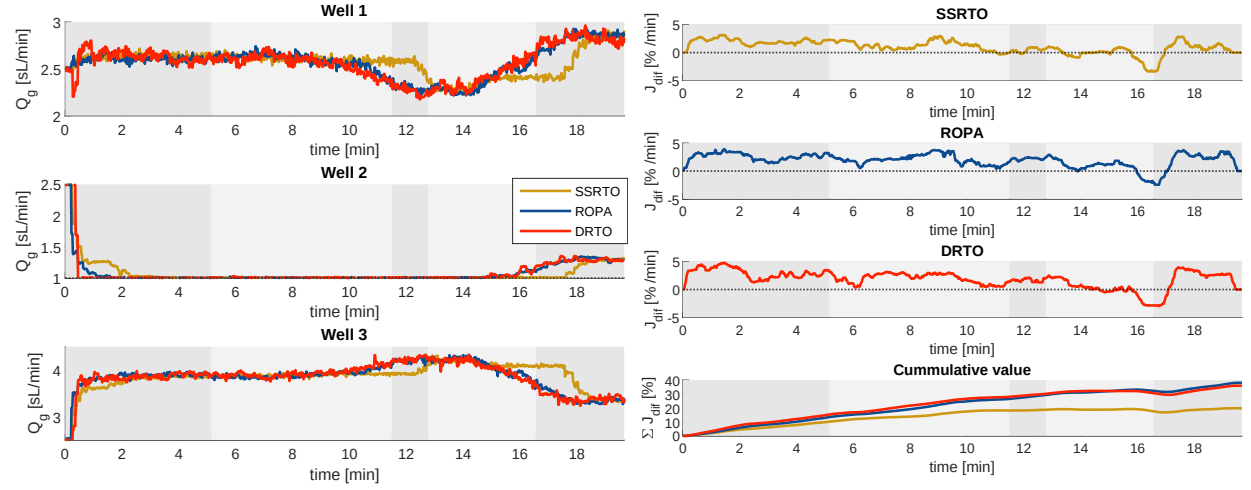
We also note that, on average, SSRTO is much slower to adapt the inputs in face of disturbances. The main changes in Q_{gl} are made only after the disturbances stops, around 12 min and 18 min. On the other hand, ROPA and DRTO quickly adapt the inputs to the disturbance. Additionally, we see that ROPA and DRTO input profiles are mostly overlapped, which indicates that there is not much advantage in optimizing the transients on this system. The main reason is that the well time constants are much faster than the disturbances time constants. This is in line with the behavior of actual subsea oil wells, where the system dynamics are mainly determined by the reservoir dynamics (Foss et al., 2018).

Figure 10d compares the profit obtained by the three ap-



(a) Estimated parameter - scaled reservoir valve coefficient.

(b) Estimated parameter - scaled top valve coefficient.



(c) Average gas lift flowrate ($1\,\mathrm{sL\,min^{-1}} \leq Q_g \leq 5\,\mathrm{sL\,min^{-1}}$). Note that the scales of the three plots are different.

(d) Average profit. The bottom figure shows the cumulative profit difference.

Figure 10: Experimental results.

proaches with the naive strategy, in which available gas lift is equally divided among the three wells (i.e. $Q_{gl,1} = Q_{gl,2} = Q_{gl,2} = 2.5\,\mathrm{sL\,min^{-1}}$). Instead of showing absolute values, we plot the difference, in percentage, between the instantaneous profit the approach of interest and fixed input approach which is calculated as $100(J - J_{fix})/J_{fix}$. In addition, we use a 60 s moving average for smoothing the profiles, because the instantaneous profit measurements are noisy.

As expected, ROPA and DRTO have a better performance than SSRTO due to the higher frequency of the production optimization execution. All the approaches present better economic results than the fixed input strategy, except around 17 min. In this case, an equal gas distribution yields better performance. However, since we apply input filters in ROPA/SSRTO and input usage regulation in DRTO, the approaches are not able to increase the inputs to this level so rapidly.

Next, we computed the total profit difference. We see ROPA and DRTO increase the obtained profit by approximately 38 % and SSRTO by 20 % when compared to the fixed nominal input approach. Such improvements represent a significant advantage of the production optimization approaches. Regarding the average instantaneous profit improvement, ROPA and DRTO are around 1.8 %, whereas SSRTO around 1 %. These values are in accordance to the profit improvement achieved in real systems by production optimization approaches (Foss et al., 2018).

Finally, we compare the three approaches in terms of computational efficiency in Figure 11, where we show the computation time distribution in one experimental run. The results show that ROPA has an average computational time approximately two times smaller than DRTO. Even though the computational times are small here, they can steeply increase with the system size and complexity. This increase will depend on particularities of the system, the model equa-

tion, the solvers, etc.

However, due to the nature of the solvers and the problems being solved, we hypothesize that in different application the relative computational time distribution will be similar to the one obtained in our validation experiments. Therefore, it can be advantageous to use ROPA in online implementations due to its lower computational time and similar economic performance. Note that the computational time of SSRTO is higher than ROPA. The main reason is the parameter estimation step. While in ROPA we use a recursive method, which is computationally cheap, in the steady-state RTO we solve an optimization problem. However, even in this case, the maximum SSRTO execution time is in the same order of magnitude that the minimum execution time of DRTO, illustrating how computationally expensive is to solve a dynamic optimization problem.

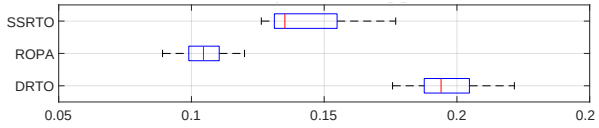


Figure 11: Distribution of the computation time of the three approaches. The average computational time values are: $\bar{\tau}_{SSRTO} = 0.1437\,\text{s}, \bar{\tau}_{ROPA} = 0.1090\,\text{s}$, $\bar{\tau}_{DRTO} = 0.2005\,\text{s}$. All computational are carried out with an Intel Core i7-8650U CPU at 2.8 GHz and 16GB RAM.

8. Conclusion

In this paper, we show the implementation of Real-time Optimization with Persistent Parameter Adaptation (ROPA) on a small pilot scale plant. ROPA is an RTO variant, in which the steady-state wait is avoided by replacing the steady-state model parameter adaptation step by a dynamic estimator.

The main contribution of this paper regards the experimental validation of this method. We implemented ROPA in a system similar to the synthetic example used in (Krishnamoorthy et al., 2018) (a gas-lift subsea oil well network). Our paper confirmed the previous in-silico findings, showing that ROPA has an economic performance similar to DRTO in the system of interest without the need of optimizing the plant transients. Therefore, ROPA becomes an interesting alternative for systems that have optimal operation around a steady-state, since it optimizes the system much more frequently than SSRTO but does not require the solution of a dynamic economic optimization like in DRTO.

As secondary contributions, we showed ROPA ability to converge to the steady-state optimum in cases without significant plant-model mismatch, and present some guidelines for its practical implementation. We believe that this latter contribution moves ROPA closer to actual implementation, increasing its potential industrial impact. Instead of emphasizing only the applied mathematics and advanced algorithms of ROPA, we used domain knowledge about the small pilot-scale plant in order to illustrate some decision that need to be made regarding modeling, model parametrization, and ROPA execution period.

References

Albuquerque, J.S., Biegler, L.T., 1996. Data reconciliation and gross-error detection for dynamic systems. AIChE Journal 42, 2841–2856. doi:10.1002/aic.690421014.

Alekman, S., 1994. Significance tests can determine steady-state with confidence. CONTROL for the Process Industries, Putman Publications, Chicago, IL 3, 62.

Alexander, R., Campani, G., Dinh, S., Lima, F., 2020. Challenges and opportunities on nonlinear state estimation of chemical and biochemical processes. Processes 8, 1462.

Amara, A.B., 2017. Gas lift: past and future, in: Society of Petroleum Engineers - SPE Middle East Artificial Lift Conference and Exhibition 2016, pp. 420–425.

Bamberger, W., Isermann, R., 1978. Adaptive on-line steady-state optimization of slow dynamic processes. Automatica 14, 223–230. doi:10.1016/0005-1098(78)90087-0.

Bard, Y., 1974. Nonlinear parameter estimation. Academic press.

Besl, H., Kossman, W., Crowe, T., Caracotsios, M., 1998. Nontraditional optimization for isom unit improves profits. Oil and Gas Journal 96.

Bhattacharya, A., Joseph, B., 1982. On-line optimization of chemical processes, in: 1982 American Control Conference, IEEE. pp. 334–337. doi:10.23919/ACC.1982.4787866.

Biegler, L.T., 2007. An overview of simultaneous strategies for dynamic optimization. Chemical Engineering and Processing: Process Intensification 46, 1043–1053.

Box, G.E.P., 1957. Evolutionary operation: A method for increasing industrial productivity. Journal of the Royal Statistical Society: Series C (Applied Statistics) 6, 81–101.

Câmara, M., Quelhas, A., Pinto, J., 2016. Performance evaluation of real industrial RTO systems. Processes 4, 44.

Cao, S., Rhinehart, R.R., 1995. An efficient method for on-line identification of steady state. Journal of Process Control 5, 363–374.

Chen, C.Y., Joseph, B., 1987. On-line optimization using a two-phase approach: An application study. Industrial and Engineering Chemistry Research 26, 1924–1930.

Curvelo, R., Delou, P.A., de Souza Jr, M.B., Secchi, A.R., 2021. Investigation of the use of transient process data for steady-state real-time optimization in presence of complex dynamics, in: Computer Aided Chemical Engineering. Elsevier. volume 50, pp. 1299–1305. doi:10.1016/B978-0-323-88506-5.50200-X.

Cutler, C., Perry, R., 1983. Real time optimization with multivariable control is required to maximize profits. Computers and Chemical Engineering 7, 663–667.

Darby, M.L., Nikolaou, M., Jones, J., Nicholson, D., 2011. RTO: An overview and assessment of current practice. Journal of Process Control 21, 874–884.

Delou, P.A., Curvelo, R., de Souza Jr., M.B., Secchi, A.R., 2021. Steady-state real-time optimization using transient measurements in the absence of a dynamic mechanistic model: A framework of hrto integrated with adaptive self-optimizing ihmpc. Journal of Process Control 106, 1–19. doi:10.1016/j.jprocont.2021.08.013.

Dirza, R., Skogestad, S., Krishnamoorthy, D., 2021. Optimal resource allocation using distributed feedback-based real-time optimization. IFAC Papers online 54, 706–711. doi:10.1016/j.ifacol.2021.08.324.

Engell, S., 2007. Feedback control for optimal process operation. Journal of Process Control 17, 203–219.

Fang, X., Shao, Z., Wang, Z., Chen, W., Wang, K., Zhang, Z., Zhou, Z., Chen, X., Qian, J., 2009. Mnemonic enhancement optimization (meo) for real-time optimization of industrial processes. Industrial and Engineering Chemistry Research 48, 499–509. doi:10.1021/ie800166p.

Findeisen, R., Allgöwer, F., 2004. Computational delay in nonlinear model predictive control. IFAC Proceedings Volumes 37, 427–432.

Forbes, J., Marlin, T.E., 1996. Design cost: a systematic approach to technology selection for model-based real-time optimization systems. Computers and Chemical Engineering 20, 717–734.

Foss, B., Knudsen, B.R., Grimstad, B., 2018. Petroleum production optimization–a static or dynamic problem? Computers and Chemical Engineering 114, 245–253.

François, G., Bonvin, D., 2013. Use of transient measurements for the optimization of steady-state performance via modifier adaptation. Industrial and Engineering Chemistry Research 53, 5148–5159.

Friedman, Y.Z., 1995. What's wrong with unit closed loop optimization? Hydrocarbon Processing 74, 107–116.

Ganesh, N., Biegler, L.T., 1987. A reduced hessian strategy for sensitivity analysis of optimal flowsheets. AIChE Journal 33, 282–296.

Gao, W., Engell, S., 2017. Dynamic mawqa: Towards efficient real-time optimization of slow dynamic processes, in: 2017 36th Chinese Control Conference (CCC), IEEE. pp. 2893–2898. doi:10.23919/ChiCC.2017.8027804.

Garcia, C.E., Morari, M., 1984. Optimal operation of integrated processing systems: Part ii: Closed-loop on-line optimizing control. AIChE Journal 30, 226–234. doi:10.1002/aic.690300209.

Golden, M.P., Ydstie, B.E., 1989. Adaptive extremum control using approximate process models. AIChE Journal 35, 1157–1169. doi:10.1002/aic.690350711.

Golshan, M., Pishvaie, M.R., Boozarjomehry, R.B., 2008. Stochastic and global real time optimization of tennessee eastman challenge problem. Engineering Applications of Artificial Intelligence 21, 215–228.

Guay, M., Dochain, D., Perrier, M., 2005. Adaptive extremum-seeking control of nonisothermal continuous stirred tank reactors. Chemical Engineering Science 60, 3671–3681. doi:10.1016/j.ces.2005.02.042.

Hermann, R., Krener, A., 1977. Nonlinear controllability and observability. IEEE Transactions on Automatic Control 22, 728–740.

Hernandez, A., 2016. Fundamentals of gas lift engineering: Well design and troubleshooting. Gulf Professional Publishing.

Jahanshahi, E., Backi, C.J., Skogestad, S., 2017. Anti-slug control based on a virtual flow measurement. Flow Measurement and Instrumentation 53, 299–307.

Kelly, J.D., Hedengren, J.D., 2013. A steady-state detection (ssd) algorithm to detect non-stationary drifts in processes. Journal of Process Control 23, 326–331.

Krishnamoorthy, D., Foss, B., Skogestad, S., 2018. Steady-state real-time optimization using transient measurements. Computers and Chemical Engineering 115, 34–45.

Krishnamoorthy, D., Ryu, J., Skogestad, S., 2019. A dynamic extremum seeking scheme applied to gas lift optimization. IFAC Papers online 52, 802–807. doi:10.1016/j.ifacol.2019.06.160.

Lee, K.S., Lee, W.K., 1985. On-line optimizing control of a nonadiabatic fixed bed reactor. AIChE Journal 31, 667–675. doi:10.1002/aic.690310417.

Ljung, L., 1999. System identification: Theory for the users. Prentice Hall.

Marchetti, A.G., François, G., Faulwasser, T., Bonvin, D., 2016. Modifier adaptation for real-time optimization—methods and applications. Processes 4, 55.

Matias, J., Jäschke, J., 2021. Online model maintenance in real-time optimization methods. Computers and Chemical Engineering 145, 107141.

Matias, J., Le Roux, G.A.C., 2018. Real-time optimization with persistent parameter adaptation using online parameter estimation. Journal of Process Control 68, 195–204.

Matias, J., Le Roux, G.A.C., 2020. Plantwide optimization via real-time optimization with persistent parameter adaptation. Journal of Process Control 92, 62–78.

Matias, J., Oliveira, J.P., Le Roux, G.A.C., Jäschke, J., 2021. Real-time optimization with persistent parameter adaptation applied to experimental rig. IFAC Papers online 54, 475–480. doi:10.1016/j.ifacol.2021.08.287.

McFarlane, R.C., Bacon, D.W., 1989. Empirical strategies for open-loop on-line optimization. Canadian Journal of Chemical Engineering 67, 665–677. doi:10.1002/cjce.5450670421.

Menezes, D.R.C., 2016. Aplicação e avaliação de desempenho de um sistema de otimização em tempo real em uma unidade de produção de propeno. Master's thesis. Universidade de São Paulo.

Prata, D.M., Schwaab, M., Lima, E.L., Pinto, J.C., 2009. Nonlinear dynamic data reconciliation and parameter estimation through particle swarm optimization: Application for an industrial polypropylene reactor. Chemical Engineering Science 64, 3953–3967. doi:10.1016/j.compchemeng.2006.05.031.

Prior, D., Lopez, S., 1999. Grangemouth ethylene plant installs closed-loop optimization solution. Oil and Gas Journal 97, 83–86.

Quelhas, A.D., de Jesus, N.J.C., Pinto, J.C., 2013. Common vulnerabilities of RTO implementations in real chemical processes. Canadian Journal of Chemical Engineering 91, 652–668.

Rao, C., Rawlings, J., 2000. Nonlinear Moving Horizon State Estimation. volume 26.

Rawlings, J.B., Bakshi, B.R., 2006. Particle filtering and moving horizon estimation. Computers and Chemical Engineering 30, 1529–1541. doi:10.1016/j.compchemeng.2006.05.031.

Rawlings, J.B., Patel, N.R., Risbeck, M.J., Maravelias, C.T., Wenzel, M.J., Turney, R.D., 2018. Economic mpc and real-time decision making with application to large-scale hvac energy systems. Computers and Chemical Engineering 114, 89–98.

Rodrigues, D., Amrhein, M., Billeter, J., Bonvin, D., 2018. Fast estimation of plant steady state for imperfectly known dynamic systems, with application to real-time optimization. Industrial and Engineering Chemistry Research 57, 3699–3716. doi:10.1021/acs.iecr.7b04631.

Rodríguez-Blanco, T., Sarabia, D., Pitarch, J.L., Prada, C., 2017. Modifier adaptation methodology based on transient and static measurements for rto to cope with structural uncertainty. Computers and Chemical Engineering 106, 480–500. doi:10.1016/j.compchemeng.2017.07.001.

Santos, J.E.W., Trierweiler, J.O., Farenzena, M., 2021. Model update based on transient measurements for model predictive control and hybrid real-time optimization. Industrial and Engineering Chemistry Research 60, 3056–3065. doi:10.1021/acs.iecr.1c00212.

Sawaragi, Y., Takamatsu, T., Fukunaga, K., Nakanishi, E., Tamura, H., 1971. Dynamic version of steady state optimizing control of a distillation column by trial method. Automatica 7, 509–516. doi:10.1016/0005-1098(71)90101-4.

Schei, T.S., 2008. On-line estimation for process control and optimization applications. Journal of Process Control 18, 821–828. doi:10.1016/j.jprocont.2008.06.014.

Schneider, R., Georgakis, C., 2013. How to not make the extended kalman filter fail. Industrial and Engineering Chemistry Research 52, 3354–3362.

Sequeira, S.E., Graells, M., Puigjaner, L., 2002. Real-time evolution for online optimization of continuous processes. Industrial and Engineering Chemistry Research 41, 1815–1825.

Shamaki, P.B., Odloak, D., 2020. Hybrid rto with zone control mpc applied to a gas-lift system, in: 2020 28th Mediterranean Conference on Control and Automation (MED), IEEE. pp. 691–696. doi:10.1109/MED48518.2020.9183175.

Srinivasan, B., Palanki, S., Bonvin, D., 2003. Dynamic optimization of batch processes: I. characterization of the nominal solution. Computers and Chemical Engineering 27, 1–26.

Valluru, J., Patwardhan, S.C., 2019. An integrated frequent rto and adaptive nonlinear mpc scheme based on simultaneous bayesian state and parameter estimation. Industrial and Engineering Chemistry Research 58, 7561–7578. doi:10.1021/acs.iecr.8b05327.

Valluru, J., Purohit, J.L., Patwardhan, S.C., Mahajani, S.M., 2015. Adaptive optimizing control of an ideal reactive distillation column. IFAC-PapersOnLine 48, 489–494. doi:10.1016/j.ifacol.2015.09.015.

Villaverde, A.F., 2019. Observability and structural identifiability of nonlinear biological systems. Complexity 2019.

Villaverde, A.F., Barreiro, A., Papachristodoulou, A., 2016. Structural identifiability of dynamic systems biology models. PLoS computational biology 12, e1005153.

Walter, E., Pronzato, L., 1997. Identification of parametric models. volume 8. Springer.

Wiggins, S., 2003. Introduction to applied nonlinear dynamical systems and chaos. volume 2. Springer Science & Business Media.

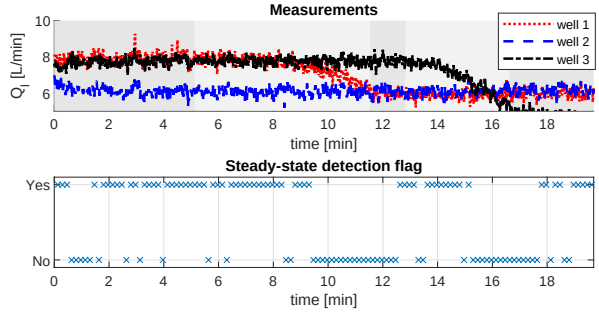
CRediT authorship contribution statement

José Matias: Conceptualization, Methodology, Coding Implementation, Formal analysis, Data Curation, Writing - Original Draft. **Julio P. C. Oliveira:** Conceptualization, Methodology, Coding Implementation, Writing - Original Draft. **Galo A.C. Le Roux:** Conceptualization, Writing - Review & Editing, Supervision, Project administration, Funding acquisition. **Johannes Jäschke:** Conceptualization, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.

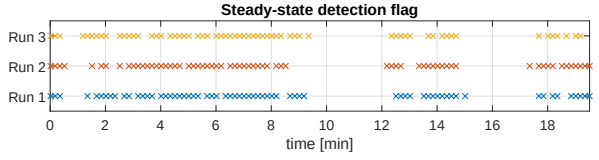
A. Extra details about SSRTO and DRTO implementation on the rig

A.1. Steady-state RTO

The first step of the SSRTO implementation is the steady-state detection (see Algorithm 2). We use the liquid flowrates $Q_{l,1}$, $Q_{l,2}$, and $Q_{l,3}$ as SS representative measurements. For performing SSD, we carry out a linear regression over a data window of $N_{SSD} = 40$ s for each of the measurements. Next, we perform a hypothesis test (with $\alpha_{SSD} = 95\%$) on the linear models' slope parameter, in which the null hypothesis is that the slope is equal to zero. If the test fails to reject the null hypothesis, the measurement is tagged as steady-state. We consider that the system is at SS if all three measurements pass the test. The results of one experimental run are shown in Figure 12a. The linear regression method presented consistent results in 3 independent runs as shown in Figure 12b.



(a) SSD results from one experimental run



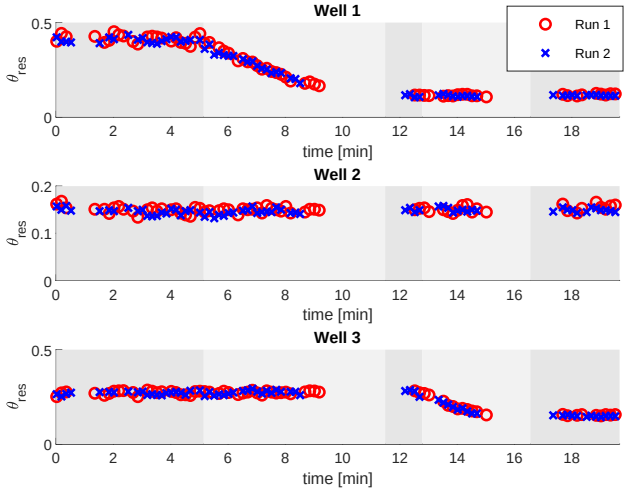
(b) Comparing the SS flag of three different runs.

Figure 12: Steady-state detection procedure analysis

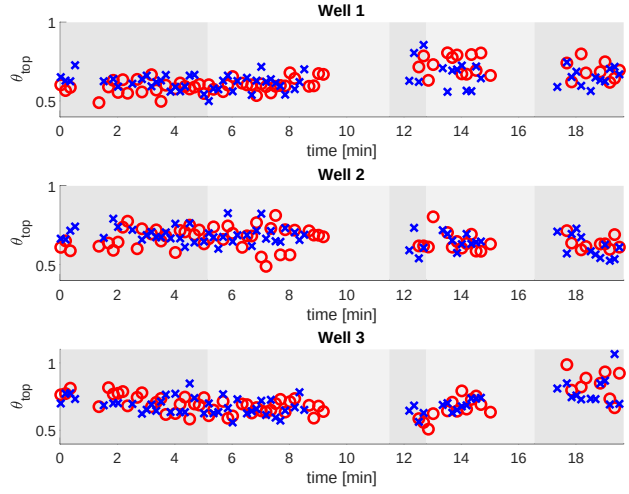
Despite not being widely used in practice (Cao and Rhinehart, 1995), this SSD procedure is easy to tune. The tuning parameter are the window length N_{SSD} and the hypothesis test power α_{SSD}. From a practical point of view, SSD procedures with fewer tuning parameters are preferable, since a poor tuning choice can significantly affect the statistical foundation of the methods and, consequently, the SSRTO results (Quelhas et al., 2013).

A.1.1. Parameter Estimates

In Figure 13, we show the value of the profiles of the estimated parameter of two independent SSRTO runs. We can see that the profiles are coherent and the variance of the estimates is relatively low. In the plot, we can also see the reason why we do not show the average values of the parameters in Figure 10. Since the steady-state RTO execution times are not the same in the two runs, using the average value for a given time instant is irrelevant.



(a) Estimated parameter - scaled reservoir valve coefficient.



(b) Estimated parameter - scaled top valve coefficient.

Figure 13: Parameter profiles for two independent SSRTO runs. Run 2 was the one shown in Figure 10

A.2. Dynamic RTO

Note that, in Equation (9), we have an input movement regulation term in the objective function, and a constraint on the input movement. By adding both terms, we guarantee smooth input profiles as seen in Figure 10c. However, two extra parameters $\Delta \boldsymbol{u}_{\max}$ and $\boldsymbol{R}$ need to be tuned. The former is easier to define given that it has physical meaning. Finding proper values of $\boldsymbol{R}$ is more challenging. Besides,

we want to assure that the chosen tuning yields a similar input usage to the other approaches, which allows us to make a fair comparison.

After a trial-and-error process, the values shown in Table 2 are used. For checking the inputs usage, we calculate the input changes ΔQ_g along one experimental run. We repeat the procedure for SSRTO and ROPA. We show the distribution of ΔQ_g for the three approaches in a box plot (Figure 14). We see that input usage of the DRTO is in a similar level similar to SSRTO and DRTO.

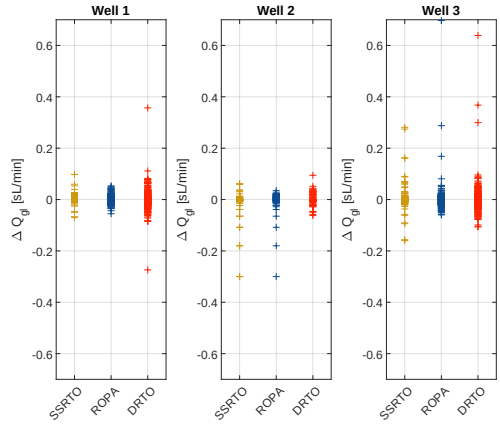


Figure 14: Comparing the input usage of the three approaches. The values indicate that both the input filter (ROPA and SSRTO) and the input regularization have similar performances.

B. Tuning parameters

Table 2 shows the tuning parameters used in the implementation of the three approaches. Refer to the codes available on our Github page[4] for a complete overview of how the methods are implemented. There, the methods are implemented in a high-fidelity dynamic model (digital twin) of the rig, where low level controller dynamics were also included and the noise levels were tuned according to the information obtained from the rig.

C. Model Parameter Estimation When Liquid Fraction is Included

In Figure 15, we illustrate the effect of choosing a different estimable parameter set $\boldsymbol{\theta}$. In this case, instead of estimating the top pressure valve coefficient, we included the liquid fraction in the pipelines, α_l, for each one of the wells in the set $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} = [\theta_{res,1}, \theta_{res,2}, \theta_{res,3}, \alpha_{l,1}, \alpha_{l,2}, \alpha_{l,3}]^T. \tag{24}$$

Consequently, we did not need the model assumption related to the liquid ratio (i.e. the outlet flowrate liquid fraction is equal to α_l. Then, we ran the same test as in Figure 7. The

[4] https://github.com/Process-Optimization-and-Control/ProductionOptRig

Table 2
Tuning parameters. I_s indicates an identity matrix of size s.

Description	Variable	Value
Experimental rig sensors sampling time	T_s	1 s
ROPA		
Execution periods	Δt_{ROPA}	10 s
ROPA Input filter gain	K_u	0.4
EKF parameters		see Codes in Github
SSRTO		
SSD execution period	Δt_{ROPA}	10 s
SSD hypothesis test power	α_{SSD}	0.9
SSD measurement window length	N_{SSD}	40 s
SSRTO Input filter gain	K_u	0.4
Model adaptation weighting matrix	V	I_6
DRTO		
DRTO sampling time	T_p	10 s
Prediction horizon	N_p	6
Input movement regulation	R	0.01
Max input change	$\Delta u_{\max}$	2 sL min^{-1}

top valve flow coefficients were set to a nominal value during the test. First, there is a clear correlation pattern between θ_{res} and α_l of the same well. Secondly, some of the estimates α_l lie on the constraints. One could potentially set new values for the upper bounds; however, this test shows that the chosen value has a large influence on the estimation of this particular parameter set. Hence, it could be misleading to to make claims about the physical meaning of α_l if the bounds are not properly defined. On the other hand, the original set led to estimates in the interior of the feasible parameter region and the bounds had no influence on them. As an extra disadvantage of this estimable parameter set, the sampling density of $\hat{\alpha}_l$ is truncated. Since it is positive on the feasible side, infinite on the constraint and zero on the unfeasible side, the computation of the covariance matrix becomes challenging (Bard, 1974).

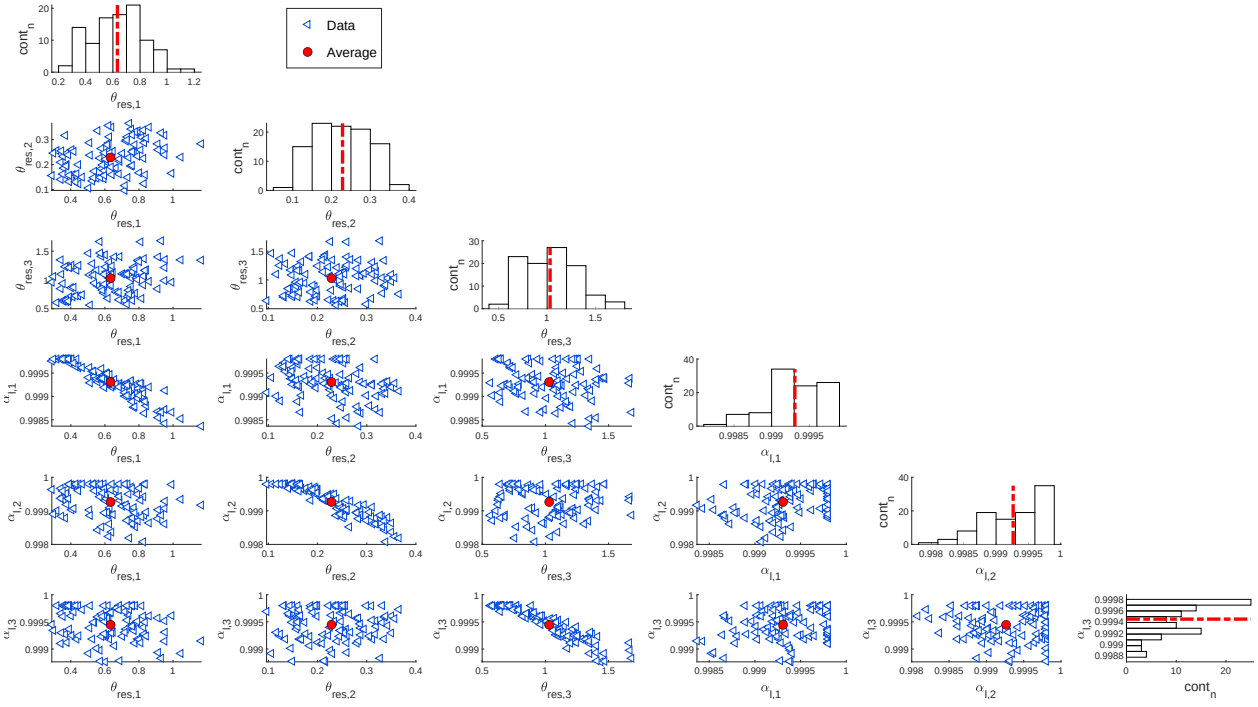


Figure 15: 100 independent steady-state model adaptation runs using historical data. The histograms of the individual parameters are plotted, with a red line indicating the mean. Also, we show the 2-dimensional distribution, analyzing two parameters at a time. In these plots we also the average (red dot).