Steady-State Real-Time Optimization using Transient Measurements

Dinesh Krishnamoorthy[a], Bjarne Foss[b], Sigurd Skogestad[a,*]

[a] *Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway*
[b] *Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway*

Abstract

Real-time optimization (RTO) is an established technology, where the process economics are optimized using rigourous steady-state models. However, a fundamental limiting factor of current static RTO implementation is the steady-state wait time. We propose a "hybrid" approach where the model adaptation is done using dynamic models and transient measurements and the optimization is performed using static models. Using an oil production network optimization as case study, we show that the Hybrid RTO can provide similar performance to dynamic optimization in terms of convergence rate to the optimal point, at computation times similar to static RTO. The paper also provides some discussions on static versus dynamic optimization problem formulations.

Keywords: Real-Time Optimization, steady-state optimization, dynamic models, production optimization, hybrid RTO

1. Introduction

Industrial processes usually consist of many operations and various components that have their own objectives and complex interconnections with other components. The safe and optimal operation of such large and complex processes requires meeting goals and objectives in different time scales ranging from planning and scheduling to fast corrective actions for regulatory control. Realizing all the goals and constraints as a whole can be a very challenging and unrealistic task. Thus the operation of any process is typically decomposed into various decision making layers [1, Ch.10], [2]. Such a hierarchical implementation is a widely accepted industry standard [3] and is also well studied in academic literature under the context of plantwide control, see for e.g. [4], [5],[6] and [7] to name a few. A typical control system hierarchy is shown in Fig.1, where the time horizon for the decisions are clearly shown for each layer. The information flow in this control hierarchy is such that the upper layers provide setpoints to the layer below, which reports back any problems in achieving this [4]. The upper three layers in Fig.1 explicitly deals with the optimal economic operation of the process. Generally, there is also more multivariable coordination as we move upwards in the hierarchy [1].

The long term decisions involve selecting an investment strategy, operation model, infrastructure etc, which is typically known as *Asset management*. Then there are decisions taken on a horizon of days such as plantwide scheduling. This is followed by decisions that have to be taken on decision horizon in the timescale of hours known as *Real-time Optimization* (RTO). This decision making step is the focus of this paper. It aims to maximize the revenue and minimize the operational costs of hour-by-hour operations, thereby optimizing the economics of the process. This is followed by a faster control and automation layer that accounts for fast corrective actions. The control layer could be broadly divided into supervisory and regulatory layers, where the objective of the supervisory layer (such as MPC) is to track the reference trajectory provided by the RTO layer and to look after other variables and constraints. On the other hand, the primary objective of the regulatory layer is to stabilize and avoid drift in the variables.

The economic optimization of any process performance is becoming more crucial in the face of growing competition. Process optimization directly enables safe operation, cost reduction, improving product quality and meeting environmental regulations and this is the main focus of the RTO layer.

A widely accepted definition of real-time optimization is that it is a work flow where the decision variables are optimized using the system model and the economic model along with the process constraints by solving some kind of mathematical optimization [8]. In order to account for process disturbances and plant-model mismatch, there has been advancements in measurement-based optimization that adjust the optimal inputs in real time, hence defining RTO as a workflow that optimizes process performance by iteratively adjusting the decision variables using

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

[*]Corresponding author
Email addresses: dinesh.krishnamoorthy@ntnu.no (Dinesh Krishnamoorthy), bjarne.foss@ntnu.no (Bjarne Foss), skoge@ntnu.no (Sigurd Skogestad)

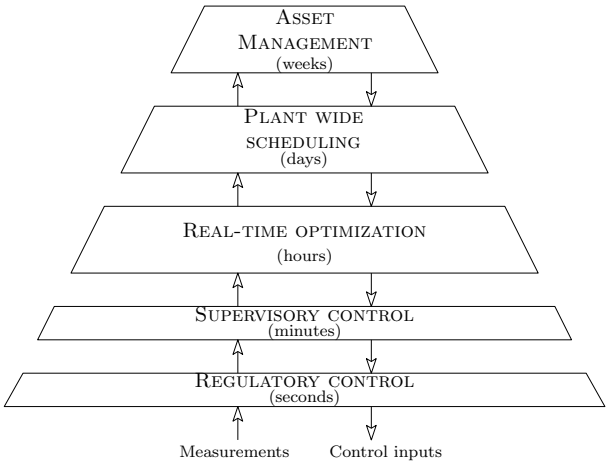


Figure 1: Typical control hierarchy in process control

measurement data [9]. A good overview and classification of the different RTO methods can be found in [9] and [10].

In many process control application, the real-time optimization uses nonlinear steady state process models to compute the optimal setpoint at steady state operation [11]. The justification for using static models is twofold; 1) the economic operation of the plant often occurs at steady state operation, 2) the computed optimal decision variables (control inputs) needs to be simply kept at constant values (provided as setpoints) over long time-periods, hence making the implementation easier [11]. RTO is also provided with constraints such as process and equipment constraints, storage and capacity constraints, product quality constraints etc. In addition, RTO uses an economic model that constitutes the cost of raw material, value of the products, operational costs, environmental regulations etc. to evaluate the economics of operation.

Although there have been recent developments in different approaches to RTO [9], the most common approach to commercial RTO implementation is the so-called two-step model-adaptation approach [12], [13] as shown in Fig.2. The static model used in the RTO is parameterized by a set of unknown or uncertain parameters, which are updated using measurement data in the first step. In the second step, the updated model is used to compute the optimal set of decision variables by solving a numerical optimization problem. The repeated identification and optimization scheme using static models is used in many commercial RTO software packages [14].

However traditional static RTO faces some challenges which limits its industrial use, including (in expected order of importance):

1. Cost of developing and updating the model structure (offline update of the model),
2. Model uncertainty, including uncertain values of disturbances and parameters and structural uncertainty (online update of the model),
3. Frequent grade changes, which makes steady-state optimization less relevant,
4. Dynamic limitations, including infeasibility due to (dynamic) constraint violation.

It may seem that reasons 2, 3, and 4 suggest towards dynamic optimization. However, except for processes with frequent grade changes (reason 3), static optimization may be close to the economic optimum, at a much lower computational cost. In a recent review paper on current practices of RTO [3], the authors conclude that a fundamental limiting factor of RTO implementation is the steady state wait-time associated with the online update of the model (reason 2). Since only static models are used, the model adaptation step must be carried out using measurements that corresponds to steady-state operation. If the process is frequently subject to disturbances or if the settling times are rather long, this can lead to the plant being operated in transients for significant periods of time. With the inadequacy of steady-state measurements, the model is not updated frequently. Consequently the plant is operated sub-optimally for long periods of time.

The authors in [3] briefly suggest the approach of using dynamic terms that would impact only the model adaptation step as a potential research direction to address this issue. In this paper, we therefore analyze the approach of using dynamic model adaptation and transient measurements. We show that by using a hybrid approach with dynamic models for estimation together with static models for optimization, the problems with steady-state detection (SSD) and model adaptation can be handled more efficiently, hence leading to an efficient RTO implementation.

The dynamic limitations (reason 4) can in most cases be handled efficiently by a setpoint tracking layer below, for example, using MPC. Therefore in many applications, the economic gain by using a dynamic RTO may be negligible compared to the proposed hybrid RTO structure. With the recent surge of interest in dynamic optimization and the so-called economic MPC, where the RTO layer and control layers are tightly integrated, there is a lack of consensus in the literature on the use of static versus dynamic optimization. Therefore, we also discuss the use of static versus dynamic optimization, namely, when static optimization is sufficient and when the use of dynamic optimization may be required.

The main contribution of this paper is the hybrid RTO approach (i.e. static optimization with dynamic model adaptation) that directly addresses the issue of steady state wait-time.

The rest of the paper is structured as follows: A brief review of traditional RTO structures and the implementation issues are provided in section 2. Modifications to the traditional RTO approach are proposed in section 3 which are illustrated using an application example in section 4. Discussions on the use of static versus dynamic problem formulation is provided in section 5 before concluding the paper in section 6.

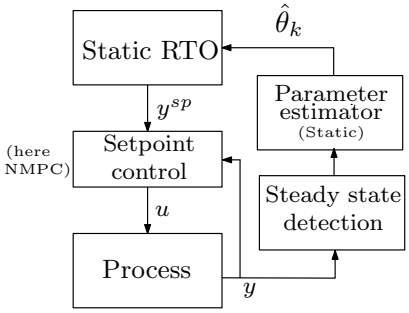


Figure 2: Traditional RTO with static model adaptation and static optimization.

2. Traditional RTO

2.1. Static RTO (SRTO)

The traditional static RTO implementation in Fig.2, based on the two-step approach, is briefly summarized below:

- Steady state detection and data pre-processing - This initial step detects if the plant is operating close enough to steady state, to start the RTO sequence.
- Static parameter estimation (Step 1) - The model parameters are adjusted to match the current data, using regression techniques. The parameter estimation step usually consists of data reconciliation and model adaptation. The measurement data is screened for unreasonable data such as gross errors, for example based on material and energy balances. Suitable actions are taken to rectify or eliminate any erroneous data before it is used to update the model. Considerable process knowledge may be required to decide which model parameters needs to be updated as noted in [11, Ch.19] and [15].
- Static Optimization (Step 2) - Given an objective function, process constraints and an updated model, the optimum setpoints are computed using mathematical optimization methods.

The setpoints computed by the RTO are then provided to the lower layer supervisory control, where a dynamic optimization problem may be solved online, typically using simplified linear models with constraints (in the framework of MPC) to minimise the deviation of the measurements from the setpoint over a period of time. Fig.2 shows a typical RTO structure, where the steady state process data is used for adapting the static model which is used in the Static RTO.

Consider a process described by a discrete-time nonlinear model,

$$x_{k+1} = f(x_k, u_k, \theta_k) \tag{1}$$
$$y_k = h(x_k, u_k)$$

where $x_k \in \mathbb{R}^{n_x}$, $u_k \in \mathbb{R}^{n_u}$ and $y_k \in \mathbb{R}^{n_y}$ are the states, process inputs and process measurements at time step k respectively. The model is parameterized by a set of time-varying parameters and disturbances jointly represented by $\theta_k = [p_k^T, d_k^T]^T \in \mathbb{R}^{n_\theta}$. The model equations are represented by $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \to \mathbb{R}^{n_x}$ and $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$.

Let the static counterpart for this model be described by,

$$y = f_{ss}(u, \theta) \tag{2}$$

where $f_{ss} : \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \to \mathbb{R}^{n_y}$ describes the static input-output equilibrium map.

Once the plant is operating at steady-state, the model parameters are updated using the steady state measurements (step 1 of 2). The model parameter adaptation scheme is based on minimizing the error between the model predicted value and the measurement data.

The updated parameter vector $\hat{\theta}_k$ is then used in the optimization problem (step 2 of 2). The optimization problem then computes the optimal decision variables u^* that optimizes the process performance, while satisfying process and operating constraints. The static optimization problem using the two-step approach for this system can thus be stated mathematically as follows,

Step 1: Static Estimation

$$\hat{\theta}_k = \arg\min_{\theta} \|y_{meas} - f_{ss}(u_k, \theta)\|_2^2 \tag{3}$$

Step 2: Static Optimization

$$u^*_{k+1} = \arg\min_{u} J(y, u) \tag{4}$$
$$\text{s.t. } y = f_{ss}(u, \hat{\theta}_k)$$
$$g(y, u) \leq 0$$

where $y_{meas} \in \mathbb{R}^{n_y}$ denotes the measurements from the plant, $J : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}$ describes the objective function, $g : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_c}$ describes vector of nonlinear constraints that may be imposed such as process and operating constraints including bounds on the the process inputs and outputs. Note that J and g are not directly dependent on the parameters θ, but implicitly via the process outputs y governed by the model (2).

Challenges with steady-state detection. Many commercial RTO softwares uses either statistical methods or heuristic methods or a combination of both to verify the stationarity of the data for a fixed window length in the past. A detailed description of the different steady-state detection routines used in commercial RTO systems can be found in [14]. Tolerances are specified by the user to determine if the process is "close enough" to steady state and the process is said to have reached steady state when all the measurements are within the specified tolerances [16]. If the tolerances are specified without proper evaluation of the data window length, then the steady-state detection might

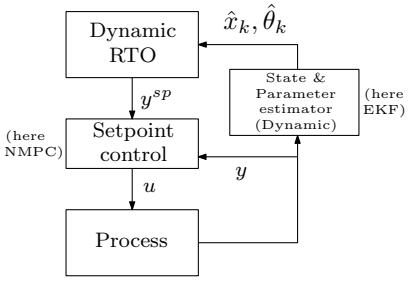


Figure 3: Dynamic RTO with dynamic model adaptation and dynamic optimization.

erroneously accept transient data as stationary data. Using transient data to update static models results in estimation errors which are then propagated to the optimization routine. The authors in [14] demonstrate this issue using data from a real industrial application.

Another challenge in many processes is that it may be frequently subject to disturbances. This results in the process being operated mostly at transients, thus hindering model adaptation. This is further worsened if the process has long settling times. In such processes, the model parameters are not updated frequently due to inadequate availability of steady state measurements. Consequently the process may be operated sub-optimally for long periods of time, until the model parameters are updated again.

2.2. Dynamic RTO (DRTO)

In the recent past, there has also been many developments in the use of *Dynamic RTO* (DRTO) and the closely related economic model predictive control (EMPC), that provides an optimal input trajectory using a dynamic model instead of a static model. Consider the dynamic system (1), the two-step approach to dynamic RTO at each time step k can be given by,

Step 1: Dynamic Estimation

$$\hat{\theta}_k = \arg\min_{\theta} \|y_{meas,k} - h(x_k, u_k)\| \quad (5)$$
$$s.t.\ \ x_k = f(x_{k-1}, u_{k-1}, \theta)$$

Step 2: Dynamic Optimization

$$u_t^* = \arg\min_{u_t} \sum_{t=k}^{k+T} J(y_t, u_t) \quad (6)$$
$$s.t.\ \ x_{t+1} = f(x_t, u_t, \hat{\theta}_k)$$
$$y_t = h(x_t, u_t)$$
$$g(y_t, u_t) \leq 0$$
$$x_k = \hat{x}_k \qquad \forall t \in \{k, \ldots, k+T\}$$

where the subscript $*_t$ represents each sample in the optimization horizon of length T.

Although the use of dynamic models for model adaptation and optimization may eliminate the requirements of steady state detection, solving a dynamic nonlinear optimization problem for large-scale systems may be challenging, even with today's computing power. The authors in [17] point that many numerical issues associated with DRTO must be addressed before it can be widely implemented in industrial applications. In addition, the dynamic model requires additional parameters including a model of the lower-layer control system. Static RTO is therefore still more prevalent in many industrial applications.

Static RTO uses the same static model in both the steps of the two-step approach. Similarly, dynamic RTO uses the same dynamic model in both the steps of the two-step approach. To address the computational challenges in dynamic optimization and to address the steady-state wait-time issue in static RTO, a "hybrid RTO" structure can be considered, where dynamic models are used in the model adaptation step and static models are used in the optimization step.

3. Hybrid RTO (HRTO)

If the primary objective is to optimize the steady-state performance of the process, then dynamic terms in the model need only be introduced in the model adaptation step. When dynamic models are used in the model adaptation, transient data can be used to update the model, without the need to discard big chunks of data. The updated model parameters can then be used in the static model used in the optimizer as shown in Fig.4. To illustrate this, consider the discrete dynamic model (1) and the corresponding steady-state model (2). The model adaptation via the two step approach would then be given by,

Step 1: Dynamic Estimation

$$\hat{\theta}_k = \arg\min_{\theta} \|y_{meas,k} - h(x_k, u_k)\| \quad (7)$$
$$s.t.\ \ x_k = f(x_{k-1}, u_{k-1}, \theta)$$

Step 2: Static Optimization

$$u_{k+1}^* = \arg\min_{u} J(y, u) \quad (8)$$
$$s.t.\ \ y = f_{ss}(u, \hat{\theta}_k)$$
$$g(y, u) \leq 0$$

At every time step k, the the dynamic model estimator provides the estimate of the uncertain variables $\hat{\theta}_k$ and the static optimization problem is solved with the updated model to find the new optimal steady-state operating point. Therefore, as opposed to the traditional static RTO, the static optimization problem in the hybrid RTO approach is solved at each time step k and the resulting optimal setpoints are provided to the setpoint tracking control.

Development of model based control design around 1960, together with seminal works such as the Kalman filter [18] led to the development of identification theory in the control literature. Different methods exists today that can be used to estimate the unknown variables in a dynamic model. Some of the methods that are commonly used include, but are not restricted to, recursive least squares

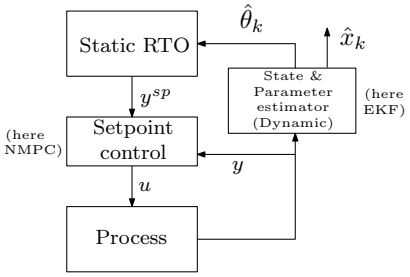


Figure 4: Hybrid RTO with dynamic model adaptation and static optimization.

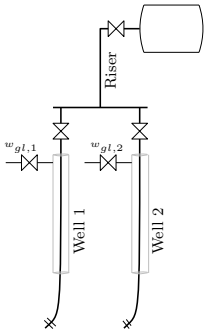


Figure 5: Oil production network with 2 gas lifted wells producing to a common riser manifold.

estimation, nonlinear Kalman filter variants such as extended Kalman filter (EKF) and unscented Kalman Filter (UKF), optimization-based methods such as the moving horizon estimator (MHE) etc. In the reminder of the paper, we consider the Hybrid RTO approach using an extended Kalman filter for online parameter estimation. The use of EKF where parameter estimation is the focus can be found in several examples in literature [19]. The framework of using an extended Kalman filter for combined state and parameter estimation can be found in Appendix A.

Modelling Effort. - In most chemical processes, the so-called first principle models used in RTO applications are based on the conservation of mass, energy and momentum which naturally gives rise to ordinary differential equations (ODE). Additional algebraic equations may be specified to model unknown variables, such as flow through an orifice, hydrostatic and frictional pressure losses, reaction stoichiometry, equation of state etc. Once the mathematical models have been defined for a single control volume, multi-staged models such as distillation columns or models with many control volumes can be easily modelled by joining up the same number of mathematical models. A good overview of mathematical modeling for many typical chemical processes can be found in [20].

Often, a chemical engineer starts out the modelling task using dynamic equations. The dynamic models developed using physical principles are converted to the static models by setting all the derivative terms to zero, [20]. In many cases, the development of models for dynamic estimation may require little extra modelling effort. However, dynamic models require additional parameters including mass and energy holdups. In addition, the dynamic model used for dynamic optimization (DRTO) needs a model of the lower-layer control system. In the hybrid RTO case (HRTO), where the dynamic model is only used for parameter estimation, a simpler representation of the lower level control system may be sufficient, for example, by not including the control system or assuming perfect control. Developing and maintaining a dynamic model for the model adaptation step can be justified if the performance improvement is significant when using the hybrid RTO approach, since this enables better implementation of the RTO.

4. Illustrative example

In this paper we study the problem of production optimization in an oil and gas production network to demonstrate the use of the hybrid RTO (HRTO) approach and compare it to traditional static RTO (SRTO) and dynamic RTO (DRTO). We consider a gas lifted well network consisting of n_w gas lifted wells connected to a common riser manifold as shown in Fig.5 for a case with two wells. In oil production wells, when the reservoir pressure is not sufficient to lift the fluids to the surface economically, artificial lift methods are used to boost the production from the wells. Gas lift is one such widely used artificial lift method where compressed gas is injected at the bottom of the well to reduce the density of the fluid column inside the well, thus reducing the pressure drop in the well tubing. The fluids from the reservoir enters the well tubing and mixes with the lift gas. The mixture then flows through the common riser manifold and finally enters the topside processing facility such as a separator where the oil and gas phases are separated. The production network may be constrained by total gas processing capacity or the total gas that is available for gas lift injection. The objective of the production optimization problem is to compute the gas lift injection rate for each well such that the total oil production is maximized while satisfying operational constraints.

A single gas lifted well model is based on the model presented in [21]. We use this as a base to build a gas lifted well network model. Production from a cluster of $\mathcal{N} = \{1, \cdots, n_w\}$ gas lifted wells was modelled using mass balances for each phase. The system is described as semi-explicit index-1 DAE system of the form.

$$\dot{x} = \mathbf{F}_c(x, z, u, \theta) \tag{9}$$

$$\mathbf{G}(x, z, u, \theta) = 0 \tag{10}$$

where $x \in \mathbb{R}^{n_x}$ and $z \in \mathbb{R}^{n_z}$ represent the differential and algebraic states respectively, $u \in \mathbb{R}^{n_u}$ represents the degrees of freedom which are the gas lift injection rates for each well w_{gl_i}. $\theta \in \mathbb{R}^{n_p}$ represents the vector of uncertain variables. In this work, we consider the gas-oil ratio

(GOR) from the reservoir for each well to be the uncertain parameter (disturbances). See Appendix B for more detailed description of the models used .

The static optimization problem for such a system can be written as

$$\max_{w_{gl_i}} J = \left(\$_o \sum_{i \in \mathcal{N}} w_{po_i} - \$_{gl} \sum_{i \in \mathcal{N}} w_{gl_i} \right) \tag{11a}$$

s.t.

$$\mathbf{F}_c(x, z, u, \theta) = 0 \tag{11b}$$

$$\mathbf{G}(x, z, u, \theta) = 0 \tag{11c}$$

$$\sum_{i \in \mathcal{N}} w_{pg_i} \leq w_{g_{max}} \tag{11d}$$

where $\$_o$ and $\$_{gl}$ prices are the value of oil and cost of gas compression respectively. w_{po_i} and w_{pg_i} are the produced oil and gas rates from each well i. The static process model is enforced as equality constraint in (11b)-(11c). $w_{g_{max}}$ is the total capacity constraint which is enforced in (11d). Hence from a process control point of view, this is equivalent to real-time optimization. For each of the RTO case shown in this paper (SRTO, DRTO and HRTO) a setpoint tracking MPC layer was used below to track the setpoints of the gas lift injection rates provided by the RTO layer above. In this paper, we use a nonlinear MPC, but similar results would be achievable with a more traditional linear MPC. The sampling time of the setpoint tracking controller was set to 5 min and a prediction horizon of 24 samples. A sufficiently long prediction horizon of 2 hours was chosen to ensure stability [22].

Any produced gas rate that exceeds the maximum capacity of $w_{g_{max}}$ in (11d) is flared to avoid pressure build-up in the topside processes. Gas flaring is often very expensive due to environmental costs in the form of carbon tax. The gas capacity constraint (11d) was therefore implemented as soft constraints using exact penalty functions and slack variables, where the slack variables are penalized in the cost function [23] as shown below,

$$\max_{w_{gl_i}} J' = \left(\$_o \sum_{i \in \mathcal{N}} w_{po_i} - \$_{gl} \sum_{i \in \mathcal{N}} w_{gl_i} \right) - \$_{fl} \|w_{fl}\|$$

$$\text{s.t.} \sum_{i \in \mathcal{N}} w_{pg_i} \leq w_{g_{max}} + w_{fl} \tag{12}$$

$$(11b) - (11c)$$

where the flared gas rate $w_{fl} \geq 0$ is the slack variable and $\$_{fl}$ is the cost associated with gas flaring that is penalized in the cost function. Note that, the exact penalty function for soft constraint would typically not have any physical meaning. However, in this example, the slack variable is the flared gas and the corresponding penalty function would be the cost of gas flaring. An alternate equivalent formulation would be to compute the flared gas as a part of the system model and minimize the gas flaring in the cost function.

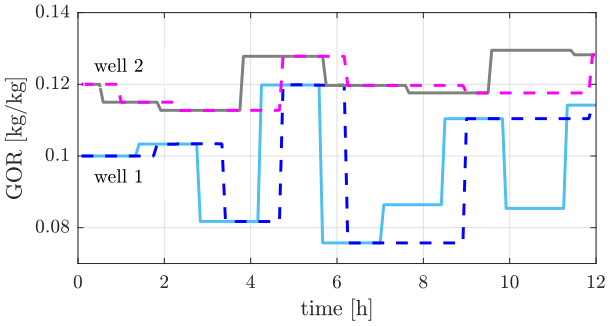


Figure 6: True GOR parameters used in the simulator (solid lines) and the estimated GOR using steady-state measurements used in SRTO (dashed lines). The steady-state wait time varies from about 30 mintues to several hours.

The NLP problem (12) was developed in `CasADi` v3.0.1 [24] using the `MATLAB` R2017a programming environment and solved using `IPOPT` version 3.12.2 [25] running with mumps linear solver on a 2.6GHz workstation with 16GB memory. The plant (simulator) was implemented using `IDAS` integrator [26].

In case of Dynamic RTO and for setpoint tracking NMPC, the system (9) is discretized using a third order direct collocation scheme. The dynamic optimization problem is similar to (11) with the static model (11b)-(11c) replaced with the discretized dynamic process model. The resulting NLP was solved using `IPOPT` as described above.

In all the simulations shown in this paper, the parameter GOR varies in the simulator as shown in Fig.6 (solid lines). We simulate the system for a total simulation time of 12 hours. The gas processing capacity is assumed to be constrained at $w_{g_{max}} = 10kg/s$. The optimal steady state gas lift injection rates for the different GOR combinations simulated in Fig.6 are summarised in Table.1. It is evident that the optimal gas lift injection rates are sensitive to changes in GOR. If the GOR used in the optimizer is not updated, then the plant will be operated sub-optimally.

4.1. Static RTO (SRTO)

We consider the traditional static RTO approach in Fig.2, where the Static RTO provides the optimal setpoints to the lower setpoint tracking layer. In the fully static RTO case, the steady state detection was based on the comparison of total variance of a signal in the recent data window of fixed length as described in [14]. When steady-state operation is detected, *GOR* is estimated from the measurement data. Fig.6 compares the true value of the GOR used in the simulator (solid lines) and the estimated GOR value used by the RTO (dashed lines). There is a significant delay before the models are updated and re-optimized due to the steady-state wait-time. This results in suboptimal operation for significant periods of time. In this simulation, the static RTO updated the setpoints 10 times. The performance of this fully static RTO approach are compared with dynamic RTO and the proposed hybrid RTO approach in Section 4.4

Table 1: The optimal gas lift injection rates for different GOR combinations used in the simulations.

GOR well 1	[kg/kg]	0.1	0.1033	0.0817	0.1198	0.0757	0.0864	0.1104	0.0854
GOR well 2	[kg/kg]	0.12	0.115	0.1127	0.1278	0.1197	0.1176	0.1176	0.1295
$w^*_{gl_1}$	[kg/s]	1.6062	1.497	2.217	0.9437	2.414	2.06	1.26	2.091
$w^*_{gl_2}$	[kg/s]	0.7812	0.9557	1.042	0.5013	0.799	0.8694	0.8626	0.4516

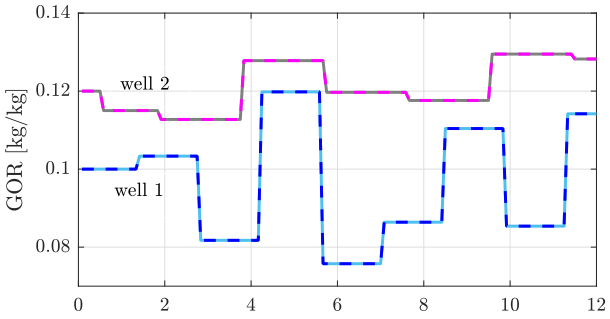


Figure 7: True GOR parameters used in the simulator (solid lines) and the almost identical estimated GOR using EKF used in HRTO and DRTO (dashed lines).

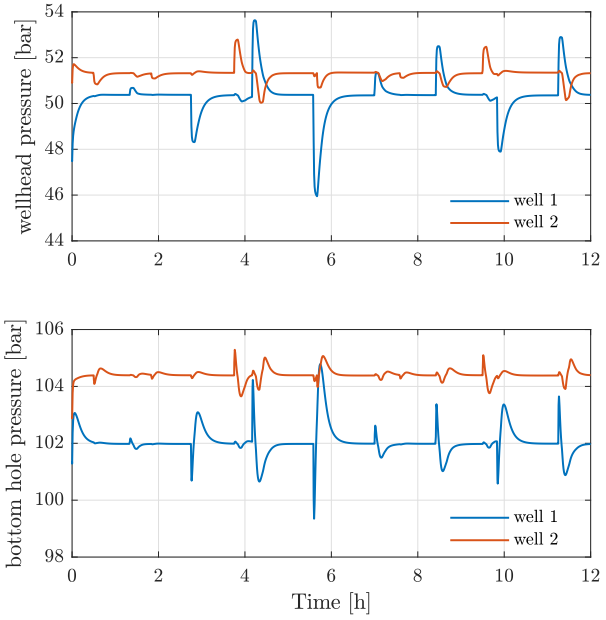


Figure 8: Plot showing some of the measurements used in EKF to estimate GOR.

4.2. Hybrid RTO (HRTO)

With the proposed hybrid RTO as shown in Fig.4, we estimate the uncertain parameters and disturbances using the dynamic model and optimize using the static model. For the dynamic model adaptation, we use a discrete time extended Kalman filter to estimate the uncertain parameters (GOR of each well) as shown in Appendix A. The annulus pressure, wellhead pressure and bottomhole pressure for each well, manifold pressure, riser-head pressure, total oil and gas flow rates at the separator are commonly available measurements in an oil production network and are here used for state and parameter estimation in the EKF.

The EKF updates the GOR estimate with the same sampling time as the optimizer, which is every 5 minutes. The estimated GOR using the EKF is shown in Fig.7. The optimizer uses the updated GOR to compute the steady state optimal gas lift rates which are given as setpoints to the lower level setpoint-tracking NMPC layer. Some measurements such as bottom hole pressure and wellhead pressure measurements are plotted in the Fig.8. It can be clearly seen that the system is in a transient phase for significant amount of time due to the frequent changes in GOR. Nevertheless, with dynamic estimation, the GOR is constantly adapted, as opposed to the steady-state wait time in the SRTO case. As a result, the hybrid RTO is able to compute the new optimal steady state gas lift rates as the GOR changes without having to wait for the system to settle to steady state.

4.3. Dynamic RTO (DRTO)

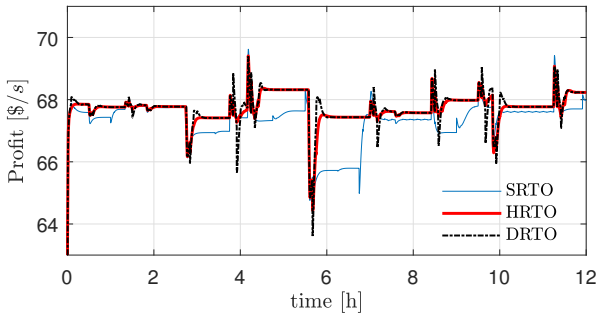
Here, we use a dynamic RTO approach as shown in Fig.3 instead of a static optimizer. But otherwise the setup was the same as for the hybrid RTO in Section 4.2. The simulation results are compared with SRTO and HRTO in the next subsection.

4.4. Comparison of SRTO, HRTO and DRTO

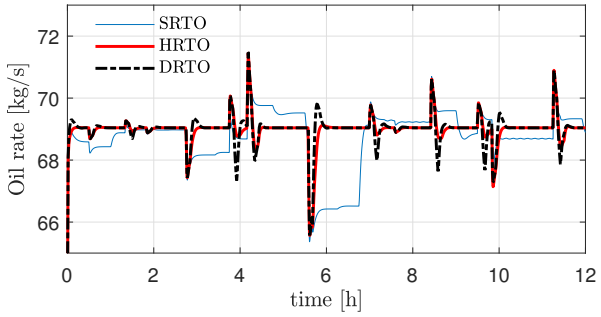
The objective function (12) along with the oil and gas production rates obtained with the SRTO, HRTO and DRTO are compared in Fig.9a, Fig.9b and Fig.9c respectively. As expected, SRTO (shown in thin blue lines) leads to suboptimal operation and the total produced gas also violates the capacity constraint of $10kg/s$ in (11d) for sig-

Table 2: Comparison of average computation time, maximum computation time and the total integrated oil production over a simulation time of 12 hours for the different RTO approaches.

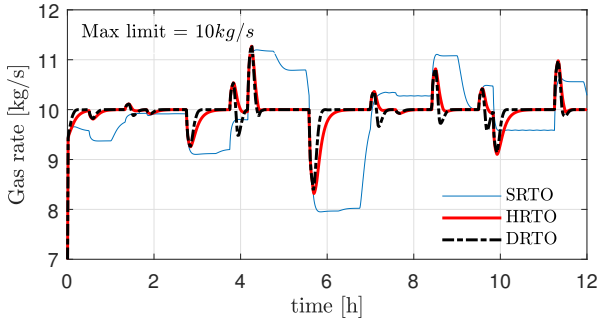
	avg. time [s]	max time [s]	Integrated Profit [$\times 10^6$\$]	Total oil [ton]	Flared gas [ton]
SRTO	0.0184	0.0223	1.8256	2969.5	10.93
HRTO	0.0199	0.0282	2.7019	2980.2	2.25
DRTO	0.9025	3.3631	2.7509	2980.9	1.77



(a)



(b)



(c)

Figure 9: Comparison of (a) objective function (11), (b) oil production rate and (c) gas production rate. SRTO is shown in thin blue lines, HRTO is shown in solid red lines and DRTO is shown in black dashed lines.

nificant periods of time. The hybrid RTO (red solid lines) and dynamic RTO (black dashed lines) have similar performance in terms of optimality. We see that the process is maintained at optimal operation and disturbances are swiftly counteracted. The integrated profit (12), oil production and the total gas flared over a period of 12 hours obtained with the three methods are summarised in Table.2.

The average computation times for the static RTO, hybrid RTO and dynamic RTO are also shown in Table 2. From the simulation results and the computation times, it can be seen that the Hybrid RTO provides a similar performance to the dynamic RTO in terms of convergence to the optimal point, but the computation time of the hybrid approach is about 100 times less than DRTO and about the same as SRTO. Additionally, for the HRTO and DRTO case, the average computation time for the EKF is 0.0026s, which is small compared to the computation time for the optimization problems.

The decision variables (setpoints) provided by the HRTO and DRTO are shown in Fig.10 whereas the gas lift rates actually implemented by the respective setpoint tracking controller are shown in Fig.11. It can be seen that, when the disturbance causes the total gas rate to exceed its limit, the dynamic RTO manipulates the setpoints to quickly come out of constraint violations, whereas the hybrid RTO simply provides the steady-state optimal setpoints. However, since the gas rate constraint of 10kg/s is included in the control layer below as a dynamic constraint, the actual gas lift rates provided by the setpoint tracking controllers are more similar. For example, consider the time between 4 and 5 hours in Fig.10 and Fig.11. This shows that dynamic limitations in many cases can be handled by the control layer below, which partly explains why the hybrid RTO scheme works well.

5. Discussion

5.1. On static versus dynamic optimization

In the previous section, we discussed the Hybrid RTO structure, where both dynamic and static models are used. One question that naturally arises is that when dynamic models are used for model adaptation, why not use the dynamic models also in the optimizer. Indeed, there is a clear trend and extensive research towards dynamic RTO and the closely related economic NMPC, see for example [27], [28] and [29]. In the face of this current trend, there is a lack of clear understanding on when static optimization is sufficient or under what conditions the use of dynamic optimization may be justified. Recently, some good discussions using case examples on appropriate problem formulations were provided in [30] for the petroleum production optimization problem, where the authors conclude that most production optimization problems can be solved using static optimization. On the other hand, batch processes, cyclic operations, operations that involves frequent

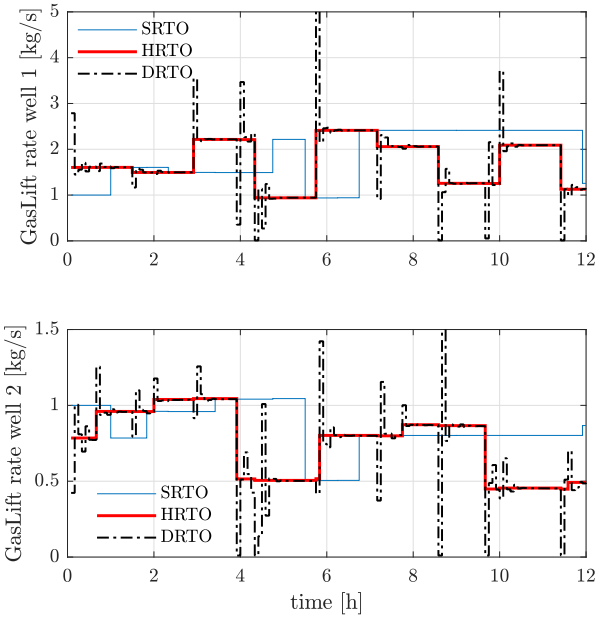


Figure 10: Optimal gas lift rate setpoint computed by SRTO (blue lines), HRTO (solid red lines) and DRTO (black dashed lines). These are the manipulated variables (decision variables) computed by the optimizers.

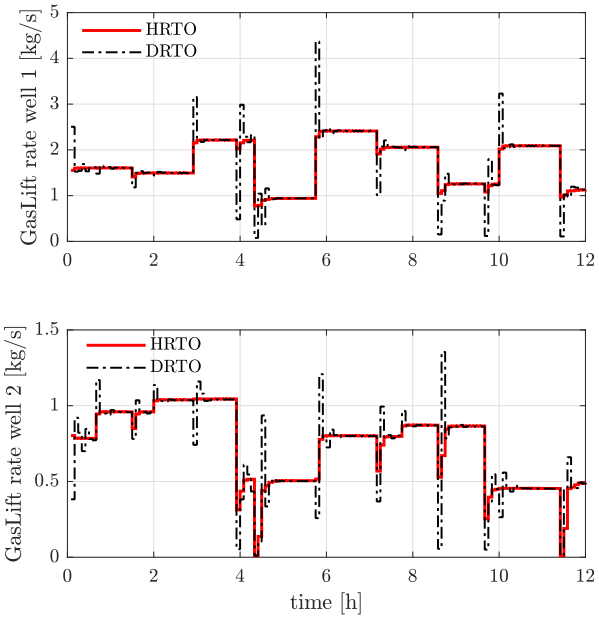


Figure 11: Optimal gas lift rates provided by respective setpoint tracking controllers for HRTO (solid red lines) and DRTO (black dashed lines). These are the implemented manipulated variables.

grade changes, start-up and shut-down etc. that involves transient operation would benefit from the use of dynamic optimization.

One of the main challenges with dynamic RTO, however, is computing power [17]. RTO often involves optimization of large-scale systems with large number of variables. This results in large nonlinear programming (NLP) problems. Additionally in dynamic optimization problems, the size of the problem increases significantly due to the additional dimension of time. As a result, dynamic optimization problems may be significantly more computationally demanding to solve than their static counterpart. For example, in our case study, the dynamic RTO had 3056 optimization variables as opposed to 22 optimization variables in SRTO and HRTO. The computational delay may impose limitations on how often the optimal setpoints can be updated. In some cases, the computational delay may even lead to performance degradation or closed loop instabilities [31].

The challenges with computational power is even more pronounced in the case where RTO has discrete integer decision variables. Mixed integer problems may be required if 1)the problem has discrete integer variables such as on-off switching, binary logics etc. or 2) if the nonlinear process model is modelled using piecewise-affine (PWA) models or surrogate models or 3) if the problem has non-convex cost or constraints. Such problem are often reformulated and solved efficiently using the mixed-integer framework. Mixed integer solvers employ methods such as branch-and-bound and cutting-plane methods combined with some heuristics and there are no solvers that guarantee solving dynamic mixed-integer problems in polynomial time. Formulating a RTO problem as mixed-integer problems is rather common in many industrial applications, see for example [32], [33], [34], [35] and the references therein. Hence static RTO remains the preferred formulation in many industrial applications.

In such cases, the proposed hybrid RTO approach can help tackle the steady-state wait-time issue, which is one of the fundamental limiting factor in traditional static RTO and at the same time circumvent the computational issues of dynamic RTO.

5.2. Advantages of static optimization (SRTO and HRTO)

However, computation cost is not the only reason static optimization is prevalent in industrial applications. A fundamental advantage of static optimization (in SRTO and HRTO) is that it does not have time as a variable. This avoids the causality issue, and allows for optimizing on decision variables other than the manipulated variables (MVs). For example, these decision variables may be pressure, level, composition and temperature. Consequently, this can 1) simplify the optimization and 2) allow the optimization to run on a much slower time scale because we can choose slow-varying variables as decision variables. This is also the principle behind self-optimizing control, where the goal is to choose the right decision variables y^{sp}

in Fig.2, which when kept constant leads to near optimal operation (i.e. acceptable loss) [4].

This advantage is not seen in our example because in all simulation cases, the decision variables are the gas lift injection rates (MVs). However, as a simple example, consider a small tank with one inflow (disturbance) and one outflow (MV). The setpoint for the level is assumed fixed. A dynamic RTO or economic NMPC would have the outflow as the decision variable, and it would need to be updated with the same frequency as the inflow disturbance. Essentially, the DRTO (or economic NMPC) is doing the job of the base layer PID controller, and needs to run at the speed of the base layer control system. However, with a static RTO, the decision variable could be the level setpoint, which would remain constant, irrespective of the disturbance.

Of course, optimizing on other decision variables assumes that we have a lower layer level controller, which takes care of disturbances on a fast time scale. In static RTO, the decision variables are setpoints to CVs of the lower-layer control system and the only assumption one needs to make about the control system is that it has integral action.

5.3. Dynamic estimation methods

As mentioned earlier, in the hybrid RTO approach, the model adaptation step is performed using dynamic models. Although we used an extended Kalman filter in this paper, we now provide a very brief discussion on the different dynamic estimation methods that can be used. Very simple methods such as filtered bias update or Implicit Dynamic Feedback (IDF) method may be used for simple parameter estimation problems. Implicit dynamic feedback is analogous to a PI controller as explained in [36] and may be used when a one-to-one pairing of measurements to the parameter is known.

For more complex multivariable systems, weighted least squares estimation or family of Kalman filters such as the extended Kalman filter (EKF) or unscented Kalman filter (UKF) may be used. Other recently developed Kalman filter based methods include sequential Monte Carlo methods and expectation maximization methods for parameter estimation [37]. It was noted in [38] that extended Kalman filter is the most widely used tool for nonlinear weighted least square estimation in chemical engineering. In the presence of uncorrelated gaussian white noise, this also corresponds to the maximum likelihood estimator (MLE) [38]. In terms of computational requirements, EKF for parameter estimation is known to be simple to implement and computationally fast compared to other methods [37]. The solution provided by EKF is accomplished through matrix multiplications and does not need to solve nonlinear optimization problems online. Additionally, there has also been some research in faster implementation of EKF for some applications, see for example [39], where the authors analyze the EKF matrices and reduce the number of computations by exploiting the sparsity and structure of the EKF matrices. The confidence interval provided by the covariance estimates may also be useful. One of the challenges with practical implementation of Kalman filter is the tuning which include the measurement and process noise covariance matrix elements and a forgetting factor, if included in the formulation [38]. Often these tuning parameters are chosen arbitrarily. Recent works introduced a computationally efficient approach to identify the noise covariance for nonlinear systems [40].

Optimization-based statistical methods such as the moving horizon estimation (MHE) has been receiving more attention recently, where a numerical optimization problem is solved to reduce the error between the estimates and the measurement. MHE is especially favourable in the case of constrained estimation. However, this method is more computationally intensive due to the dynamic optimization problem that has to be solved at each sampling instant. This may not be favourable if one of the motivations of using static optimization is to avoid solving dynamic optimization problems. MHE also often require additional observers in parallel such as EKF for estimating the arrival cost. For more detailed review of dynamic estimation methods the reader is referred to [36], [38] and the references therein.

5.4. Challenges with structural uncertainty

The model adaptation step mostly works under the assumption that any plant-model mismatch observed in the measurements arises from the set of uncertain parameters θ. Although this may reduce the error between the model predictions and the plant observations, this may not result in optimal operation if the model has the wrong structure. This is clearly demonstrated using a case example in [9], where the model parameters alone do not provide sufficient flexibility, because the assumed model structure was wrong.

The models may be structurally wrong either due to model simplification or lack of knowledge. Any unmodelled disturbance that enters the plant also results in structural uncertainty. A simple way to partly handle unmodelled disturbances is to add a bias term Δ and adapt the bias term online that describes the observations better.

$$x_{k+1} = f(x_k, u_k, \theta_k) + \Delta_k \tag{13}$$

This is similar to how process noise is added in a Kalman filter [19]. Integrated white noise models for Δ_k can be used in the adaptation step to account for the unmodelled effects that influences the steady-state predictions. For example, this can be achieved by modifying the EKF presented in (A.3) to include integrated white noise disturbance models. Using integrated white noise models to account for unmodelled effects is common and has also been used in offset-free model predictive control as described in [41]. Integrated disturbances can be modelled to enter either through the inputs or outputs or partially through

both as shown in [41]. The extended Kalman filter framework to estimate the bias using integrating white noise models is also shown in Appendix A. The estimated bias terms Δ_k are then brought onto the optimization models to account for the unmodelled effects.

It is important to note that, although one may be able to match the measurements by using the bias term Δ, this may not eradicate the structural mismatch and does not guarantee that the optimization problem converges to the optimum. This is partly due to the fact that the optimization objective (4) is unrelated to the parameter estimation objective in (3) as explained and demonstrated in [9].

In the recent literature, different approaches to real-time optimization have been proposed other than the standard two-step model adaptation method. The so-called modifier adaptation method introduced in [42] is a general way to handle structural mismatch. In this method, cost measurements are used by a slower model-free upper layer to compute the so-called modifier terms to adjust the computed gradient of the cost and constraint functions predicted by the optimization model [9]. However, it requires the cost to be measured directly in order to evaluate the plant gradient, which may not be readily available in many applications. The modifier adaptation method also uses a static optimization method and is plagued by the same issue of having to wait for steady-state before using the cost measurements. The possibility of using transient measurements in modifier adaptation has been recently explored in [43] and [44].

Related model-free RTO approaches such as extremum seeking control [45] and NCO (Necessary Condition of Optimiality) tracking control [46] were developed to avoid developing and updating a model[1]. In contrast to traditional RTO schemes, these methods rely solely on the process measurements to drive the system to its optimum. This is done by estimating the steady-state gradient from the cost to the inputs using the process measurements [47]. The estimated gradients are then driven to zero. Due to the estimation of steady-state gradient, these methods cannot use transient measurements directly. Using transient measurements may result in erroneous gradient estimation, resulting in undesired control actions. As a result, the convergence to the optimum may be prohibitively slow, which is the main disadvantage of such methods. Additionally, such purely data-based methods are also affected by abrupt disturbances, which may cause unnecessary deviations in the control inputs, as motivated in [48].

Modifier adaptation and model-free methods are currently an active area of research and although many application examples can be found in literature, these methods are far from real-industrial implementation in large-scale chemical plants [14]. The proposed hybrid RTO approach can therefore enable efficient implementation of the two-step model-adaptation based RTO approach in many industrial applications today. Static optimization solvers as well as dynamic estimation methods such as extended Kalman filters are commercially available tools that can be easily implemented using today's computing power.

[1] Reason 1 for limited industrial use of RTO described in Section 1

6. Conclusion

By using a hybrid RTO with dynamic models for model adaptation, we are able to efficiently use transient data for updating the models. Hence the optimization does not need to wait for the process to settle before the model parameters are updated. Additionally, by adopting a Hybrid RTO structure, with static models for optimization, numerical and computational issues associated with dynamic optimization can be avoided. This may lead to better utilization of the potential of RTO in many industrial applications. The use of hybrid RTO was demonstrated using an oil production network as case study, where similar performance to dynamic RTO was achieved at computation times similar to the traditional static RTO.

References

[1] S. Skogestad, I. Postlethwaite, Multivariable feedback control: analysis and design, 2nd Edition, Wiley New York, 2005.

[2] W. Findeisen, F. N. Bailey, M. Brdys, K. Malinowski, P. Tatjewski, A. Wozniak, Control and coordination in hierarchical systems, John Wiley & Sons, 1980.

[3] M. L. Darby, M. Nikolaou, J. Jones, D. Nicholson, RTO: An overview and assessment of current practice, Journal of Process Control 21 (6) (2011) 874–884.

[4] S. Skogestad, Plantwide control: the search for the self-optimizing control structure, Journal of process control 10 (5) (2000) 487–507.

[5] S. Skogestad, Control structure design for complete chemical plants, Computers & Chemical Engineering 28 (1) (2004) 219–234.

[6] T. Larsson, S. Skogestad, Plantwide control-a review and a new design procedure, Modeling, Identification and control 21 (4) (2000) 209.

[7] G. P. Rangaiah, V. Kariwala, Plantwide control: Recent developments and applications, John Wiley & Sons, 2012.

[8] K. G. Hanssen, B. Foss, Production optimization under uncertainty-applied to petroleum production, IFAC-PapersOnLine 48 (8) (2015) 217–222.

[9] B. Chachuat, B. Srinivasan, D. Bonvin, Adaptation strategies for real-time optimization, Computers & Chemical Engineering 33 (10) (2009) 1557–1567.

[10] B. Srinivasan, D. Bonvin, E. Visser, S. Palanki, Dynamic optimization of batch processes: II. role of measurements in handling uncertainty, Computers & chemical engineering 27 (1) (2003) 27–44.

[11] D. E. Seborg, D. A. Mellichamp, T. F. Edgar, F. J. Doyle III, Process dynamics and control, John Wiley & Sons, 2010.

[12] C. Y. Chen, B. Joseph, On-line optimization using a two-phase approach: an application study, Industrial & engineering chemistry research 26 (9) (1987) 1924–1930.

[13] T. E. Marlin, A. N. Hrymak, Real-time operations optimization of continuous processes, in: AIChE Symposium Series, Vol. 93, New York, NY: American Institute of Chemical Engineers, 1971-c2002., 1997, pp. 156–164.

[14] M. M. Câmara, A. D. Quelhas, J. C. Pinto, Performance evaluation of real industrial RTO systems, Processes 4 (4) (2016) 44.

[15] A. D. Quelhas, N. J. C. de Jesus, J. C. Pinto, Common vulnerabilities of rto implementations in real chemical processes, The Canadian Journal of Chemical Engineering 91 (4) (2013) 652–668.
[16] S. Cao, R. R. Rhinehart, An efficient method for on-line identification of steady state, Journal of Process Control 5 (6) (1995) 363–374.
[17] M. Campos, H. Teixeira, F. Liporace, M. Gomes, Challenges and problems with advanced control and optimization technologies, IFAC Proceedings Volumes 42 (11) (2009) 1–8.
[18] R. E. Kalman, A new approach to linear filtering and prediction problems, Transactions of the ASME–Journal of Basic Engineering 82 (Series D) (1960) 35–45.
[19] D. Simon, Optimal state estimation: Kalman, H infinity, and nonlinear approaches, John Wiley & Sons, 2006.
[20] R. G. Franks, Mathematical modeling in chemical engineering, Wiley, 1967.
[21] G. O. Eikrem, O. M. Aamo, H. Siahaan, B. Foss, Anti-slug control of gas-lift wells-experimental results, IFAC Proceedings Volumes 37 (13) (2004) 799–804.
[22] J. M. Maciejowski, Predictive control: with constraints, Pearson education, 2002.
[23] E. C. Kerrigan, J. M. Maciejowski, Soft constraints and exact penalty functions in model predictive control, in: Proc. UKACC International Conference (Control2000), 2000.
[24] J. Andersson, A General-Purpose Software Framework for Dynamic Optimization, PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium (October 2013).
[25] A. Wächter, L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Mathematical programming 106 (1) (2006) 25–57.
[26] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, C. S. Woodward, Sundials: Suite of nonlinear and differential/algebraic equation solvers, ACM Transactions on Mathematical Software (TOMS) 31 (3) (2005) 363–396.
[27] J. Kadam, W. Marquardt, M. Schlegel, T. Backx, O. Bosgra, P. Brouwer, G. Dünnebier, D. Van Hessem, A. Tiagounov, S. De Wolf, Towards integrated dynamic real-time optimization and control of industrial processes, Proceedings foundations of computer-aided process operations (FOCAPO2003) (2003) 593–596.
[28] M. Ellis, H. Durand, P. D. Christofides, A tutorial review of economic model predictive control methods, Journal of Process Control 24 (8) (2014) 1156–1178.
[29] M. G. Forbes, R. S. Patwardhan, H. Hamadah, R. B. Gopaluni, Model predictive control in industry: Challenges and opportunities, IFAC-PapersOnLine 48 (8) (2015) 531–538.
[30] B. Foss, B. R. Knudsen, B. Grimstad, Petroleum production optimization–a static or dynamic problem?, Computers & Chemical Engineering (In Press).
[31] R. Findeisen, F. Allgöwer, Computational delay in nonlinear model predictive control, IFAC Proceedings Volumes 37 (1) (2004) 427–432.
[32] V. Gunnerud, B. Foss, Oil production optimizationa piecewise linear model, solved with two decomposition strategies, Computers & Chemical Engineering 34 (11) (2010) 1803–1812.
[33] E. O. Hülse, E. Camponogara, Robust formulations for production optimization of satellite oil wells, Engineering Optimization (2016) 1–18.
[34] J. Kallrath, Mixed integer optimization in the chemical process industry: Experience, potential and future perspectives, Chemical Engineering Research and Design 78 (6) (2000) 809–822.
[35] F. Trespalacios, I. E. Grossmann, Review of mixed-integer nonlinear and generalized disjunctive programming methods, Chemie Ingenieur Technik 86 (7) (2014) 991–1012.
[36] J. D. Hedengren, A. N. Eaton, Overview of estimation methods for industrial dynamic systems, Optimization and Engineering 18 (1) (2017) 155–178.
[37] X. Sun, L. Jin, M. Xiong, Extended Kalman filter for estimation of parameters in nonlinear state-space models of biochemical networks, PloS one 3 (11) (2008) e3758.
[38] M. Leibman, T. Edgar, L. Lasdon, Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques, Computers & chemical engineering 16 (10-11) (1992) 963–986.
[39] K. D. Rao, J. L. Narayana, An approach for a faster gps tracking extended Kalman filter, Navigation 42 (4) (1995) 619–630.
[40] M. Ge, E. C. Kerrigan, Noise covariance identification for time-varying and nonlinear systems, International Journal of Control 90 (9) (2017) 1903–1915.
[41] M. R. Rajamani, J. B. Rawlings, S. J. Qin, Achieving state estimation equivalence for misassigned disturbances in offset-free model predictive control, AIChE Journal 55 (2) (2009) 396–407.
[42] A. Marchetti, B. Chachuat, D. Bonvin, Modifier-adaptation methodology for real-time optimization, Industrial & engineering chemistry research 48 (13) (2009) 6022–6033.
[43] G. François, D. Bonvin, Use of transient measurements for the optimization of steady-state performance via modifier adaptation, Industrial & Engineering Chemistry Research 53 (13) (2014) 5148–5159.
[44] T. Rodríguez-Blanco, D. Sarabia, J. Pitarch, C. de Prada, Modifier adaptation methodology based on transient and static measurements for rto to cope with structural uncertainty, Computers & Chemical Engineering 106 (2017) 480–500.
[45] K. B. Ariyur, M. Krstic, Real-time optimization by extremum-seeking control, John Wiley & Sons, 2003.
[46] G. François, B. Srinivasan, D. Bonvin, Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty, Journal of Process Control 15 (6) (2005) 701–712.
[47] B. Srinivasan, G. François, D. Bonvin, Comparison of gradient estimation methods for real-time optimization, in: 21st European Symposium on Computer Aided Process Engineering-ESCAPE 21, no. EPFL-CONF-155235, Elsevier, 2011, pp. 607–611.
[48] D. Krishnamoorthy, A. Pavlov, Q. Li, Robust extremum seeking control with application to gas lifted oil wells, IFAC-PapersOnLine 49 (13) (2016) 205–210.
[49] D. Krishnamoorthy, B. Foss, S. Skogestad, Real time optimization under uncertainty - applied to gas lifted wells, Processes 4 (4). doi:10.3390/pr4040052.
[50] D. Krishnamoorthy, B. Foss, S. Skogestad, Gas lift optimization under uncertainty, Computer Aided Chemical Engineering 40 (2017) 1753–1758.

Appendix A. Dynamic model adaptation using extended Kalman filter

Extended Kalman filter for parameter estimation uses an augmented state vector $x' = [x^T, \theta^T]^T \in \mathbb{R}^{n_x+n_\theta}$ constructed using the states and the uncertain variables.

The set of uncertain variables θ is modelled using an integrated noise term $\theta_{k+1} = \theta_k + w_{\theta,k}$ where, $w_{\theta,k} \sim \mathcal{N}(0, Q_\theta)$ is a small artificial noise with zero mean and covariance Q_θ term that allows the Kalman filter to adjust the estimate the of the parameter [19]. The augmented system is then given by

$$x'_{k+1} = \begin{bmatrix} x_{k+1} \\ \theta_{k+1} \end{bmatrix} = f'(x_k, u_k, \theta_k) + w'_k \tag{A.1}$$

$$y_{meas,k} = \begin{bmatrix} h(x_k, u_k) & 0 \end{bmatrix} \begin{bmatrix} x_k \\ \theta_k \end{bmatrix} + v_k$$

where $v_k \sim \mathcal{N}(0, R)$ is the normally distributed measurement noise with zero mean and covariances R and the augmented system model $f'(x_k, u_k, \theta_k)$ is constructed as shown,

$$f'(x_k, u_k, \theta_k) = \begin{bmatrix} f(x_k, u_k, \theta_k) + w_k \\ \theta_k + w_{\theta,k} \end{bmatrix} \tag{A.2}$$

where $w_k \sim \mathcal{N}(0, Q)$ is the normally distributed process noise with zero mean and covariances Q

The discrete-time extended Kalman filter for the augmented system is then given by [19],

$$\hat{x}'_{k|k-1} = f'(\hat{x}_{k-1|k-1}, u_{k-1}, \hat{\theta}_{k-1|k-1}) \tag{A.3a}$$
$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q'_{k-1} \tag{A.3b}$$
$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k + R_k)^{-1} \tag{A.3c}$$
$$\hat{x}'_{k|k} = \hat{x}'_{k|k-1} + K_k(y_{meas,k} - h(\hat{x}_{k|k-1}, u_k)) \tag{A.3d}$$
$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{A.3e}$$

where P is the covariance of the state and parameter estimates, K is the Kalman gain, F and H depicts the linearized system around the current estimate, given by

$$F = \left.\frac{\partial f'(x, u, \theta)}{\partial x'}\right|_{x'=\hat{x}'} \quad H = \left.\frac{\partial h(x, u)}{\partial x'}\right|_{x'=\hat{x}'} \tag{A.4}$$

and the augmented covariance Q' is given by,

$$Q' = \begin{bmatrix} Q & 0 \\ 0 & Q_\theta \end{bmatrix} \tag{A.5}$$

The estimated parameter $\hat{\theta}_k$ is then used in the static optimizer as shown in (8).

Appendix A.1. Augmenting unmodelled effects

Let the unmodelled disturbances be represented as additive disturbance to (1) as shown,

$$x_{k+1} = f(x_k, u_k, \theta_k) + \Delta_k \tag{A.6}$$

and the corresponding static counterpart is expressed as

$$y = f'_{ss}(u, \theta, \Delta) \tag{A.7}$$

The additive disturbance $\Delta_k \in \mathbb{R}^{n_x}$ is expressed as integrating disturbance $\Delta_{k+1} = \Delta_k + \xi_k$ with some small unknown covariance Q_ξ for $\xi_k \sim \mathcal{N}(0, Q_\xi)$. The new augmented state is $\tilde{x} = [x^T, \theta^T, \Delta^T]^T$ and the new augmented system is,

$$\tilde{f}(x_k, u_k, \theta_k, \Delta_k) = \begin{bmatrix} f(x_k, u_k, \theta_k) + w_k \\ \theta_k + w_{\theta,k} \\ \Delta_k + \xi_k \end{bmatrix} \tag{A.8}$$

and the augmented covariance matrix $\tilde{Q}$ is,

$$\tilde{Q} = \begin{bmatrix} Q & 0 & 0 \\ 0 & Q_\theta & 0 \\ 0 & 0 & Q_\xi \end{bmatrix} \tag{A.9}$$

The combined state, parameter and the unmodelled disturbances ($\tilde{x}$) are estimated using an extended Kalman filter by replacing (A.2) and (A.5) with (A.8) and (A.9) in (A.3).

Appendix B. Gas Lift Network model

Production from a cluster of $\mathcal{N} = \{1, \cdots, n_w\}$ gas lifted well can be described using differential and algebraic equations [49],[50]. The dynamics are include in the model due to the mass balances in each well which are described by the following differential equations.

$$\dot{m}_{ga_i} = w_{gl_i} - w_{iv_i} \tag{B.1a}$$
$$\dot{m}_{gt_i} = w_{iv_i} - w_{pg_i} + w_{rg_i} \tag{B.1b}$$
$$\dot{m}_{ot_i} = w_{ro_i} - w_{po_i} \qquad \forall i \in \mathcal{N} \tag{B.1c}$$

where, m_{ga_i} is the mass of gas in the annulus, m_{gt_i} is the mass of gas in the well tubing, m_{ot_i} is the mass of oil in the well tubing, w_{gl_i} is the gas lift injection rate, w_{iv_i} is the gas flow from the annulus into the tubing, w_{pg_i} and w_{po_i} are the produced gas and oil flow rates respectively and, w_{rg_i} and w_{ro_i} are the gas and oil flow rates from the reservoir for each well i. The mass balance in the riser for oil and gas phase is given by,

$$\dot{m}_{gr} = \sum_{i=1}^{n_w} w_{pg_i} - w_{tg} \tag{B.2a}$$
$$\dot{m}_{or} = \sum_{i=1}^{n_w} w_{po_i} - w_{to} \tag{B.2b}$$

where, m_{gr} is the mass of gas in the riser and m_{or} is the mass of oil in the riser and w_{tg} and w_{to} are the total gas and oil flow rates respectively. The densities ρ_{a_i} (density of gas in the annulus in each well) and ρ_{m_i} (fluid mixture density in the tubing for each well) and ρ_r (fluid mixture density in the riser) are given by,

$$\rho_{a_i} = \frac{M_w p_{a_i}}{T_{a_i} R} \tag{B.3a}$$
$$\rho_{w_i} = \frac{m_{gt_i} + m_{ot_i} - \rho_o L_{bh_i} A_{bh_i}}{L_{w_i} A_{w_i}} \tag{B.3b}$$
$$\rho_r = \frac{m_{gr} + m_{or}}{L_r A_r} \qquad \forall i \in \mathcal{N} \tag{B.3c}$$

where M_w is the molecular weight of the gas, R is the gas constant, T_{a_i} is the temperature in the annulus in each well, ρ_o is the density of oil in the reservoir, L_{bh_i} and L_{w_i} are the lengths of each well above and below the injection point respectively and A_{bh_i} and A_{w_i} are the cross-sectional area of each well above and below the injection point respectively. L_r and A_r are the length and the cross sectional area of the riser manifold. The annulus pressure p_{a_i}, wellhead pressure p_{wh_i}, well injection point pressure p_{iv_i} and

the bottom hole pressure p_{bh_i} for each well are given by,

$$p_{a_i} = \left(\frac{T_{a_i}R}{V_{a_i}M_w} + \frac{gL_{a_i}}{L_{a_i}A_{a_i}}\right) m_{ga_i} \tag{B.4a}$$

$$p_{wh_i} = \frac{T_{w_i}R}{M_w}\left(\frac{m_{gt_i}}{L_{w_i}A_{w_i} + L_{bh_i}A_{bh_i} - \frac{m_{ot_i}}{\rho_o}}\right) - \frac{1}{2}\left(\frac{m_{gt}+m_{ot}}{L_wA_w}gH_w\right) \tag{B.4b}$$

$$p_{wi_i} = p_{wh_i} + \frac{g}{L_{w_i}A_{w_i}}(m_{ot_i} + m_{gt_i} - \rho_o L_{bh_i}A_{bh_i})H_{w_i} + \Delta p^t_{fric} \tag{B.4c}$$

$$p_{bh_i} = p_{wi_i} + \rho_{w_i}gH_{bh_i} + \Delta p^{bh}_{fric} \qquad \forall i \in \mathcal{N} \tag{B.4d}$$

where L_{a_i} and A_{a_i} are the length and cross sectional area of each annulus, T_{w_i} is the temperature in each well tubing, H_{r_i} and H_{w_i} are the vertical height of each well tubing below and above the injection point respectively and g is the acceleration of gravity constant. Δp^t_{fric} and Δp^{bh}_{fric} represents the frictional pressure drop in the well tubing above and below the gas injection point respectively. The manifold pressure p_m and the riser head pressure p_{rh} are given by,

$$p_{rh} = \frac{T_rR}{M_w}\left(\frac{m_{gr}}{L_rA_r}\right) \tag{B.5a}$$

$$p_m = p_{rh} + \rho_r g H_r + \Delta p^r_{fric} \tag{B.5b}$$

where T_r is the average temperature in the riser, H_r is the vertical height of the riser and Δp^r_{fric} is the frictional pressure drop in the riser. The flow through the downhole gas lift injection valve w_{iv_i}, total flow through the production choke w_{pc_i}, produced gas and oil flow rate, and the reservoir oil and gas flow rates are given by,

$$w_{iv_i} = C_{iv_i}\sqrt{max(0, \rho_{a_i}(p_{a_i} - p_{wi_i}))} \tag{B.6a}$$

$$w_{pc_i} = C_{pc_i}\sqrt{max(0, \rho_{w_i}(p_{wh_i} - p_m))} \tag{B.6b}$$

$$w_{pg_i} = \frac{m_{gt_i}}{m_{gt_i} + m_{ot_i}}w_{pc_i} \tag{B.6c}$$

$$w_{po_i} = \frac{m_{ot_i}}{m_{gt_i} + m_{ot_i}}w_{pc_i} \tag{B.6d}$$

$$w_{ro_i} = PI_i(p_{r_i} - p_{bh_i}) \tag{B.6e}$$

$$w_{rg_i} = GOR_i \cdot w_{ro_i} \qquad \forall i \in \mathcal{N} \tag{B.6f}$$

where, C_{iv_i} and C_{pc_i} are the valve flow coefficients for the downhole injection valve and the the production choke for each well respectively, PI_i is the reservoir productivity index, p_{r_i} is the reservoir pressure and GOR_i is the gas-oil ratio for each well. The two wells produce to a common manifold, where the manifold pressure is denoted by p_m and the flow rates from the two well mixes together. The total flow through the riser head choke w_{rh}, the total produced oil and gas rates are then given by,

$$w_{rh} = C_{rh}\sqrt{\rho_r(p_{rh} - p_s)} \tag{B.7a}$$

$$w_{tg} = \frac{m_{gr}}{m_{gr} + m_{or}}w_{rh} \tag{B.7b}$$

$$w_{to} = \frac{m_{or}}{m_{gr} + m_{or}}w_{rh} \tag{B.7c}$$

where C_{rh} is the valve flow coefficient for the riser head valve and p_s is the separator pressure, which is assumed to be held at a constant value.

As seen from (B.1a) - (B.7c), the gas lifted well is modelled as a semi-explicit index-1 DAE system of the form

$$\dot{x} = \mathbf{F}_c(x, z, u, \theta) \tag{B.8a}$$

$$\mathbf{G}(x, z, u, \theta) = 0 \tag{B.8b}$$

where $\mathbf{F}_c(x, z, u, \theta)$ is the set of differential equations (B.1a) - (B.2b) and $\mathbf{G}(x, z, u, \theta)$ is the set of algebraic equations (B.3a) - (B.7c), $x \in \mathbb{R}^8$ are the set of differential variables, $z \in \mathbb{R}^{30}$ are the set of algebraic variables, $u \in \mathbb{R}^2$ are the set of control inputs and $\theta \in \mathbb{R}^2$ are the set of uncertain parameters.

Table B.3: List of well parameters and their corresponding values used in the results.

Parameter	**units**	**Well 1**	**Well 2**
L_w	[m]	1500	1500
H_w	[m]	1000	1000
D_w	[m]	0.121	0.121
L_{bh}	[m]	500	500
H_{bh}	[m]	500	500
D_{bh}	[m]	0.121	0.121
L_a	[m]	1500	1500
H_a	[m]	1000	1000
D_a	[m]	0.189	0.189
ρ_o	[kg m^{-3}]	800	800
C_iv	[m^2]	0.1E-3	0.1E-3
C_pc	[m^2]	2E-3	2E-3
p_r	[bar]	150	155
PI	[kg s^{-1} bar^{-1}]	0.7	0.7
T_a	[°C]	28	28
T_w	[°C]	32	32
GOR	[kg/kg]	0.1±0.05	0.12±0.02

Table B.4: List of riser parameters and their corresponding values used in the results.

Parameter	**units**	**Riser**
L_r	[m]	500
H_r	[m]	500
D_r	[m]	0.121
$C_r h$	[m^2]	10E-3
p_s	[bar]	20
T_r	[°C]	30
M_w	[g mol^{-1}]	20
R	[J mol^{-1} K^{-1}]	8.314