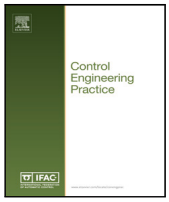
Control Engineering Practice 126 (2022) 105253

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac



Experimental validation of distributed feedback-based real-time optimization in a gas-lifted oil well rig☆

Risvan Dirza, Jose Matias, Sigurd Skogestad, Dinesh Krishnamoorthy *

Department of Chemical Engineering, Norwegian University of Science & Technology (NTNU), Trondheim, NO-7491, Norway

ARTICLE INFO

Keywords:
Production optimization
Distributed optimization
Feedback-optimizing control
Real-time optimization
Experimental validation
Oil & gas

ABSTRACT

This paper considers the problem of steady-state real-time optimization (RTO) of large-scale processes with a common constraint for several units, for example, a shared resource. Such problems are often studied under the context of distributed optimization, where each subsystem is locally optimized for a given shadow price of the shared resource. A central coordinator is then used to coordinate the allocation of the shared resource. In traditional RTO, such a framework would require repeatedly solving the subproblems and the central problems until convergence, which can be computationally expensive. To address this issue, we use a feedback-based distributed RTO scheme based on Lagrangian decomposition, where the local subproblems and the central problems are converted into feedback control problems. That is, by appropriately choosing the controlled variables in each subproblem, the overall process can be asymptotically driven to its optimal operation using feedback controllers. In this paper, we validate this approach using a lab-scale experimental rig that emulates a subsea oil production network, where the common resource is the gas lift that must be optimally allocated among the wells. We also benchmark its performance with a numerical optimization-based RTO scheme.

1. Introduction

Operating a process optimally involves taking decisions in real-time to meet production targets and ensure constraint satisfaction as the operating conditions change. This is typically done in the context of real-time optimization (RTO). The traditional RTO paradigm involves solving a numerical optimization problem using detailed process models that are updated using process measurements corresponding to steady-state operating conditions.

Many process systems involve operation of several parallel units/subsystems that are coupled by streams of energy or material flows (typically in the form shared resources) (Jose & Ungar, 2000; Martí, Navia, Sarabia, & De Prada, 2012; Stojanovski, Maxeiner, Kramer, & Engell, 2015; Wenzel et al., 2016). For example, in large industrial processes a common power plant may be used to deliver steam to different subprocesses (Jose & Ungar, 2000; Nazari, Sonntag, Stojanovski, & Engell, 2015; Stojanovski et al., 2015), or in oil production networks, oil wells from different reservoirs may share the same topside processing facility (Dirza, Skogestad, & Krishnamoorthy, 2021a; Gunnerud & Foss, 2010; Krishnamoorthy, Aguiar, Foss and Skogestad, 2018; Krishnamoorthy, Valli, & Skogestad, 2020). In such processes, it is often desirable to decompose the problem and optimize the subprocesses locally for various reasons. For example, distributed decision-making tools are often easier to implement and maintain, as opposed to a large-scale centralized optimization. They offer more flexibility and resilience, and also enable coupling with subsystems from different companies/stakeholders with limited data sharing. However, when the subsystems are coupled in one form or another, the local decision, must also be coordinated to account for the coupling. This can be done by using a centralized coordinator that ensures optimal operation of the overall process. One common framework is the Lagrangian decomposition (also known as dual decomposition), where the different subproblems are solved for a given shadow price (i.e., Lagrange multiplier corresponding to the shared resource constraint), and a central coordinator updates the shadow prices and broadcasts to the local subproblems. This results in a tatonnement process or price-based coordination in an Walrasian auction, where each subsystem adjusts its consumption of the shared resource based on its price, and the central coordinator updates the price to reach market equilibrium (Boyd, Parikh, Chu, Peleato, & Eckstein, 2011; Uzawa, 1960).

Under the traditional RTO paradigm, such a framework involves repeatedly solving the optimization subproblems and the central coordinator until convergence. This implies that at each time step, the

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

* Corresponding author.
E-mail addresses: risvan.dirza@ntnu.no (R. Dirza), jose.o.a.matias@ntnu.no (J. Matias), skoge@ntnu.no (S. Skogestad), dinesh.krishnamoorthy@ntnu.no (D. Krishnamoorthy).

https://doi.org/10.1016/j.conengprac.2022.105253
Received 11 March 2022; Received in revised form 11 June 2022; Accepted 16 June 2022
Available online 14 July 2022

subproblems are repeatedly solved several times, which can be computationally intensive, or may even be susceptible to numerical issues. It is also important to note that, although the numerical RTO problem is decomposed, it is still a "centralized" solution as seen from the process, because all the results, i.e. the setpoints from the optimization are generated at the same time, obtained upon convergence of the overall distributed RTO problem.

In order to circumvent the need for a centralized numerical solver, one can solve the decomposed optimization problem using feedback controllers in a distributed manner (Krishnamoorthy, 2021). Firstly, this may avoid some of the numerical problems just described. Secondly, the solution may be easier to implement, maintain and tune by plant personnel because it is more transparent. Lastly, and more importantly, it allows for a separation in time scale of the subproblems and central coordinator problem. That is, the manipulated variables from the subproblems can be updated on a faster time scale than that of the central coordinator problem. Furthermore, it is possible to have different tunings, closed-loop time constants, and sampling intervals for individual subproblems. This may be a significant advantage in many large-scale industrial processes, for example, because the process dynamics, measurement delays, etc., may vary for different subprocesses.

To implement the RTO problem using feedback controllers, the economic objectives must be translated into control objectives, such that the economic optimal operation is asymptotically achieved using feedback controllers. This notion of *feedback optimizing control* was first introduced by Morari, Arkun, and Stephanopoulos (1980). Here, the main objective is to find the right controlled variables, which when kept at constant setpoint, leads to economically optimal operation. Since then there has been several developments in the direction of feedback optimizing control, see Chachuat, Srinivasan, and Bonvin (2009), Engell (2007), Krishnamoorthy and Skogestad (2022) and Skogestad (2000) and the references therein. Some approaches, such as self-optimizing control, focus on how to select the controlled variables (Skogestad, 2000). Whereas other approaches, such as NCO-tracking, hill-climbing control, extremum seeking control, feedback RTO etc., are based on the idea of estimating[1] and driving the steady-state cost gradient to zero (Ariyur & Krstić, 2003; François, Srinivasan, & Bonvin, 2005; Krstić & Wang, 2000; Tsamardinos, Brown, & Aliferis, 2006).

Most of these approaches only considers the unconstrained optimization problem. That is, they assume that the different active constraint regions are identified a priori, and in each active constraint region, the active constraints are tightly controlled, and only consider the reduced unconstrained optimization problem (which we categorize as *region-based methods* Krishnamoorthy & Skogestad, 2022). The changes in the active constraint regions are typically handled by classical advanced control tools such as selectors, split-range etc.

An alternative approach to region-based approach (which is also the focus of this paper), is to transform the constrained economic optimization problem into an unconstrained optimization problem using Lagrangian relaxation. Economically optimal operation is then asymptotically reached by controlling the constraints using the Lagrange multipliers (dual variables) in the slow time scale, and the gradient of the Lagrangian is controlled to zero using the physical manipulated variables (i.e., the primal decision variables) in a cascade fashion. This approach is also known as *primal–dual feedback optimizing control*. See Krishnamoorthy and Skogestad (2022) for more detailed discussion on the different approaches to feedback optimizing control.

The decomposed version of primal–dual feedback optimizing control was developed for a distributed optimization problem by Krishnamoorthy (2021), where the dual decomposition method was converted to a distributed feedback control problem by relaxing the share resource constraints. Here, the central coordinator problem updates the Lagrange multipliers for the shared resource constraints in the slow time scale, and the local subproblems control the gradient of the Lagrangian for the given shadow prices on a faster time scale. This distributed feedback-based optimization problem with linear coupling constraints was also analytically shown to converge to the stationary point of the overall optimization problem under reasonable assumptions by Krishnamoorthy (2021). This approach was then applied in a simulation study to a large-scale subsea production system with both linear and nonlinear coupling constraints by Dirza et al. (2021a), where it was shown that the distributed feedback-based RTO (DFRTO) approach was able to drive the system to its overall optimal operation without the need to repeatedly solve numerical optimization problems online.

Building on our previous work (Dirza et al., 2021a; Krishnamoorthy, 2021), we now experimentally validate this approach on a lab-scale gas-lifted oil well rig consisting of three wells with the lift gas being the shared resource that couples the three wells together (Matias, Oliveira, Roux, & Jaschke, 2021). To this end, the main contribution of this paper is to experimentally validate the distributed feedback-based RTO (DFRTO) scheme (Dirza et al., 2021a; Krishnamoorthy, 2021) and benchmark its performance with a numerical optimization based RTO. The reminder of the paper is organized as follows. Section 2 describes the primal–dual and the distributed feedback-based RTO scheme. Section 3 describes the experimental facility and the implementation of DFRTO, and Section 4 presents the results from the experimental facility.

2. Problem formulation for DFRTO scheme

Consider a steady-state optimization problem

$$\min_{\mathbf{u}} \quad J(\mathbf{u},\mathbf{p}) \tag{1a}$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{u},\mathbf{p}) \le 0, \tag{1b}$$

$$\mathbf{u} \in \mathcal{U} \tag{1c}$$

where $\mathbf{u} := [u_1, \dots, u_N]$ are the set of manipulated variables, $\mathbf{p} \in \mathbb{R}^{n_p}$ denotes the set of parameters/disturbances, $J : \mathcal{U} \times \mathbb{R}^{n_p} \to \mathbb{R}$ is the cost function, $\mathbf{g} : \mathcal{U} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_g}$ denotes the constraints.

Introducing the Lagrange function $\mathcal{L}(\lambda, \mathbf{u}, \mathbf{p}) = J(\mathbf{u},\mathbf{p}) + \lambda^\top \mathbf{g}(\mathbf{u},\mathbf{p})$ the necessary conditions for optimality (KKT conditions) for the problem (1) can be expressed as

$$\nabla_{\mathbf{u}} J(\mathbf{u},\mathbf{p}) + \nabla_{\mathbf{u}} \mathbf{g}(\mathbf{u},\mathbf{p})^\top \lambda = \mathbf{0} \tag{2a}$$

$$\mathbf{g}(\mathbf{u},\mathbf{p}) \le \mathbf{0} \tag{2b}$$

$$\lambda \ge \mathbf{0} \tag{2c}$$

$$\lambda_i g_i(\mathbf{u},\mathbf{p}) = 0, \ \forall i = 1, \dots, n_g \tag{2d}$$

The unknown variables in equation set (2) are $\mathbf{u}$ and λ. The equation set can be solved using dual ascent (Boyd et al., 2011). Here we solve (2a) with respect to $\mathbf{u}$ with a fixed value of λ, and then iteratively change λ in an outer loop to satisfy the remaining equations, where the most important is to keep $\mathbf{g} = \mathbf{0}$ for the case when the constraints are active, which corresponds to a nonzero λ.

Since the constraint value $\mathbf{g}$ is often measured, this provides an excellent opportunity to use feedback control to solve the equations. For example, we may use an I-controller, (or a PI-controller for faster convergence). The use of feedback control for solving (2) with respect to λ has the additional advantage that we do not actually use the model for the constraints $\mathbf{g}$ in (2b) and (2d), which means that we do not need to update this part of the model. It is also possible to use feedback control to solve Eq. (2a) with respect to $\mathbf{u}$, but this does not come with the additional feedback advantages because the cost and constraints gradients are not measured.

Therefore, we can reach stationarity condition at steady-state by controlling $c(\lambda) := \nabla_{\mathbf{u}} J(\mathbf{u},\mathbf{p}) + \nabla_{\mathbf{u}} \mathbf{g}(\mathbf{u},\mathbf{p})^\top \lambda$ to a constant setpoint of

[1] These methods essentially differ on how the steady-state cost gradient is estimated.

$c^{sp} = \mathbf{0}$ for any given λ. The fast time-scale gradient controllers (also known as primal controllers) are responsible for this task.

In an outer loop, the Lagrange multipliers λ_i are then used as manipulated variables to control their corresponding constraints $g_i(\mathbf{u}, \mathbf{p})$ to its limit of $\mathbf{0}$ for all $i = 1, \dots, n_g$. By pairing λ_i to $g_i(\mathbf{u}, \mathbf{p})$ and using a max selector, the conditions that $\lambda \geq \mathbf{0}$ (2c) as well as the complementary slackness condition (2d) are satisfied. The slow time-scale centralized constraint controllers (also known as dual controllers) are responsible for this task. Therefore, the primal and dual controllers together satisfy the necessary conditions of optimality (2) at steady-state.

Considering only integral action, the primal–dual feedback law in this case can be expressed in continuous time as

$$\begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} K_I & 0 \\ 0 & \boldsymbol{\alpha} \end{bmatrix} \begin{bmatrix} \nabla_{\mathbf{u}} J(\mathbf{u}, \mathbf{p}) + \nabla_{\mathbf{u}} \mathbf{g}(\mathbf{u}, \mathbf{p})^{\top} \lambda \\ \mathbf{g}(\mathbf{u}, \mathbf{p}) \end{bmatrix} \tag{3}$$

$$\lambda = \max(0, \lambda) \tag{4}$$

where K_I and $\boldsymbol{\alpha}$ are the gain matrix for gradient controllers and central constraint controllers, respectively. This is done in a cascade fashion as shown in Fig. 1. The discrete representation of the primal–dual feedback law shown in (3)–(4) is as follows.

$$\mathbf{u}^{k+1} = \mathbf{u}^k + K_I \left[\nabla_{\mathbf{u}} J(\mathbf{u}^k, \mathbf{p}^k) + \nabla_{\mathbf{u}} \mathbf{g}(\mathbf{u}^k, \mathbf{p}^k)^{\top} \lambda^k \right] \tag{5}$$

$$\lambda^{k+1} = \max \left[\mathbf{0}, \lambda^k + \boldsymbol{\alpha} \mathbf{g}(\mathbf{u}^k, \mathbf{p}^k) \right] \tag{6}$$

where k is the discrete sample time, which in this case is assumed to be the same for all the controllers, although in general, one can implement the different controllers with different sampling times.

Now consider the case of a large scale process with N subsystems with a shared resource constraint that couples the different subsystems together. In this case, the optimization problem (1) has a special structure,

$$J(\mathbf{u}, \mathbf{p}) := \sum_{i=1}^{N} f_i(u_i, p_i) \tag{7a}$$

$$\mathbf{g}(\mathbf{u}, \mathbf{p}) := \sum_{i=1}^{N} g_i(u_i, p_i) - g^{max} \tag{7b}$$

where f_i is the local cost of the subsystem, $\mathbf{u} = [u_1, \dots, u_N]$, $\mathbf{p} = [p_1, \dots, p_N]$ with u_i and p_i being the decision variables and the parameters in the local subsystem, $\mathcal{U} := \mathcal{U}_1 \times \cdots \times \mathcal{U}_N$, and g^{max} is the limit of the constraint.

In this case, the controlled variable

$$c(\lambda) := \sum_{i=1}^{N} \nabla_{u_i} f_i(u_i, p_i) + \lambda^{\top} \sum_{i=1}^{N} \nabla_{u_i} g_i(u_i, p_i)$$

is additively separable. Therefore, we can easily decompose this and in each subsystem control the following

$$c_i(\lambda) := \nabla_{u_i} f_i(u_i, p_i) + \lambda^{\top} \nabla_{u_i} g_i(u_i, p_i)$$

to a setpoint of $c_i^{sp} = 0$ by manipulating the primal variable u_i,

$$\dot{u}_i = K_{I,i} c_i(\lambda) \tag{8}$$

where $K_{I,i}$ is the integral gain.

The Lagrange multiplier λ is then updated to control the constraints $\mathbf{g}(\mathbf{u}, \mathbf{p})$ to its limit of $\mathbf{0}$ with a max selector, i.e., the central coordinator is given by

$$\dot{\lambda} = \boldsymbol{\alpha} \sum_{i=1}^{N} g_i(u_i, p_i) \tag{9a}$$

$$\lambda = \max(0, \lambda) \tag{9b}$$

where $\boldsymbol{\alpha}$ is the integral gain.

To this end, in each subsystem we control $c_i(\lambda)$ in the fast time scale, and in the slow time scale we update the Lagrange multipliers λ to control the coupling constraint $\mathbf{g}(\mathbf{u}, \mathbf{p})$. Assuming that the stationary point is also the local minimum, this leads to optimal operation of the local subsystem for a given Lagrange multiplier λ. As central constraint controller (9) updates λ, this leads to optimal performance of the overall optimization problem (1).

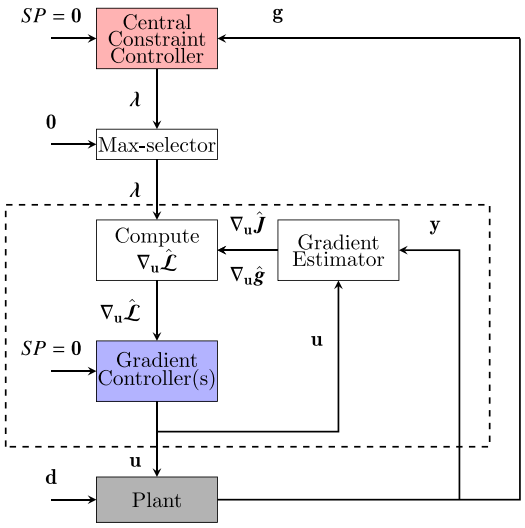


Fig. 1. Block diagram of primal–dual DFRTO scheme. The gray box represents a given plant. The white boxes represent computational blocks. The red and blue boxes represent controller blocks with different timescales. If the objective function (1a) is additively separable, one can decompose problem (the inner loop control structure located inside dashed black lines) into N subproblems. A more detailed and case-oriented control structure is shown in Section 3.3. The symbol of hat $\hat{(\cdot)}$ represents estimated values, and $\mathbf{y}$ denotes the measurement set (output variables). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Note that the local controlled variables $c_i(\lambda)$ in each subsystem requires local cost and constraint gradient information. The gradient of the Lagrange function $\nabla_{\mathbf{u}} \mathcal{L}(\mathbf{u}, \mathbf{p}, \lambda)$ is not a directly measured variable. Therefore, it is necessary to utilize the measurement set to estimate this gradient. See Krishnamoorthy and Skogestad (2022) and Srinivasan, François, and Bonvin (2011) for several available gradient estimation techniques that could be used with this approach. In this work, we use the forward sensitivity analysis to estimate the gradient (see Appendix A for the detailed description)

Remark 1. It is important to emphasize that Primal–Dual DFRTO scheme mainly consists of central constraint controllers (or dual controllers) and gradient controllers (or primal controllers), which can be implemented using simple tools such as PID controllers. Therefore, this framework can achieve optimal performance asymptotically without any numerical optimization solver.

Remark 2. It is interesting to note that the estimation of the local steady-state cost and constraint gradients in each subsystem needs only the models and the real-time process measurements of the local subsystem. This is a useful property that enables implementation of DFRTO in systems where privacy and data sharing issues are important as indicated by Krishnamoorthy (2021) and Wenzel et al. (2016) for example.

3. Experimental setup

In subsea production systems, wells are located on the seabed to extract the hydrocarbons trapped in the underground reservoir. The produced oil and gas is transported along the seabed in pipelines to the processing facility where the riser pipeline takes it from the seabed to the surface. If the reservoir pressure is low, either naturally or

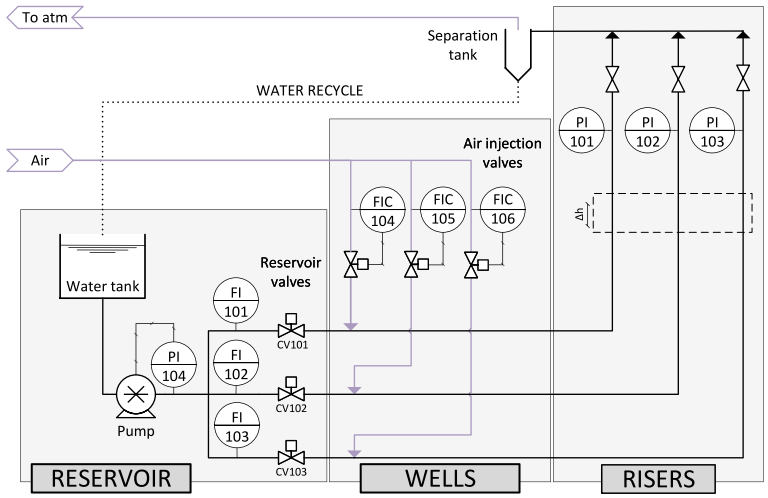


Fig. 2. Experiment schematic. Adapted from Matias et al. (2021). The system measurements $\mathbf{y_p}$ are the well top pressures (PI101, PI102 and PI103), the pump outlet pressure (PI104), the liquid flowrates (FI101, FI102, and FI103), and the gas flowrates (FI104, FI105, and FI106). Three PI controllers are used for controlling the gas flowrates, $\mathbf{u} = \begin{bmatrix} Q_{gl,1} & Q_{gl,2} & Q_{gl,3} \end{bmatrix}^\top$, to the calculate setpoints, $\mathbf{u}^{sp} = \begin{bmatrix} Q^{sp}_{gl,1} & Q^{sp}_{gl,2} & Q^{sp}_{gl,3} \end{bmatrix}^\top$. 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.

due to depletion, artificial lift methods may be needed to overcome the pressure losses and bring the hydrocarbons to the surface. Gas-lift is a commonly used artificial lift method, in which compressed gases are injected into the well tubing to reduce the fluid mixture density and, consequently, the hydrostatic pressure losses. However, injecting too much gas increases the frictional pressure drop in the well tubing, which has a counteracting effect (Aamo, Eikrem, Siahaan, & Foss, 2005). Therefore, each well has a local optimum corresponding to the gas lift injection rate. The total available lift gas is often a limited resource that must be optimally allocated among the wells to maximize the production from the overall production network. For more practical information of production optimization in oil and gas industry, the reader is referred to Bieker, Slupphaug, and Johansen (2007) and Dirza, Skogestad, and Krishnamoorthy (2021b) and the references therein.

3.1. Experimental rig as a subsea production system

To emulate the subsea gas-lifted oil production system, we use a lab-scale experimental rig that uses water and air as working fluids instead of oil and gas for simplification. The choice of working fluids does not influence the gas lift phenomenon, which one can still see in the lab rig. Thus, the rig is suited for studying production optimization methods, where the gas lift effect is the phenomenon of interest. Fig. 2 shows a simplified flowsheet of the system indicating three sections: a reservoir, well, and riser section.

The reservoir section contains a stainless steel tank, a centrifugal pump, and the three control valves (CV101, CV102, and CV103). These valves are used to represent disturbances from reservoir (for example, to emulate pressure oscillations, or reservoir depletion). With this setup, the reservoir produces only liquid with outflow rates ranging from 2 L/min to 15 L/min. Flow meters (FI101, FI102, and FI103) are located before the reservoir valves to measure the outflow rates. The pump's outlet pressure (PI104) is kept constant at 0.3 barg in this experiment using a PI controller that adjusts the pump rotation.

The wells consist of three parallel flexible hoses with 2 cm inner diameters and length of 1.5 m. Approximately 10 cm after the reservoir valves, pressurized air at approximately 1 barg is injected by three air flow controllers (FIC104, FIC105, and FIC106) within the range of 1 sL/min to 5 sL/min.

The risers are 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, three manual valves are kept open during the experiments. The air is vented out to the atmosphere, and the liquid is recirculated to the reservoir water tank. More detailed description of the test setup can be found in Matias et al. (2021). The pictures of this rig showing the three sections can be found in Appendix D.

3.2. Optimization problem setup

The objective of the optimization problem in this experimental setup is to maximize the network liquid flow rate (i.e. the summation of the liquid production of the three wells) given a limited amount of gas-lift injection. Considering problem (1) with a special structure (7), we can express the economic objectives as follows,

$$\begin{aligned} J(\mathbf{u},\mathbf{p}) &:= \sum_{i=1}^{3} f_i(u_i,p_i) \\ &= -20Q_{l,1}(u_1,p_1) - 25Q_{l,2}(u_2,p_2) - 30Q_{l,3}(u_3,p_3) \end{aligned} \tag{10}$$

where $Q_{l,1}$, $Q_{l,2}$, and $Q_{l,3}$ are the produced liquid flowrates of wells 1, 2, and 3, respectively. For illustration, we assume that the wells have different hydrocarbon prices as shown above. The input vector is given by

$$\mathbf{u} = \begin{bmatrix} Q_{gl,1} & Q_{gl,2} & Q_{gl,3} \end{bmatrix}^\top$$

where $Q_{gl,1}$, $Q_{gl,2}$, and $Q_{gl,3}$ are the injected gas flowrates of wells 1, 2, and 3, respectively.

In the context of optimization, these flowrates are the decision variables. Meanwhile, for the plant, these flowrates are the setpoints that need to be tracked. As shown in Fig. 2, the experimental lab rig has flow indicator and controllers (FICs) 104, 105, and 106 to regulate the air injection flowrates to their setpoints. Therefore, we denote these decision variables of the optimization problem as $\mathbf{u}^{sp} = \begin{bmatrix} Q^{sp}_{gl,1} & Q^{sp}_{gl,2} & Q^{sp}_{gl,3} \end{bmatrix}^\top$. One may consider the valve opening of FICs as the decision variables. However, this alternative may face some practical issues due to the non-linearity and the hysteresis behavior of the valves. Furthermore, three elements of $\mathbf{p}$, which are the reservoir valve openings CV101, CV102, and CV103, are time-varying. It implies that the cost is also a function of $\mathbf{p}$.

The total gas availability, which is a shared (input) constraint, can also be expressed using the structure shown in (7), where:

$$\begin{aligned} g(\mathbf{u},\mathbf{p}) &:= \sum_{i=1}^{3} g_i(u_i,p_i) - g^{max} \\ &= Q_{gl,1} + Q_{gl,2} + Q_{gl,3} - Q^{max}_{gl} \end{aligned} \tag{11}$$

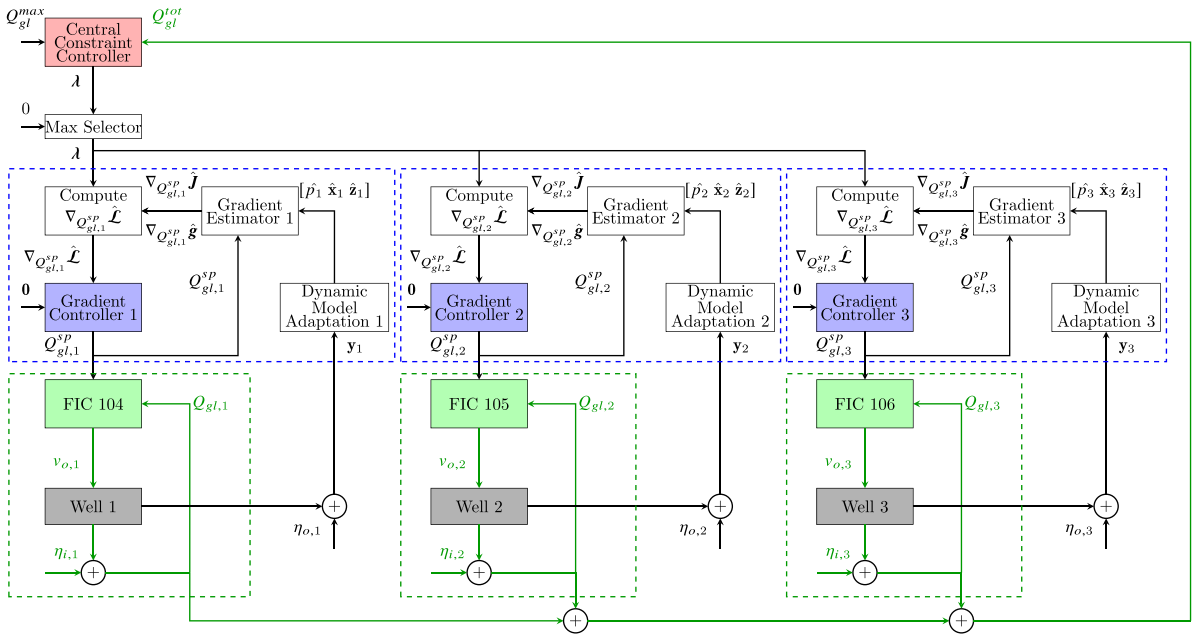


Fig. 3. DFRTO control structures implemented in the experimental lab rig. The diagram has three subproblems because the economic objective (10) is additively separable. The plant defined in Fig. 2 consists of well 1, well 2, well 3, FIC 104, FIC 105 and FIC 106, which are inside dashed green lines in this diagram. The actual gas-lift flowrates $(Q_{gl,i})$ are states in the plant. The decision variables for DFRTO scheme are the gas-lift flowrates setpoints $\left(Q^{sp}_{gl,i}\right)$. Using the flow controllers, the real manipulated variables for the well rig are the valve openings, labeled by $v_{o,i}$. The well rig has both input and output measurement noises labeled by $\eta_{i,i}$ and $\eta_{o,i}$, respectively. Using local measurement set $(\mathbf{y}_i)$, and the local dynamic model adaptation, we estimate differential states $(\hat{\mathbf{x}}_i)$, algebraic states $(\hat{\mathbf{z}}_i)$, and parameters/disturbances, $(\hat{p}_i)$. In this work, the "Dynamic Model Adaptation" is an extended Kalman filter (see Appendix A). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where we directly measure the constraint, and we use FICs to drive $Q_{gl,i}$ to $Q^{sp}_{gl,i}$.

Remark 3. Note that the total gas availability in general is an inequality constraint. However, in this experimental setup, we found that the gas lift constraint is always active at the optimum, which is also common in many gas-lifted oil fields. Hence we can equivalently consider the coupling constraint to be an equality constraint, and then we do not need the max selector (9b) for λ update in the constraint control.

3.3. Distributed feedback RTO setup

Control structure design. Based on the problem formulation from Section 2, we now implement the distributed feedback-based RTO (DFRTO) structure for our experimental setup. Given that we have three wells in the experimental setup, we decompose the problem into three subsystems. For each subsystem, we use a local gradient estimator that estimates the gradient of the local cost $\nabla Q_{l,i}$ and the constraint $\nabla Q_{gl,i}$ (cf. Remark 1). Each subsystem has a local PI controllers that controls $c_i(\lambda)$ to 0 (gradient controller). The output from the gradient controller, denoted by $Q^{sp}_{gl,i}$ is then given as setpoint to the flow controller FIC$_i$ that manipulates the air injection valve opening to achieve these setpoints. The control structure implemented in the lab-rig is illustrated in Fig. 3.

For the central constraint controller, we can directly measure the constraint $\mathbf{g}$ value for updating the Lagrange multiplier. However, we have to estimate both cost and constraint gradient for the gradient controllers (i.e primal controllers) as shown in Eq. (3). Since a reliable dynamic model of the system is available, we use forward sensitivity analysis to estimate these values (see Appendix A). To compute the local sensitivities, we also need to estimate the current states of the system (both differential and algebraic). Here, an extended Kalman filter (EKF) is used in each subsystem to estimate the states using only the local measurements. However, any suitable dynamic estimator can be applied, as long as it provides accurate estimates of the states, filtering the measurement noise properly.

Controller tuning. Once the control structure is defined, we need to tune the controllers. In the experimental rig, the fastest possible sampling rate of the data acquisition software is 1 s. In theory, we could execute both the gradient and the central constraint controllers at the same rate. However, depending on their tuning, they might compete against each other, which might drive the system to instability. Therefore, we need a time scale separation between these controllers. Given the cascaded structure, and need for proper timescale separation for smooth operation, we now provide in depth discussion on how the controllers were tuned.

The idea of dual decomposition is that the subproblems (represented by the dotted boxes in Figs. 1 and 3) are solved for each update (iteration) of the central coordinator problem (represented by the central constraint controller). However, the subproblems controlling $c_i(\lambda)$ to a setpoint of c_i^{sp} are also solved by iteration, so in practice the subproblems will not reach full convergence to their setpoint of c_i^{sp} within each central coordinator problem update. Fortunately, it is possible to estimate the approach to convergence when solving the equations using feedback controllers, as in this paper.

The rate of convergence to the setpoint in each subproblem is given by the closed-loop time constant τ_{c_i} of the corresponding control loop. More specifically, for a linear first-order system, the approach to convergence (or steady state) is $(1 - e^{-t/\tau_{c_i}})$ where τ_{c_i} is the closed-loop time constant of the ith control loop, and t is the convergence time of the central coordinator problem (9). Thus, the approach to convergence increases from 63% to 95.0% to 99.3% as t/τ_{c_i} increases from 1 to 3 to 5. Thus, at 5 time constants the approach is 99.3%, and convergence (or steady state) has for practical purposes been reached. This may be regarded as the basis for the rule of thumb of requiring a time scale separation between control layers of at least 5 (Skogestad & Postlethwaite, 2005). If the time scale separation gets too small, typically 3 or less, the layers will start interacting and we may experience undesired oscillatory behavior or even instability (Baldea & Daoutidis, 2007). A larger value (larger than 5) allows for robustness to process gain variations which will affect the closed-loop time constants of the control loops. However, with a too large value, the overall

convergence (including the central constraint controller) will be slow, so for practical purposes, a value for the time scale separation of 5 to 10 is often recommended.

The limiting case of infinite time scale separation corresponds to $\epsilon = \tau_{c_i}/t \to 0$, which is the singular perturbation condition in the mathematical literature. Note that a time scale separation between 5 and 10, corresponds to ϵ between 0.2 and 0.1.

Remark 4. Recently, it was shown analytically by Krishnamoorthy (2021) that the distributed feedback RTO framework is guaranteed to converge to the stationary point (2a) of the overall optimization problem under the assumption of perfect control of the subproblems. This assumption is also satisfied by using a timescale separation between 5 and 10.

In summary, the constraint should be controlled in a slow timescale ($\tau_{\lambda,c}$), and the gradient in a fast timescale ($\tau_{\mathbf{u},c}$), where we typically select the ratio to be 5 to 10. In this paper, we use integral controllers that are tuned using SIMC-rules (Skogestad, 2003). For the central constraint controller the integral gain is given by

$$\boldsymbol{\alpha} = \frac{1}{K_\lambda \left(\tau_{\lambda,c} + \theta_\lambda\right)} \tag{12}$$

where K_λ and θ_λ are the step response and the time delay of the constraint by the dual variable (Lagrange multiplier), and $\tau_{\lambda,c}$ is the time scale that governs the evolution of $\mathbf{g}$. For the three local gradient controllers, the integral gain is given by

$$K_{I,i} = \frac{1}{K_{\mathbf{u}_i}\left(\tau_{\mathbf{u}_i,c} + \theta_{\mathbf{u}_i}\right)} \tag{13}$$

where $i = 1, 2, 3$ is the well index, $K_{\mathbf{u}_i}$ and $\theta_{\mathbf{u}_i}$ are the step response and the time delay of the gradient by the primal variable (Decision variables/inputs), and $\tau_{\mathbf{u}_i,c}$ is the time scale that governs the evolution of $c_i(\lambda)$.

To determine K_λ, θ_λ, $K_{\mathbf{u}_i}$, and $\theta_{\mathbf{u}_i}$, we analyze the step responses. Meanwhile, $\tau_{\mathbf{u}_i,c}$ and $\tau_{\lambda,c}$ are the tuning parameters that should carefully consider the concept of time scale separation, where $\epsilon_i = \frac{\tau_{\mathbf{u}_i,c}}{\tau_{\lambda,c}}$ should be less than 1, which implies that the outer loop has slower time scale than the inner one. We can ideally choose $\tau_{u_i,c} = 1$ since want to drive the inner loop to the steady-state as fast as possible. However, it may be too aggressive. Therefore, we adjust them (i.e., the controller's parameter tuning) based on our practical justification and observation.

Besides avoiding "too aggressive" controllers, we also consider the fact that local gradient controllers are not the lowest in the hierarchy (see Fig. 3), and thus the timescale of the gradient controllers should be slower than the plant (that contains the FICs and the lab rig).

The type of PID Controller of FICs is designed by the manufacture, where the valve drive is calculated based on the following PID controller equation.

$$\delta v_{o,i}^k = I_{FIC,i}^{k-1} + I_{FIC,i}^k - \frac{K_{P,FIC,i}}{65.536} Q_{gl,i}^k \tag{14a}$$

$$I_{FIC,i}^k = \frac{K_{I,FIC,i}}{65.536}\left(Q_{gl,i}^{sp} - Q_{gl,i}^k\right) \tag{14b}$$

where $I_{FIC,i}$ is the integral value, $K_{P,FIC,i}$ is the proportional gain, $K_{I,FIC,i}$ is the integral gain, and $\delta v_{o,i}$ is the valve drive. This controller contributes in creating plant dynamic in which the time constant may vary between 5–10 s.

Table 1 shows the controller and tuning parameters that we obtain. A more detailed Table showing the step response parameters can be seen in Appendix C. Note that the largest ϵ_i is 0.2113, indicating that the time-scale ratio of the overall experimental oil rig system is still within the acceptable condition of the time-scale separation concept.

Table 1
Controller and Tuning parameters.

Description	Variable	Value
Experimental rig sensors sampling time	T_s	1 s
HRTO		
Execution periods	Δt_{HRTO}	10 s
HRTO Input filter gain	K_u	0.4
EKF parameters	See codes in Github	
DFRTO		
Execution periods	Δt_{PD}	2 s
Central constraint controller step length	α	0.0117
Gradient controller Input 1 gain	$K_{I,1}$	0.0769
Gradient controller Input 2 gain	$K_{I,2}$	0.0444
Gradient controller Input 3 gain	$K_{I,3}$	0.0893
FIC 104 Proportional gain	$K_{P,FIC,1}$	8560
FIC 105 Proportional gain	$K_{P,FIC,2}$	8560
FIC 106 Proportional gain	$K_{P,FIC,3}$	8560
FIC 104 Integral gain	$K_{I,FIC,1}$	100
FIC 105 Integral gain	$K_{I,FIC,2}$	100
FIC 106 Integral gain	$K_{I,FIC,3}$	50

Controller tuning validation. Before implementing the D-FRTO in the experimental rig, we first validated the controller tunings in a lab rig model developed in MATLAB. The model is a high-fidelity dynamic model of the rig, that includes the lower layer controller dynamics, i.e. FICs, input and output noise. The noise was tuned according to the information obtained from the rig. This modeling structure implies that both the lower layer controller dynamics and the noise are part of the plant as presented by the diagram block shown in Figs. 1 and 3. The reader is referred to the code available on our Github page[2] for detailed parameters. In addition, Matias et al. (2021) contains a detailed description of the model.

Remark 5. Note that the simulator model of the test rig is only used to determine the controller tuning parameters, before it is implemented on the actual rig. All results that are presented later in Section 4 are from the actual experimental rig, and not the simulator.

Benchmark method. To benchmark the performance of the distributed feedback-based RTO approach, this paper considers the Hybrid Real-time Optimization (HRTO)[3] that solves the centralized numerical optimization problem. We choose the HRTO approach as our benchmark since this circumvents the steady-state wait time issue in traditional steady-state RTO (see Krishnamoorthy, Foss and Skogestad (2018) and Matias and Le Roux (2018)). Simply put, HRTO continuously estimates the parameters using dynamic models and transient measurements, e.g. by applying an extended Kalman filter. Then, the corresponding updated steady-state model is used for computing the solution of the economic optimization problem. In our experimental setup, we use the exact same state and parameter estimator for the HRTO and our proposed DFRTO scheme. The optimal setpoints $\mathbf{u}^{sp}$ computed by the HRTO layer are given to the flow controllers FIC. We consider HRTO with the execution period of 10 s in this experiment. Fig. 4 illustrates this method.

To summarize, the HRTO and DFRTO differ in the fact that in HRTO, a steady-state optimization problem is solved to determine the optimal setpoint u^{sp}, whereas in DFRTO the optimal setpoints u^{sp} are given by the feedback controllers. Thus, HRTO is a good benchmark for DFRTO.

4. Experimental results and discussions

Utilizing the control and tuning parameters shown in Table 1, we implemented our proposed DFRTO and the HRTO to serve as

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

[3] HRTO is the same as RTO with persistent parameter adaptation (ROPA) (Matias & Le Roux, 2018).

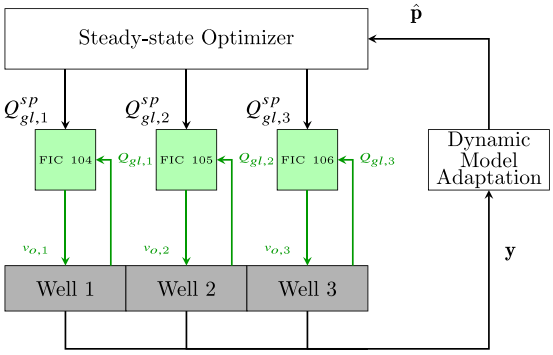


Fig. 4. Block diagram of Hybrid RTO (HRTO). The gray box represents a given plant. The white boxes represent computational blocks. In this work, the "Dynamic Model Adaptation" is an Extended Kalman Filter (see Appendix A).

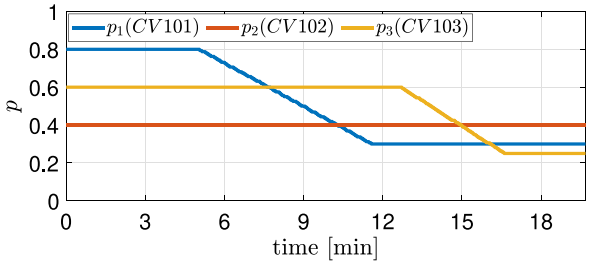


Fig. 5. The change of reservoir valve openings (CV101, CV102, and CV103) during the experiments for representing different reservoir behaviors. These reservoir valve openings are system disturbances.

benchmark. Fig. 5 shows the reservoir valve openings ($CV101$, $CV102$, $CV103$) that we consider as the disturbance in this experiment. The first disturbance occurs when the opening of $CV101$ gradually decreases from $t = 5$ to $t = 11$ min. We expect a decrease in the gas-lift injection in well 1, and a redirection of the gas supply to the other wells. The second disturbance occurs when the opening of $CV103$ also gradually decreases from $t = 13$ to $t = 16.5$ min. We expect that the gas supply to well 3 reduces with larger rate since the "hydrocarbon price" of this well is higher. Meanwhile the other wells will obtain more gas supply with larger rate as well. We try to avoid sudden disturbance to ensure that the controller can adjust the plant smoothly.

In the rig, we used a programming environment (LABVIEW Bitter, Mohiuddin, & Nawrocki, 2006) to automate the implementation of these disturbance. Therefore, it is possible to repeat the independent experiments with the same disturbance profile.

Comparison of the optimal setpoints. We first run the experimental results comparing DFRTO and HRTO, which are presented in Figs. 6–8. Fig. 7 depicts the actual gas-lift flow rate. The measured trajectories are slightly different from the calculated input setpoint shown in Fig. 6. This difference occurs due to input measurement noise and the fact that the gas flowrates controllers (FIC 104, FIC 105, FIC 106, see Fig. 2) need time to settle the actual gas-lift flow rate $\mathbf{u}$ to the setpoint of gas-lift flow rate $\mathbf{u}^{sp}$. Sometimes the setpoint change calculated by HRTO is quite significant such that the gas flowrates controllers results in input spikes (see around $t = 11$ min in Fig. 7). The number of these spikes is reduced because we have implemented first order input filter in the setpoints computed by HRTO, i.e.:

$$\mathbf{u}^{sp,k+1} = \mathbf{u}^{sp,k} + K_u \left(\mathbf{u}^{sp,\star,k} - \mathbf{u}^{sp,k}\right)$$

where $\mathbf{u}^{sp,\star}$ is the optimal setpoint given by HRTO solver. Meanwhile, DFRTO does not have the input filter because it has gradient controller(s) as input filter.

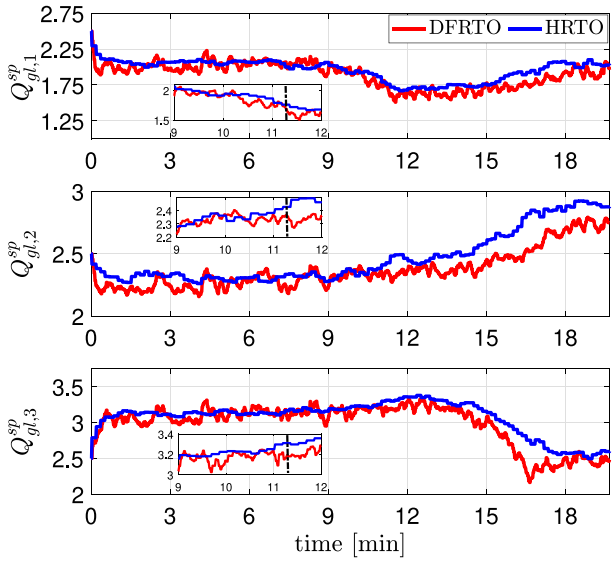


Fig. 6. The gas-lift flow rate setpoint $\left(\mathbf{u}^{sp} = \mathbf{Q}^{sp}_{gl}\right)$ of every wells due to reservoir parameter changing (disturbance) from the experimental lab rig. Each plot has a magnifying plot in the time window 9 to 12 min. These plots show that when disturbance p_1 start settling down at around 11 h, the calculated input setpoints $\mathbf{u}^{sp}$ seem slightly off initially.

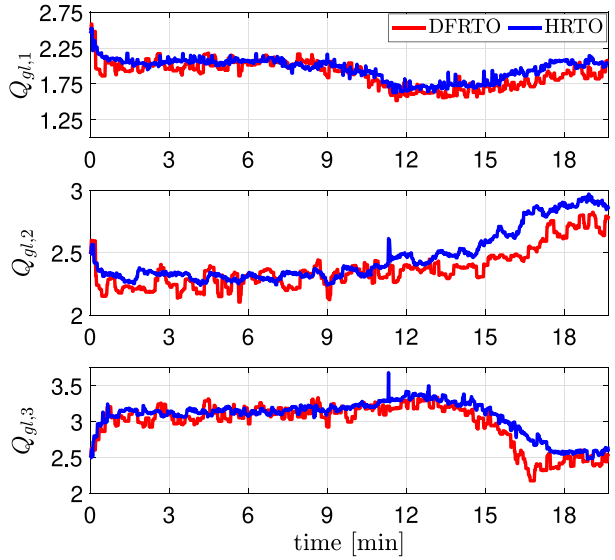


Fig. 7. The measured gas-lift flow rate ($\mathbf{u}$) of every wells due to reservoir parameter changing (disturbance) from the experimental lab rig.

Figs. 6–7 show that DFRTO is slower in responding to the second disturbance (see around $t = 15$ min). This slow response is the consequence of considering time-scale separation concept explained in Section 3.3.

Fig. 8 shows the constraint satisfaction and its associated Lagrange multiplier, where the initial guess of the Lagrange multiplier is slightly off in DFRTO. The Lagrange multiplier λ of the DFRTO is around 9.5 at $t = 0$ min. Meanwhile, the mean value of Lagrange multiplier λ of HRTO is around 9.1 at that time. However, DFRTO slowly drives the Lagrange multiplier to converge to a better value that is closer to HRTO solver obtains. We can observe the effect of the more accurate λ estimate on the performance of DFRTO, since, around $t = 6$ min, the active constraint is slightly better controlled. In any case the difference here is not significant, and the variations are mainly due to measurement noise.

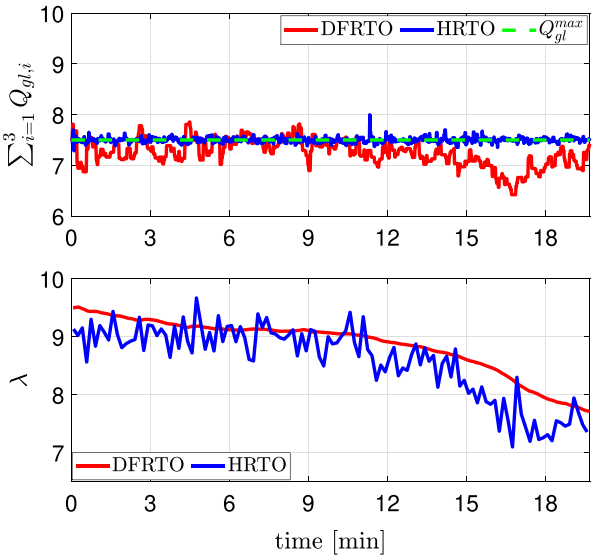


Fig. 8. Constraint satisfaction and Lagrange multiplier evolution due to reservoir parameter changing (disturbance) from the experimental lab rig.

From around $t = 12$ min, DFRTO converges slower than HRTO following the second disturbance. This is mainly due to the tuning of the constraint controller, where it can be seen in Fig. 8 that the λ converges slower than the HRTO. As λ converges, the DFRTO converges to the same value as HRTO at around $t = 18$ min, confirming that DFRTO is able to provide the same asymptotic performance as HRTO. This behavior is expected as we carefully design the central constraint controller. By considering the timescale separation concept, we avoid undesired behaviors, such as oscillatory behavior. On the other hand, the constraint control performance becomes relatively slow, and thus any dynamic constraint violation may last quite some time. Since we expressed the total gas capacity constraint as an equality constraint (see Eq. (11)), the dynamic constraint violation also includes the situation when the total of gas-lift flowrates is less than Q_{gl}^{max}.

Due to various process and measurement noise, the actual system is noisy. However, it can be seen that the general process trend clearly reaches steady-state. Due to this condition, tightly controlling the hard constraints maybe a challenging task for DFRTO (which may need a back-off depending on the noise levels). Nevertheless, DFRTO still drives the system to the optimal steady-state value.

In addition, the control structure of DFRTO allows the possibility to have other sources of error such as additional control dynamic from central constraint coordinator, gradient estimation error, transmission error, and measurement error. Any error (due to improper tuning or design) from one of them leads to the additional disturbance for constraint satisfaction.

Surprisingly, although the Lagrange multiplier calculated by HRTO is sometimes slightly off and less smooth than DFRTO, HRTO has a relatively good constraint satisfaction. This condition happens because here the constraints are on the inputs. Therefore, even a high degree of plant-model mismatch does not influence the constraint satisfaction performance. In other words, the constraint model ends up with a simple summation of a 'known' input. Therefore, HRTO may have a better constraint satisfaction performance in handling an input constraint in this specific case.

To summarize, by observing Figs. 6–9, we can conclude that the solutions of the two compared approaches, i.e., DFRTO and HRTO are similar in this experiment run.

To ensure reproducibility, we then re-run the experiments once again with the same set of disturbance profile for the different approaches. The average results from the two independent experiment runs are shown in Appendix B. This confirms that DFRTO can provide very similar performance as HRTO, without the need for numerical solvers.

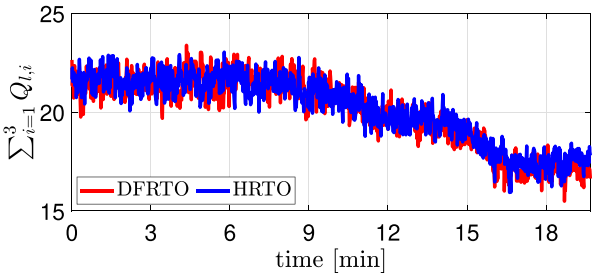


Fig. 9. Total production comparison due to reservoir parameter changing (disturbance) from the experimental lab rig.

Comparison of the optimal cost. To analyze the optimization performance of DFRTO and HRTO, we compare the profit (from both individual experiments) obtained by the two approaches with the naive approach, where we consider fixed inputs, i.e.,

$$\mathbf{u} = \begin{bmatrix} Q_{gl,1}^{sp} & Q_{gl,2}^{sp} & Q_{gl,3}^{sp} \end{bmatrix}^{\top} = \begin{bmatrix} \frac{Q_{gl}^{max}}{3} & \frac{Q_{gl}^{max}}{3} & \frac{Q_{gl}^{max}}{3} \end{bmatrix}^{\top}$$

The naive approach illustrates the case in which no information about the system is available. Hence, the best alternative is to divide the available gas equally among the wells. This latter approach is another benchmark to show that the optimization methods, i.e, DFRTO and HRTO, will give more profit compared to not doing any optimization at all.

To measure the performance, we plot the difference, in percentage, between the instantaneous profit of the approach of interest (i.e. DFRTO or HRTO) and the naive approach. The difference is calculated as

$$J_{diff} = \frac{J - J_{naive}}{J_{naive}}.100 \tag{15}$$

where J is the profit of the approach of interest, and J_{naive} is the profit of the naive approach. In addition, we use a 60 s moving average for smoothing the profiles, because the instantaneous profit measurements are noisy.

Fig. 10 shows that DFRTO and HRTO are more profitable than the naive strategy. Although both the approaches converge to the same optimal steady-state solution, DFRTO had slower transients (especially around t =15 min) due to the chosen controller tuning parameters. This resulted in a slightly smaller cumulative profit $\sum J_{diff}$ compared to HRTO.

5. Conclusion

In this work, we have done experiments to validate the DFRTO method. Based on the experiments we can conclude that

- DFRTO is able to provide the same asymptotic optimal performance as HRTO. The transient behavior is slightly different which is affected by the choice of the controller tuning parameters.
- In the DFRTO framework, it is necessary to consider the timescale separation between the gradient and constraint controllers. If the central constraint controller is tuned to be in the same timescale as the gradient controllers, then it can lead to instability or oscillatory behavior. However, if the central constraint controller is tuned to be too slow, then the convergence to the optimal steady-state can be too slow. This was also seen in the presented experiment results, where DFRTO converges slightly slower than the HRTO following some disturbances.

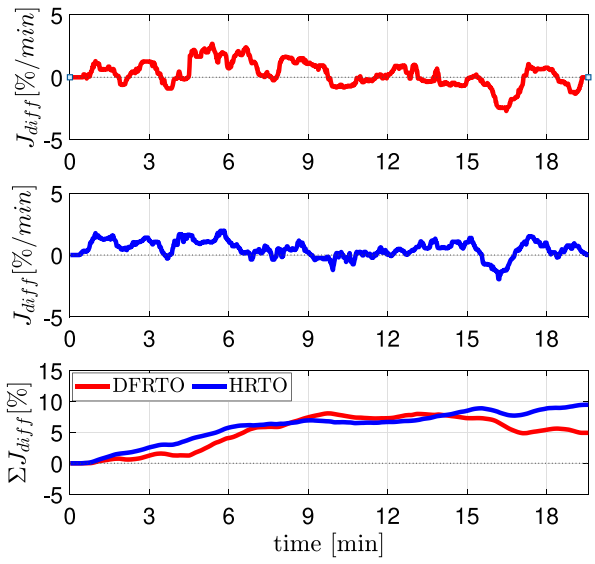


Fig. 10. The average profit from the experimental lab rig. The bottom subplot shows the cumulative average profit.

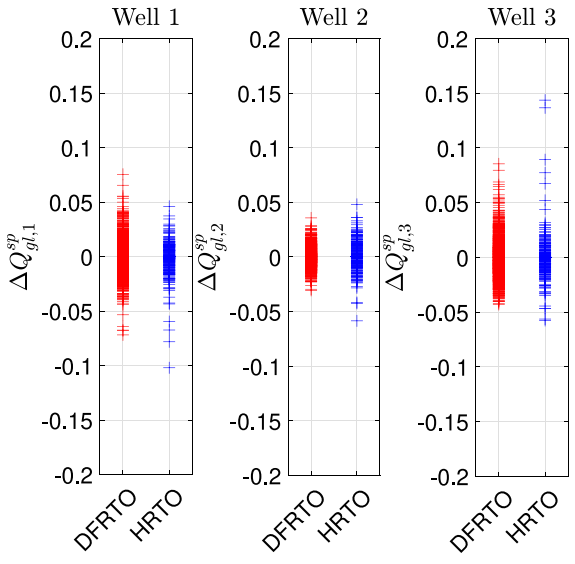


Fig. B.11. The comparison of $\Delta Q^{sp}_{gl,i} = Q^{sp,k}_{gl,i} - Q^{sp,k-1}_{gl,i}$ of each well due to reservoir parameter changing (disturbance) from the experimental lab rig.

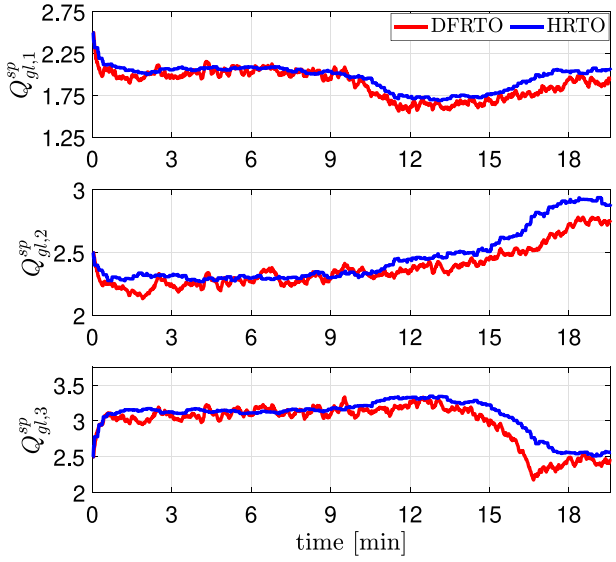


Fig. B.12. The comparison of average inputs setpoint trajectories from the experimental lab rig.

As future work, we suggest the following,

- In this experiment, the shared resource constraint was on the inputs, leading to linear coupling constraints. Since the experimental rig does not have topside separation of liquid and gas, we are not able to include the total produced gas capacity constraint, as done in Dirza et al. (2021a). Validating the DFRTO approach for nonlinear coupling constraints such as in Dirza et al. (2021a) could be a valuable future research.
- We also used integral controllers in the DFRTO framework for its simplicity. Another future research direction could be to consider more advanced controllers for the gradient and constraint controllers in the DFRTO framework.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Steady-state gradient estimation

This paper uses forward sensitivity analysis to estimate the gradient. This gradient estimation has two main steps. First, we use the current plant information to update the state and parameters of the model using a dynamic adaptation scheme (here, extended Kalman filter). Next, we use the updated model to compute the steady-state gradients via the forward sensitivity analysis.

Remark 6. The system has a differential–algebraic model, where $\mathbf{x} \in \mathbb{R}^{n_\mathbf{x}}$ is a vector of differential states, $\mathbf{z} \in \mathbb{R}^{n_\mathbf{z}}$ is a vector of algebraic states, and $\mathbf{y} \in \mathbb{R}^{n_\mathbf{y}}$ is a vector of output (measured states). This model is necessary for state and parameter estimation, and not explicitly shown in problem (1).

A.1. Extended Kalman filter

In order to use the Kalman filter equations, we first linearize the available model. Note that the model is a differential–algebraic equation (DAE) system; however, since it is an index-1 model, it can be easily re-arranged into an ordinary differential equation (ODE). Additionally, we assume that the unknown parameters are time-varying. Their dynamics are determined by a random walk model:

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{v}_k \tag{A.1}$$

where $\mathbf{v}^k$ follows a normal distribution with mean zero and covariance V_θ. Additionally, we assume that the increments $\mathbf{v}^k$ are independent of $\mathbf{v}^{\neq k}$.

Then, we combine the system dynamics and parameter dynamics to obtain an extended model that is used for parameter estimation. Since the model was linearized, we can apply extended Kalman filter equations for estimating $\mathbf{x}^k$, $\mathbf{z}^k$, and $\mathbf{p}^k$ simultaneously. For a complete derivation of the EKF equations, please refer to Walter and Pronzato (1997).

Table C.2
Controller and Tuning parameters.

Description	Variable	Value
Experimental rig sensors sampling time	T_s	1 s
HRTO		
Execution periods	Δt_{HRTO}	10 s
HRTO Input filter gain	K_u	0.4
EKF parameters	See codes in Github	
DFRTO		
Execution periods	Δt_{PD}	2 s
Central constraint controller step response	K_λ	1.2060
Central constraint controller time delay	θ_λ	0
Central constraint controller time scale	$\tau_{\lambda,c}$	71
Central constraint controller step length	α	0.0117
Gradient controller Input 1 step response	K_{u_1}	1.2
Gradient controller Input 1 time delay	θ_{u_1}	0
Gradient controller Input 1 time scale	$\tau_{u_1,c}$	15
Gradient controller Input 1 gain	$K_{I,1}$	0.0769
Gradient controller Input 2 step response	K_{u_2}	1.5
Gradient controller Input 2 time delay	θ_{u_2}	0
Gradient controller Input 2 time scale	$\tau_{u_2,c}$	15
Gradient controller Input 2 gain	$K_{I,2}$	0.0444
Gradient controller Input 3 step response	K_{u_3}	3.4
Gradient controller Input 3 time delay	θ_{u_3}	0
Gradient controller Input 3 time scale	$\tau_{u_3,c}$	15
Gradient controller Input 3 gain	$K_{I,3}$	0.0893
FIC 104 Proportional gain	$K_{P,FIC,1}$	8560
FIC 105 Proportional gain	$K_{P,FIC,2}$	8560
FIC 106 Proportional gain	$K_{P,FIC,3}$	8560
FIC 104 Integral gain	$K_{I,FIC,1}$	100
FIC 105 Integral gain	$K_{I,FIC,2}$	100
FIC 106 Integral gain	$K_{I,FIC,3}$	50

A.2. Forward sensitivity analysis

The original nonlinear DAE model is in the form:

$$\begin{aligned} \mathbf{x}^{k+1} &= \check{F}(\mathbf{x}^k, \mathbf{z}^k, \mathbf{u}^k; \mathbf{p}^k) \\ 0 &= \check{G}(\mathbf{x}^k, \mathbf{z}^k, \mathbf{u}^k; \mathbf{p}^k) \end{aligned} \tag{A.2}$$

The steady-state gradients are estimated using the stationary value of forward sensitivity equations:

$$\begin{aligned} 0 &= \frac{\partial \check{F}^\top}{\partial \mathbf{x}} S_{SS} + \frac{\partial \check{F}^\top}{\partial \mathbf{z}} R_{SS} + \frac{\partial \check{F}^\top}{\partial \mathbf{u}} \\ 0 &= \frac{\partial \check{G}^\top}{\partial \mathbf{x}} S_{SS} + \frac{\partial \check{G}^\top}{\partial \mathbf{z}} R_{SS} + \frac{\partial \check{G}^\top}{\partial \mathbf{u}} \end{aligned} \tag{A.3}$$

where S_{SS} and R_{SS} are the sensitivities of the differential states $\mathbf{x}$ and algebraic states $\mathbf{z}$ w.r.t. the inputs $\mathbf{u}$.

Since, in our specific case, the objective J and constraint function $\mathbf{g}$ are linear functions of the algebraic states ($J = \check{H}_J z$, $\mathbf{g} = \check{H}_\mathbf{g} z$), we use the chain rule to obtain $\nabla_\mathbf{u} J$ and $\nabla_\mathbf{u} \mathbf{g}$:

$$\begin{aligned} J = \check{H}_J z \implies \nabla_\mathbf{u} J = \check{H}_J R_{ss} \\ \mathbf{g} = \check{H}_\mathbf{g} z \implies \nabla_\mathbf{u} \mathbf{g} = \check{H}_\mathbf{g} R_{ss} \end{aligned} \tag{A.4}$$

Appendix B. DFRTO vs. HRTO: Average values

Fig. B.11 compares the input setpoint rate $\Delta Q^{sp}_{gl,i} = Q^{sp,k}_{gl,i} - Q^{sp,k-1}_{gl,i}$ of the implemented approaches from two independent experiments.

Fig. B.11 also shows that the input rate setpoints of DFRTO are, in general, comparable to HRTO. However, we can still note that for well 1 and 3, HRTO has significant outliers (−0.1 for well 1 and around 0.15 for well 3). These outliers might occur due to a numerical optimizer with an imperfect numerical condition or a bad parameter estimation. In an extreme case, the HRTO may suffer from numerical robustness issues.

Fig. B.12 shows that the average inputs setpoint trajectory resulting from both approaches are similar over time. Based on these results, we can conclude that DFRTO has a similar performance to HRTO in general, which also supports the same observation stated in the previous discussion in Section 4.

Fig. B.12 also shows that in the time window $t = 8$ to $t = 14$ min, the trajectories of average input solutions of DFRTO seem mild.

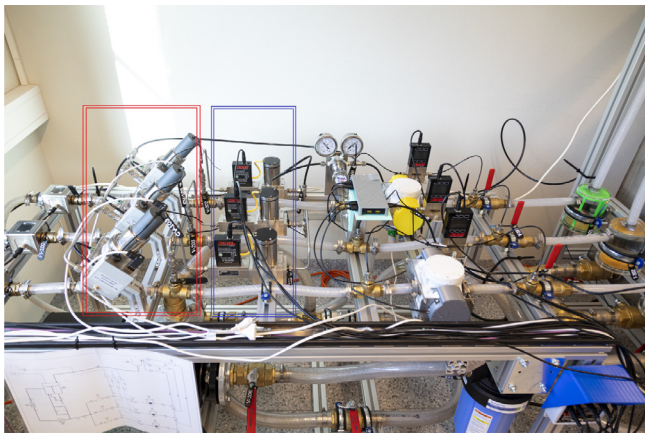


Fig. D.13. The equipment inside the red box are Reservoir valves, and inside the blue box are the Injection valves. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

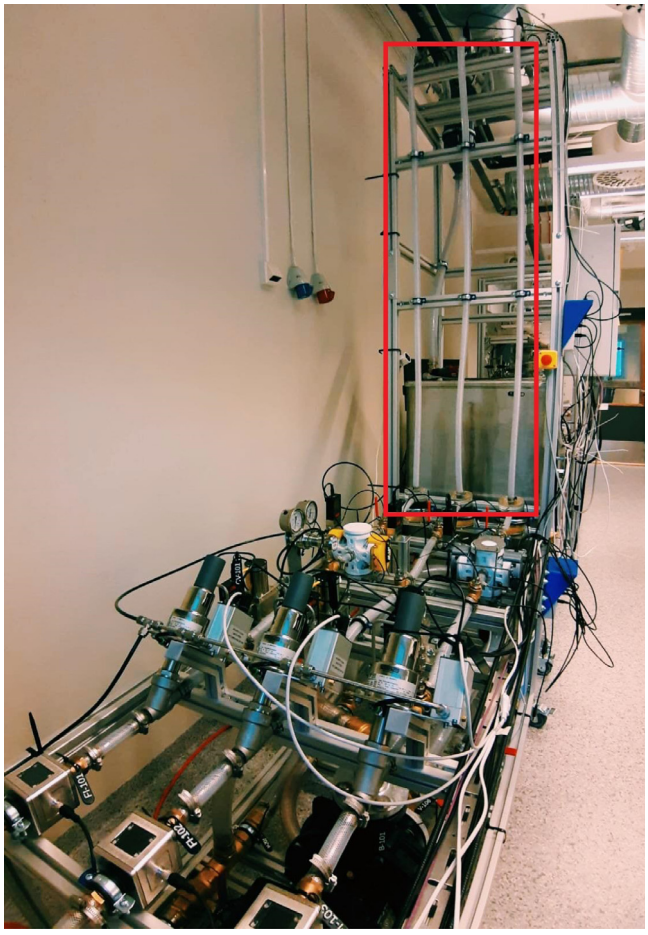


Fig. D.14. The equipment inside the red box are the risers. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Both approaches have quite similar cumulative profits. However, after $t = 14$ min, the input trajectories need to be more aggressive to achieve optimal performance. This condition seems to create challenges for DFRTO as it may take more time to obtain the optimal performance.

Appendix C. Controller and tuning parameters

Table C.2 shows the controller and tuning parameters we used in this work.

Appendix D. The experimental lab rig

The experimental lab rig we use in this work are shown in Figs. D.13 and D.14.

References

Aamo, O., Eikrem, G., Siahaan, H., & Foss, B. (2005). Observer design for multiphase flow in vertical pipes with gas-lift—-theory and experiments. *Journal of Process Control*, *15*(3), 247–257. http://dx.doi.org/10.1016/j.jprocont.2004.07.002.

Ariyur, K. B., & Krstić, M. (2003). *Real-time optimization by extremum-seeking control*. Hoboken, NJ, USA: John Wiley & Sons, Inc., http://dx.doi.org/10.1002/0471669784.

Baldea, M., & Daoutidis, P. (2007). Control of integrated process networks—A multi-time scale perspective. *Computers & Chemical Engineering*, *31*(5), 426–444.

Bieker, H. P., Slupphaug, O., & Johansen, T. A. (2007). Real-time production optimization of oil and gas production systems: A technology survey. *SPE Production & Operations*, *22*(04), 382–391. http://dx.doi.org/10.2118/99446-PA.

Bitter, R., Mohiuddin, T., & Nawrocki, M. (2006). *LabVIEW: Advanced programming techniques*. Crc Press.

Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, *3*(1), 1–122. http://dx.doi.org/10.1561/2200000016.

Chachuat, B., Srinivasan, B., & Bonvin, D. (2009). Adaptation strategies for real-time optimization. *Computers & Chemical Engineering*, *33*(10), 1557–1567. http://dx.doi.org/10.1016/j.compchemeng.2009.04.014.

Dirza, R., Skogestad, S., & Krishnamoorthy, D. (2021a). Optimal resource allocation using distributed feedback-based real-time optimization. *IFAC-PapersOnLine*, *54*(3), 706–711. http://dx.doi.org/10.1016/j.ifacol.2021.08.324.

Dirza, R., Skogestad, S., & Krishnamoorthy, D. (2021b). Real-time optimal resource allocation and constraint negotiation applied to a subsea oil production network. In *2021 annual technical conference and exhibition (ATCE)*. SPE, http://dx.doi.org/10.2118/206102-MS.

Engell, S. (2007). Feedback control for optimal process operation. *Journal of Process Control*, *17*(3), 203–219. http://dx.doi.org/10.1016/j.jprocont.2006.10.011.

François, G., Srinivasan, B., & Bonvin, D. (2005). Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty. *Journal of Process Control*, *15*(6), 701–712. http://dx.doi.org/10.1016/j.jprocont.2004.11.006.

Gunnerud, V., & Foss, B. (2010). Oil production optimization—A piecewise linear model, solved with two decomposition strategies. *Computers & Chemical Engineering*, *34*(11), 1803–1812. http://dx.doi.org/10.1016/j.compchemeng.2009.10.019.

Jose, R. A., & Ungar, L. H. (2000). Pricing interprocess streams using slack auctions. *AIChE Journal*, *46*(3), 575–587. http://dx.doi.org/10.1002/aic.690460316.

Krishnamoorthy, D. (2021). A distributed feedback-based online process optimization framework for optimal resource sharing. *Journal of Process Control*, *97*, 72–83. http://dx.doi.org/10.1016/j.jprocont.2020.11.006.

Krishnamoorthy, D., Aguiar, M. A., Foss, B., & Skogestad, S. (2018). A distributed optimization strategy for large scale oil and gas production systems. In *2018 IEEE conference on control technology and applications (CCTA)* (pp. 521–526). IEEE, http://dx.doi.org/10.1109/CCTA.2018.8511385.

Krishnamoorthy, D., Foss, B., & Skogestad, S. (2018). Steady-state real-time optimization using transient measurements. *Computers & Chemical Engineering*, *115*, 34–45. http://dx.doi.org/10.1016/j.compchemeng.2018.03.021.

Krishnamoorthy, D., & Skogestad, S. (2022). Real-time optimization as a feedback control problem - A review. *Computers & Chemical Engineering*, Article 107723. http://dx.doi.org/10.1016/j.compchemeng.2022.107723.

Krishnamoorthy, D., Valli, C., & Skogestad, S. (2020). Real-time optimal resource allocation in an industrial symbiotic network using transient measurements. In *2020 American control conference (ACC)* (pp. 3541–3546). IEEE, http://dx.doi.org/10.23919/ACC45564.2020.9147578.

Krstić, M., & Wang, H.-H. (2000). Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica*, *36*(4), 595–601. http://dx.doi.org/10.1016/S0005-1098(99)00183-1.

Martí, R., Navia, D., Sarabia, D., & De Prada, C. (2012). Shared resources management by price coordination. (pp. 902–906). http://dx.doi.org/10.1016/B978-0-444-59520-1.50039-7.

Matias, J. O., & Le Roux, G. A. (2018). Real-time optimization with persistent parameter adaptation using online parameter estimation. *Journal of Process Control*, *68*, 195–204. http://dx.doi.org/10.1016/j.jprocont.2018.05.009.

Matias, J., Oliveira, J. P. C., Roux, G. A. C. L., & Jaschke, J. (2021). Steady-state real-time optimization using transient measurements on an experimental rig.

Morari, M., Arkun, Y., & Stephanopoulos, G. (1980). Studies in the synthesis of control structures for chemical processes: Part I: Formulation of the problem. Process decomposition and the classification of the control tasks. Analysis of the optimizing control structures. *AIChE Journal*, *26*(2), 220–232. http://dx.doi.org/10.1002/aic.690260205.

Nazari, S., Sonntag, C., Stojanovski, G., & Engell, S. (2015). A modelling, simulation, and validation framework for the distributed management of large-scale processing systems. (pp. 269–274). http://dx.doi.org/10.1016/B978-0-444-63578-5.50040-2.

Skogestad, S. (2000). Plantwide control: the search for the self-optimizing control structure. *Journal of Process Control*, *10*(5), 487–507. http://dx.doi.org/10.1016/S0959-1524(00)00023-8.

Skogestad, S. (2003). Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control*, *13*(4), 291–309. http://dx.doi.org/10.1016/S0959-1524(02)00062-8.

Skogestad, S., & Postlethwaite, I. (2005). *Multivariable feedback control*. New York, NY, USA: John Wiley and Sons.

Srinivasan, B., François, G., & Bonvin, D. (2011). Comparison of gradient estimation methods for real-time optimization. In E. N. Pistikopoulos, M. C. Georgiadis, & A. C. Kokossis (Eds.), *Computer Aided Chemical Engineering*: *vol. 29*, *21st European symposium on computer aided process engineering* (pp. 607–611). Elsevier, http://dx.doi.org/10.1016/B978-0-444-53711-9.50122-X.

Stojanovski, G., Maxeiner, L., Kramer, S., & Engell, S. (2015). Real-time shared resource allocation by price coordination in an integrated petrochemical site. In *2015 European control conference (ECC)* (pp. 1498–1503). IEEE, http://dx.doi.org/10.1109/ECC.2015.7330751.

Tsamardinos, I., Brown, L. E., & Aliferis, C. F. (2006). The max–min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, *65*(1), 31–78. http://dx.doi.org/10.1007/s10994-006-6889-7.

Uzawa, H. (1960). Walras' tatonnement in the theory of exchange. *Review of Economic Studies*, *27*(3), 182–194.

Walter, E., & Pronzato, L. (1997). *Identification of parametric models: From experimental data*. Springer Verlag.

Wenzel, S., Paulen, R., Stojanovski, G., Krämer, S., Beisheim, B., & Engell, S. (2016). Optimal resource allocation in industrial complexes by distributed optimization and dynamic pricing. *At - Automatisierungstechnik*, *64*(6), 428–442. http://dx.doi.org/10.1515/auto-2016-0003.