Journal of Process Control 138 (2024) 103208

Contents lists available at ScienceDirect

Journal of Process Control

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

Primal–dual feedback-optimizing control with override for real-time optimization☆

Risvan Dirza, Sigurd Skogestad *

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

ARTICLE INFO

Keywords:
Production optimization
Feedback-optimizing control
Real-time optimization
Override control

ABSTRACT

Primal–dual feedback-optimizing control is a simple yet powerful approach to optimally handle active constraint changes at steady state. It is composed of two layers: Constraint control in the upper master layer and unconstrained optimization or gradient control in the layer below. The master constraint controllers operate on a slow time scale by updating the dual variables (Lagrange multipliers). This can result in too slow control of the constraints, for example, for hard constraints that cannot be violated dynamically. To address this issue, we propose introducing a third fast override constraint control layer. Additionally, to optimally coordinate the constraint handling between the master and override layers, we need to introduce *auxiliary* constraints for the master controllers. A gas-lift oil production optimization case study demonstrates the power of the proposed scheme.

1. Introduction

Traditional real-time optimization (RTO) solves online a steady-state optimization problem on a slow timescale (typically every hour) and implements the optimization results as setpoints to the control layer [1,2]. Despite the economic benefits, RTO is less used in practice than one might expect [3], mainly because of the cost of developing and updating the process model and partly because of too infrequent updates. Consequently, the full potential of RTO remains unexploited in process industries [2].

One obvious solution to increase the update frequency (and reduce dynamic constraint violations), is to use dynamic optimization, such as dynamic RTO or economic nonlinear model predictive control (ENMPC). Although extensively studied in research papers, recent papers [4,5] highlight numerical challenges hindering widespread adoption of dynamic optimization. Additionally, many process industries require deploying automatic tools on embedded platforms such as programmable logic controllers (PLC), which is currently unsuitable for solving nonlinear optimization online [6].

An alternative approach to significantly increase the update frequency, which is much easier to implement, is to move the optimization into the control layer. This approach is known as *feedback-optimizing control* and was introduced by Morari et al. [7]. In this paper, we study primal–dual feedback-optimizing control and the main novelty is to include constraint override to reduce the constraint backoff. One important advantage of feedback-optimizing control is the possibility to decompose the optimization problem and have different closed-loop response times for each process unit. This can be particularly advantageous in large-scale industrial processes where process dynamics, measurement delays, and other factors may vary across different units.

Lagrangian relaxation is a useful technique for transforming a constrained RTO problem into an unconstrained optimization problem [8]. In dual decomposition, the Lagrange multipliers (λ), also known as dual variables, may be used as manipulated variables to control the constraints on a slow time scale [3]. In practice, this may be realized using a single-loop PID controller with a selector for each constraint [9]. This corresponds to the diagonal master controller in Fig. 1. The integral action in the controller permits asymptotically optimal operation, and the selector automatically switches between active constraint regions. This avoids the need to identify a new control structure in each active constraint region.

In the faster layer below (see Fig. 1), in order to satisfy the optimality condition $\nabla_u \mathcal{L} = 0$ [10], we may control the estimated Lagrangian cost gradient to zero ($\nabla_u \hat{\mathcal{L}} = 0$) using the physical manipulated variables u (primal decision variables). Alternatively, especially if the local optimization problems are coupled, one may instead use a numerical solver.

The strategy in Fig. 1 is known as primal–dual or simply dual feedback-optimizing control, and variants have been studied by several

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

* Corresponding author.
E-mail addresses: risvan.dirza@ntnu.no, risvan.dirza@umsu.ac.id (R. Dirza), skoge@ntnu.no (S. Skogestad).

https://doi.org/10.1016/j.jprocont.2024.103208
Received 16 December 2023; Received in revised form 9 March 2024; Accepted 22 March 2024
Available online 8 April 2024

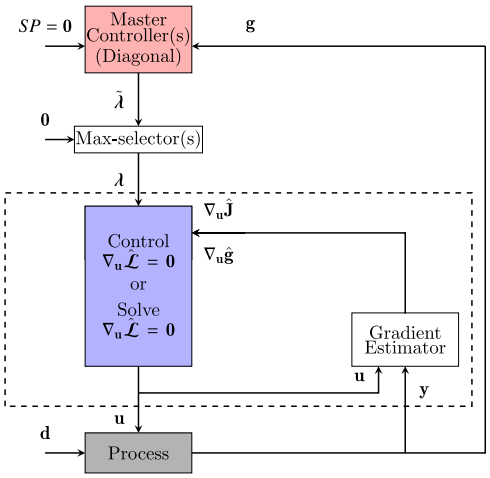


Fig. 1. Primal–dual feedback-optimizing control scheme with a scalar master controller (typically, PID) for each constraint, followed by a max-selector [9]. For equality constraints, the max-selector is omitted. The large dashed box represents the faster lower layer, where the objective is to drive the Lagrangian cost gradient ($\nabla_u \mathcal{L} = \nabla_u \mathbf{J} + \lambda^\top \nabla_u \mathbf{g}$) to zero.

researchers [3,9,11–13]. The effectiveness of this strategy has been validated experimentally on a lab-scale rig [14]. However, the optimality concerning constraint satisfaction is only asymptotic (at steady state) and because the constraints are handled by the slow master controller, there may be significant dynamic violations.

If the dynamic violations are not acceptable, one needs to introduce a *back-off* from the constraint value for the setpoint in the master controller, which will result in an economic loss. In particular, this applies to *hard* constraints, such as those related to safety and the environment, where no dynamic violations are acceptable.

Therefore, there is a need for *an effective way to improve the dynamic handling of constraints within the primal–dual feedback-optimizing control framework*. To address this challenge, we propose incorporating override constraint control on a fast time scale. The paper serves as an extended version of our previous studies, where we provided a short introduction to primal–dual with override [15], a simulation case study for multi-input override [16], and override pairing procedure for minimum economic loss in the case of a non-performing estimator [17].

The paper is organized as follows. In Section 2, we study the existing primal–dual strategy and describe its limitations. To address the problem with constraint violations, Section 3 introduces override constraint controllers. To get consistency with the constraint control in the master controller, it is necessary to introduce auxiliary constraints. It is also shown how to use multiple inputs for override control, which is important in many practical cases. The gas lift optimization case study is considered in Section 4. In Section 5, some alternative and extended approaches are discussed, before concluding the paper is Section 6.

2. Problem formulation

Consider a general steady-state optimization problem:

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

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

Here, $\mathbf{u} \in \mathbb{R}^{n_\mathbf{u}}$ represents the decision variables, $n_\mathbf{u}$ is the number of decision variables, $\mathbf{d} \in \mathbb{R}^{n_\mathbf{d}}$ is the disturbances, and $n_\mathbf{d}$ is the number of disturbances. The scalar function $J : \mathbb{R}^{n_\mathbf{u}} \times \mathbb{R}^{n_\mathbf{d}} \to \mathbb{R}$ is the cost or objective function. Similarly, the function $\mathbf{g} : \mathbb{R}^{n_\mathbf{u}} \times \mathbb{R}^{n_\mathbf{d}} \to \mathbb{R}^{n_\mathbf{g}}$ represents the constraints, and $n_\mathbf{g}$ is the number of constraints.

For simplicity, we have not included the state variable $\mathbf{x}$ in problem (1); that is, any states have been formally eliminated by making use of the steady-state model equations.

The Lagrange function associated with problem (1) is

$$\mathcal{L}(\mathbf{u}, \mathbf{d}, \lambda) = J(\mathbf{u}, \mathbf{d}) + \sum_{j=1}^{n_\mathbf{g}} \lambda_j g_j(\mathbf{u}, \mathbf{d}) \tag{2}$$

where $\lambda \in \mathbb{R}^{n_\mathbf{g}}$ denotes the dual variables (Lagrange multipliers) related to the constraints (1b). The necessary conditions of optimality, known as the Karush–Kuhn–Tucker (KKT) conditions, can then be expressed for the problem (1) as follows [18]:

$$\nabla_{u_i} \mathcal{L}(\mathbf{u}, \mathbf{d}, \lambda) = 0, \text{ for all } i = 1, \dots, n_\mathbf{u} \tag{3a}$$

$$g_j(\mathbf{u}, \mathbf{d}) \le 0, \text{ for all } j = 1, \dots, n_\mathbf{g} \tag{3b}$$

$$\lambda_j \ge 0, \text{ for all } j = 1, \dots, n_\mathbf{g} \tag{3c}$$

$$\lambda_j g_j(\mathbf{u}, \mathbf{d}) = 0, \text{ for all } j = 1, \dots, n_\mathbf{g} \tag{3d}$$

Here, (3a) is the stationary condition, (3b) is the primal feasibility, (3c) is the dual feasibility, and (3d) is the complementary slackness condition.

Dual decomposition involves iteratively solving this set of equations, where the last three equations related to the constraints are used to compute the dual variables (λ), and the stationary gradient conditions (3a) are used to compute the primal variables ($\mathbf{u}$). Arrow et al. [19] proved convergence of dual decomposition for the case with equality constraints.

2.1. Primal–dual feedback-optimizing control

Recently, Krishnamoorthy [13] showed how dual decomposition can be used as a basis for solving the optimization problem in (3a)–(3d) using feedback control. He used master controllers for constraints and local gradient controllers. To enable the use local gradient controllers, he assumed decomposed subsystems (where the cost can be decomposed for each subsystem as $J(\mathbf{u}, \mathbf{d}) = J_1(u_1, \mathbf{d}) + J(u_2, \mathbf{d}) + \cdots$).

Dirza et al. [9] extended this feedback-optimizing control approach to include inequality constraints. This requires introducing a max-selector for the dual variable (λ) to satisfy conditions (3b)–(3d). Fig. 1 illustrates the final solution.

Note that in Fig. 1, we have indicated the possible use of a solver, rather than feedback control [9], for driving the gradient $\nabla_\mathbf{u} \mathcal{L}$ to zero. The use of a solver requires that we have a model for how the gradients $\nabla_\mathbf{u} J$ and $\nabla_\mathbf{u} g$ depend on the inputs u, so it is not relevant in situations where these are just signals. One reason for allowing for the use of a solver, is that the focus in the present paper is on override constraint control, so we do not want to restrict the treatment. Note here that decentralized feedback control (PID) is most effective for cases with decomposable subsystems. Also, note that the use of feedback control, may be viewed as a trick to numerically solve the equations, see Fig. 2.

2.2. Master controller

The *master controller* is decentralized and consists on n_g SISO controllers (usually PID). The integral action in the j'th controller drives the active constraint $g_j(\mathbf{u}, \mathbf{d})$ (the controlled variable) to zero at steady state using $\tilde{\lambda}_j$ as the manipulated variable (MV). For equality constraints (which are always active), λ_j can take on also negative values and a max-selector is not needed. For inequality constraints, we require $\lambda_j \ge 0$ in (3c), which may be realized using a "max" selector with a minimum value of $\lambda_j = 0$.

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

This selector will deselect the constraint controller when the constraint becomes inactive.

The master controller is a decentralized (diagonal) controller, which generally requires a choice of pairings. However, in this case there is only one possibility: To satisfy the complementary slackness condition (3d), which states that at steady state either $g_j = 0$ or $\lambda_j = 0$, we need to pair the measured constraint g_j (CV or input for the master controller) with the corresponding dual variable λ_j (MV or output for the master controller):

$$\tilde{\lambda}_j \leftrightarrow g_j \tag{5}$$

Typically, the master controllers are PID-controllers, and often simply integral controllers ($c(s) = K_i/s$). Because the controllers may be deselected by the selector, it is necessary to use anti-windup on the integral action.

Note that we control the *measured* constraints, which means that we do not control the constraint via a model, as is done with normal RTO. This is an important benefit of primal–dual feedback-optimizing control.

2.3. Gradient solvers/controllers

We next consider satisfying the optimality (stationary) conditions

$$\nabla_{u_i} \mathcal{L}(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda}) = 0 \tag{6}$$

where

$$\nabla_{u_i} \mathcal{L}(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda}) = \nabla_{u_i} \mathbf{J}(\mathbf{u}, \mathbf{d}) + \sum_{j=1}^{n_\mathbf{g}} \lambda_j \nabla_{u_i} \mathbf{g}_j(\mathbf{u}, \mathbf{d}) \tag{7}$$

There are two main approaches for solving (6).

2.3.1. Equation solver

The most obvious approach is to solve (6) directly for the primal variables (inputs) $\mathbf{u}$ using the model equations, similar to traditional steady-state RTO. This requires that we have the equations for how the estimated gradients depend on the inputs u, and even if we have this, a numerical solution may be required. Also note that a direct implementation of the inputs (primal variables) $\mathbf{u}$ into the process may result in instability (because the estimator might not immediately provide accurate state/parameter estimation), so in most cases, a lower layer controller, such as a first-order filter or a setpoint controller, would need to be added (not shown in Fig. 1).

2.3.2. Feedback control

An alternative approach is to use feedback control as a trick to solve (6). Fig. 2 illustrates the idea. In this case, a filter is not needed because of the tuneable dynamics in the gradient controllers.

The gradient controller may be one "big" controller (e.g., MPC), but an important benefit of feedback control is that it allows for easily decomposing the system by using decentralized control. An obvious choice, is to use single-loop control and pair the gradient $\nabla_{u_i} \mathcal{L}(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda})$ (CV for the controller) with its corresponding primal variable u_i (MV for the controller):

$$u_i \leftrightarrow \nabla_{u_i} \mathcal{L}(\mathbf{u}, \mathbf{d}, \boldsymbol{\lambda}) \tag{8}$$

A main problem here is that the set of equations is generally coupled, so the use of individual gradient controllers (e.g., PID) may not work well. In such cases, it may be possible to decompose the system into multivariable subsystems (each with several inputs u_i), and use multivariable gradient controllers (e.g., MPC). To check for the degree of interactions, and also for tuning purposes, one may consider the square linearized transfer matrix $\mathbf{G}(s)$ from the inputs u to the controlled variables (gradients $\mathbf{c} = \nabla_\mathbf{u} \mathcal{L}$)

$$\mathbf{c} = \mathbf{G}(s)\, \mathbf{u} \tag{9}$$

However, note that $\mathbf{G}(s)$ will change depending on the values of λ_j, which depends on the disturbances. In particular, there will be non-smooth changes in $\mathbf{G}(s)$ when there are changes in the active constraints

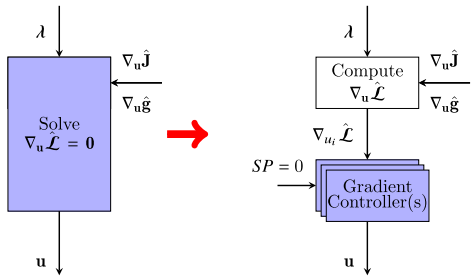


Fig. 2. Using feedback control as a *trick* to solve $\nabla_u \mathcal{L} = 0$ for cases with decoupled subsystems. The gradient controller for each subsystem is typically PID, but it may be MPC for a multivariable subsystem.

(when λ_j changes to or from zero). In summary, the use of feedback with gradient controllers is most effective for *non-interactive* or *weakly interactive* subsystems. An important potential advantage of feedback control is that it allows for using different closed-loop time constants for each subsystem.

2.4. Limitation of primal–dual approach in Fig. 1

In Fig. 1, the constraints are controlled by the master controller on a slow time scale, which is undesirable for many constraints. When a constraint g_j is encountered, the constraint will be violated (i.e., $g_j > 0$) for some time until the master constraint controller has the time to increase the dual variable λ_j to its new optimal value, corresponding to satisfying the constraint. Because the master controller is slow, this violation can be considerable.

For some constraints, dynamic violations are allowed, as only the average (steady-state) value matters. For other constraints, we need to reduce (or completely avoid in the case of hard constraints) the constraint violation. The simplest approach is to introduce a "back-off" b_j on the setpoint (SP) for the constraint j. That is, rather than using $SP_j = 0$ in Fig. 1, we choose

$$SP_j = -|b_j| < 0$$

This will lead to an economic loss (e.g., [20])

$$L = \Sigma_j |\lambda_j b_j| \tag{10}$$

where j is an active constraints, λ_j is the corresponding Lagrange multiplier and L [$/s] is the economic loss caused by constraint back-off. The loss in (10) is exact for a small back-off, but it is generally a reasonable approximation.

The best way to reduce this loss is by tightening the control of the constraint, allowing for a reduction in $|b_j|$. For this reason, we propose adding override constraint control (Fig. 3) for "critical" constraints, which are here defined as constraints where it is important reduce or avoid constraint violation, This is discussed in more detail in the next section. In addition to the override control, we need to introduce auxiliary constraints for use by the master controllers, so that we can release constraints when it is economically optimal.

3. Proposed control scheme

3.1. Override constraint controllers

To reduce constraint violation (and thus minimize the need for back-off) for a "critical" constraint g_j, we propose to pair it with a selected manipulated input (primal variable) u_i and control it using a fast override controller. This is implemented at the bottom of the control hierarchy, as illustrated in Fig. 3. At any given time, the selector (`max`- or `min`- selector depending on the case) will choose the process input u_i as either the output u_i^g from the override controller or the output $\tilde{u}_i$ from the gradient controller (or equation solver).

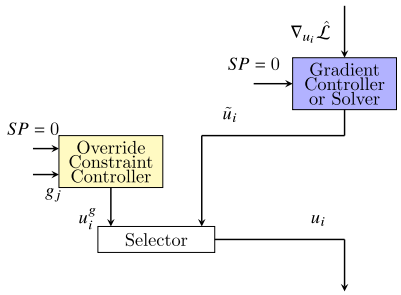


Fig. 3. Proposed override control for a "critical" constraint g_j.

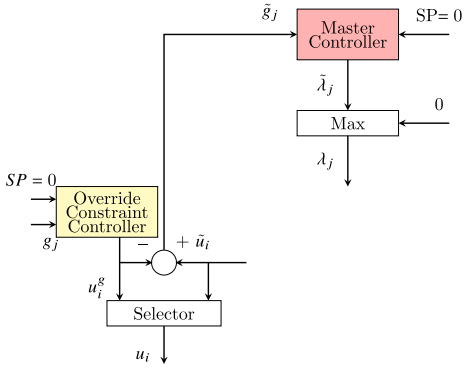


Fig. 4. Proposed override scheme for primal–dual feedback-optimizing control scheme. For override constraints g_j, the master controller controls the auxiliary constraint $\tilde{g}_j = \tilde{u}_i - u_i^g$. The sign of the gain in the master controller depends on whether there is a "min" or "max" selector at the bottom of the hierarchy.

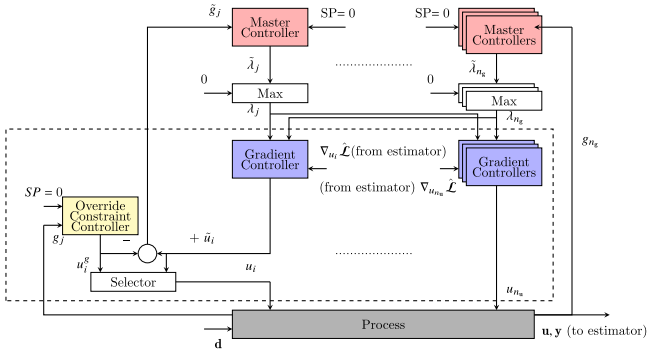


Fig. 5. Complete structure of the proposed scheme with gradient controllers and one override controller with its associated auxiliary constraint $\tilde{g}_j$.

Choice of override pairings. It is important to find a good pairing between the constraint g_j and the selected input u_i:

$$u_i \leftrightarrow g_j$$

For choosing pairings and tuning purposes, one may consider the linear transfer matrix $\mathbf{G^g}(s)$ from the inputs to the override constraints:

$$\mathbf{g} = \mathbf{G^g}(s)\ \mathbf{u} \tag{11}$$

The following pairing rules are useful [21].

1. "Pair close rule": Select an input u_i with a large and direct effect on the constraint g_j [22]. For example, if the element $G^g_{ij}(s)$ from u_i to g_j is approximated as a first-order plus delay transfer function, then prefer a pairing with a large gain, a small delay, and a small time constant. A common way of obtaining such a relationship between the input u_i and the constraint g_j is a step response experiment (using a dynamic model or the actual process).
2. "Input saturation rule": Select an input u_i that is not likely to saturate, for example, at a fully open or closed valve.

Choice of `max` *- or* `min`*-selector.* The selector rules given in [20] says that, if the constraint g_j is satisfied by a large input u_i, then use a `max`-selector:

$$u_i = \texttt{max}\{\tilde{u}_i, u_i^g\}$$

Conversely, if the constraint g_j is satisfied by a small input u_i, then use a `min`-selector:

$$u_i = \texttt{min}\{\tilde{u}_i, u_i^g\}$$

Tuning of override constraint controller. The name "override" is appropriate because we aim to make short-term corrections u_i^g to the steady-state optimal solution $\tilde{u}_i$, with the goal of avoiding undesired dynamic constraint violations (see Fig. 3). However, on a longer time scale, the steady-state optimization should take over and provide the optimal value of the input, $u_i = \tilde{u}_i$, and also decide whether or not a constraint is active. This has implications for tuning (i.e., if the slower/outer layer is not slow enough, the closed-loop system may eventually lead to instability), and to prevent the override controller from interfering with the steady-state optimization, we need a time scale separation between the fast override controller and its slower gradient controller, typically in the range 4 (minimum) to 10 (desired) [21].

Maximum number of override constraints. In most cases, each override constraint is paired with a different input, meaning that the number of override constraints ($n_{g,o}$) cannot exceed the number of inputs, i.e., $n_{\mathbf{g,o}} \leq n_{\mathbf{u}}$. However, if two override constraints cannot be active simultaneously (for example, a variable with both an upper and lower bound), it may be possible to pair two override constraints with one input.

3.2. Auxiliary constraints

Assigning both the master and override constraint controllers (Fig. 5) to control the same constraint g_j may seem like a viable solution, but it fails to function as desired. The problem is that once the override takes over the constraint, the master controller will no longer update λ_i, and thus it will not release the constraint even when it no longer should be active.

To avoid this problem, we propose for override constraints to replace, in the master controller, the original constraint g_j with an auxiliary constraint $\tilde{g}_j$, which is the difference between the process input computed by the gradient controller and the process input computed by the override constraint controller:

$$\tilde{g}_j = \tilde{u}_i - u_i^g \tag{12}$$

The proposed solution is shown in Fig. 4. The complete block diagram, which includes the gradient controller, is shown in Fig. 5.

For the master controller, the auxiliary constraint is $\tilde{g}_j < 0$ if the constraint g_j is satisfied by a small input u_i (and thus we have a min-selector for the override), and $-\tilde{g}_j < 0$ if the constraint g_j is satisfied by a large input u_i (and thus we have a max-selector for the override).

To better understand the use of auxiliary constraints, consider the two directions of constraint switching:

- If a constraint g_j is originally *not* active and a disturbance causes g_j to be violated, the override controller will change u_i^g until the selector assigns $u_i = u_i^g$. At this point, we get a violation (nonzero value) of the auxiliary constraint, and the master controller will increase (slowly) the associated dual variable λ_j (which is zero in the unconstrained case), until we achieve $\tilde{g}_j = 0$, where

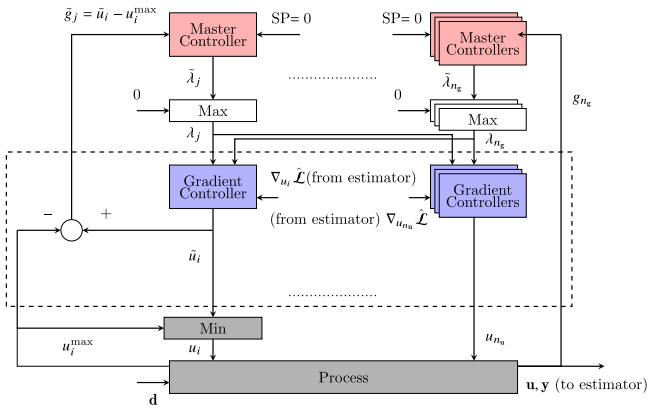


Fig. 6. Proposed use of auxiliary constraint for case with maximum input constraint on u_i. Note that the physics provide an implicit override, so the grey Min-selector at the bottom is not part of the control system, but represents the physical input saturation (valve).

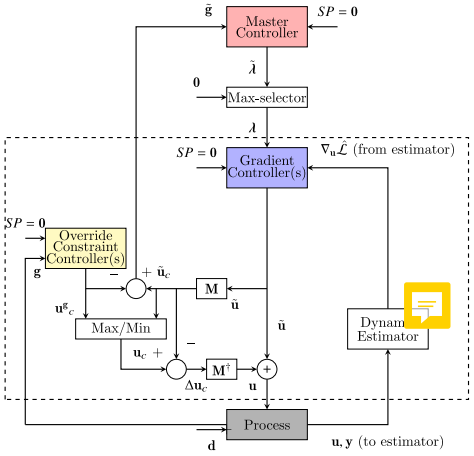


Fig. 7. Proposed override constraint control with multiple inputs [16].

the override and gradient controllers agree on the value for the process input.

- If a constraint g_j is originally active, and a disturbance causes this to no longer be optimal, the master controller will decrease λ_j, which again changes $u_i = \tilde{u}_i$ (because the override controller is no longer active), until we get $\lambda_j = 0$ where the constraint is no longer controlled.

3.3. Implicit override for input constraints

All physical process inputs have upper and lower constraints (also know as saturation limits)

$$u_i \le u_i^{max}$$

$$u_i \ge u_i^{min}$$

or equivalently,

$$g_j^{max} = u_i - u_i^{max} \le 0 \tag{13a}$$

$$g_j^{min} = u_i^{min} - u_i \le 0 \tag{13b}$$

These input constraints are always "hard" (and thus "critical") because they cannot be physically violated. Thus, physics provide an implicit override, and an override controller is not necessary for input constraints. However, because there is an implicit override, we still need to use auxiliary constraints for the master controller (Fig. 6):

$$\tilde{g}_j^{max} = \tilde{u}_i - u_i^{max} \le 0 \tag{14a}$$

$$\tilde{g}_j^{min} = u_i^{min} - \tilde{u}_i \le 0 \tag{14b}$$

If the master controller instead were to control the (physical) constraint (u_i^{max} or u_i^{min}), then it would not be possible to exit an input constraint when it is no longer optimally active.

3.4. Override using multiple inputs

So far, we have for simplicity assumed that an override constraint g_j is paired with a single selected input (u_i). This is not a good solution in some cases, because it may lead to (dynamic) violation of the corresponding input constraint (u_i^{min} or u_i^{max}), which means that we may (at least temporarily) lose control of the constraint g_j, which may be unacceptable. For example, assume that we have many users of a scarce common resource (say, steam in a large chemical plant), then we should not let only one user take care of the balancing.

The obvious solution is to pair the override constraint g_j with a combined input u_c,

$$u_c \leftrightarrow g_j$$

The simplest is to use a linear input combination

$$u_c = \mathbf{M}\mathbf{u} = \sum_k m_k u_k = m_1 u_1 + m_2 u_2 + \cdots \tag{15}$$

where $\mathbf{M}$ is a constant vector for each override constraint g_j. Here, selecting all $m_k = 0$ except for $m_i = 1$ gives the choice $u_c = u_i$ studied so far. Many concerns may determine the selection of $\mathbf{M}$. If all the inputs u_k have been scaled in the same way (say from $u_k^{min} = 0$ to $u_k^{max} = 1$), then a reasonable choice for $\mathbf{M}$ may be $\mathbf{M} = \mathbf{G}^{g_j}$ where $\mathbf{G}^{g_j}$ is gain matrix from the inputs $\mathbf{u}$ to the override constraint g_j (at approximately the frequency corresponding to the closed-loop time constant of the override controller). It is also possible to make other choices for $\mathbf{M}$, including nonlinear combinations, for example, based on the ratio between how the subsystems use the common resource g.

An implementation for a single override constraint g with multiple inputs is shown in Fig. 7. We here use the pseudoinverse (right inverse) of $\mathbf{M}$ when computing the contribution of the override action on the inputs u. The use of the pseudoinverse minimizes the two-norm (sum of squares) of the individual input changes for a given value of u_c (e.g., see (A.65) in [22]). The block diagram in Fig. 7 may seem a bit complicated, but note that we get

$$u = \mathbf{M}^{\dagger} u_c + (I - \mathbf{M}^{\dagger}\mathbf{M})\tilde{u}$$

where $\mathbf{M}^{\dagger}$ is the pseudoinverse of $\mathbf{M}$. When the override is not active, we have $u_c = \mathbf{M}\tilde{u}$ and we get as expected, $u = \tilde{u}$. Also note that the auxiliary constraint is $\tilde{g} = \mathbf{M}\tilde{u} - u_c^g$. The use of $\mathbf{M}$ also works in the scalar case. For example, with $\mathbf{M} = [1\ 0\ 0]$ (pairing the override constraint with input 1), we get as desired

$$\mathbf{M}^{\dagger} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad (I - \mathbf{M}^{\dagger}\mathbf{M}) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

4. Case study: Gas lift optimization

This section presents a case study to illustrate the effectiveness of the primal–dual feedback-optimizing control structure, both with and without override constraint control. The case study involves a gas-lift oil production system with four oil production wells (see Fig. 8). The system includes four gas lift valves (MVs), a manifold, a riser, and a separator which gives the export gas and produced (export) oil. This system is a variant of a system that has been studied and used in [9,23–25]. The model description, and its parameters are available as supplementary materials.

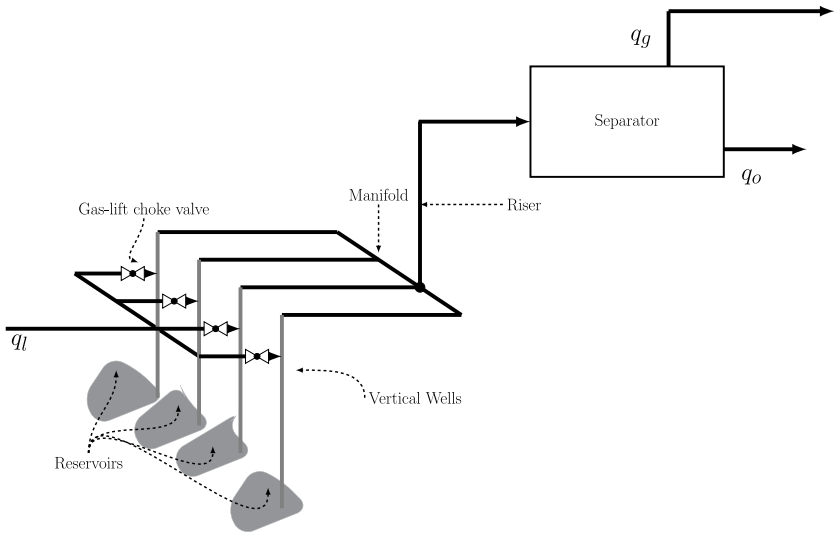


Fig. 8. Flowsheet for gas lift case study.

4.1. Problem description

The system is described by a set of coupled differential–algebraic equations (DAEs). The detailed model is available in supplementary material. There are four manipulated variables (MVs, primal variables, inputs), which are the openings of the four gas-lift valves:

$$\mathbf{u} = \begin{bmatrix} u_1 & \dots & u_4 \end{bmatrix}^\top$$

The system is weakly interactive because a change in one gas lift valve opening impacts, through the effect on pressure, the gas lift and oil and gas flow rates of all wells. The considered disturbances are the gas-oil ratio in the inflow from the reservoir for well 2 and the maximum gas export q_g^{max}:

$$\mathbf{d} = \begin{bmatrix} GOR_2 & q_g^{max} \end{bmatrix}^\top$$

The main objective is to maximize the total oil production q_o. At steady state we have

$$q_o = q_{o,1} + q_{o,2} + q_{o,3} + q_{o,4}$$

where $q_{o,i}$ is the produced oil from reservoir i. However, at the same time, one should try to minimize the total gas lift supply,

$$q_l = q_{l,1} + q_{l,2} + q_{l,3} + q_{l,4}$$

The maximum constraint on the total export gas, $q_g \le q_g^{max}$, provides a coupling constraint. At steady state, we have

$$q_g = q_{g,1} + q_{g,2} + q_{g,3} + q_{g,4} + q_l$$

where $q_{g,i}$ is the produced gas from reservoir i.

Any excess gas, $g_1 = q_g - q_g^{max}$, must be flared (burned), which is strongly undesirable for environmental reasons. Therefore, override control is used for this coupling constraint (denoted g_1). In addition to the coupling constraint, each gas-lift choke u_i has a physical constraint with a maximum valve opening of 1, which provide implicit overrides. In summary, the steady-state optimization problem can be expressed as follows:

$$\min_{\mathbf{u}} \quad J = -p_o q_o + p_l q_l \tag{16a}$$

$$\text{s.t.} \quad g_1 = q_g - q_g^{max} \le 0, \tag{16b}$$

$$g_i^{u_i^{\max}} = u_i - u_i^{max} \le 0, \ i = 1, \dots, 4 \tag{16c}$$

Here, $u_i^{max} = 1$ is the maximum opening of gas lift valve i, p_o is the price of oil, p_l is the cost of gas lift supply, and q_g^{max} is the maximum gas export (see supplementary material for data).

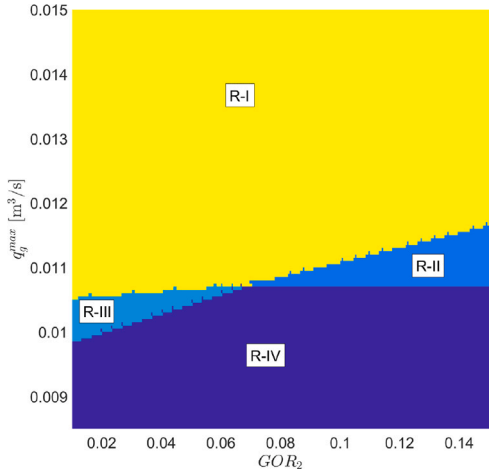


Fig. 9. Optimal active constraint regions as a function of the two disturbances, q_g^{max}, and GOR_2. The four possible active constraint regions are labeled R-I to R-IV: R-I is a region where g_1, g_2, and g_3 are active, R-II is a region where g_1 and g_2 are active, R-III is a region where g_1 and g_3 are active, and R-IV is a region where only the coupling constraint g_1 (max. gas handling capacity) is active.

4.2. Active constraint regions

For this particular problem, we have five constraints, $n_\mathbf{g} = 5$ (one coupling constraint and four max-constraints on the inputs), and therefore a maximum of $2^{n_\mathbf{g}} = 32$ active constraint regions is possible [26]. However, only a subset of these regions are encountered in practice. To illustrate this, consider a scenario where q_g^{max} varies between 0.0085 and 0.0150 m^3/s (equivalent to between 734.40 and 1296.00 m^3/day) and GOR_2 varies between 0.01 and 0.15 m^3/m^3, while the other GOR_i (for $i = 1, 3, 4$) are constant. For these two disturbances, Fig. 9 shows the 4 possible active constraint regions for optimal operation.

Actually, for our purposes the details are not important, because it does not matter for our proposed method how many regions we may encounter and which constraints transitions may occur. This is because the primal–dual feedback optimizing scheme can handle any number of regions and transitions between them. However, to guarantee that we can optimally implement override control for g_1, we must require that $n_u \ge n_{g,o}$, where $n_{g,o}$ is the number of override constraints. This condition is not satisfied in our case since the four maximum gas-lift

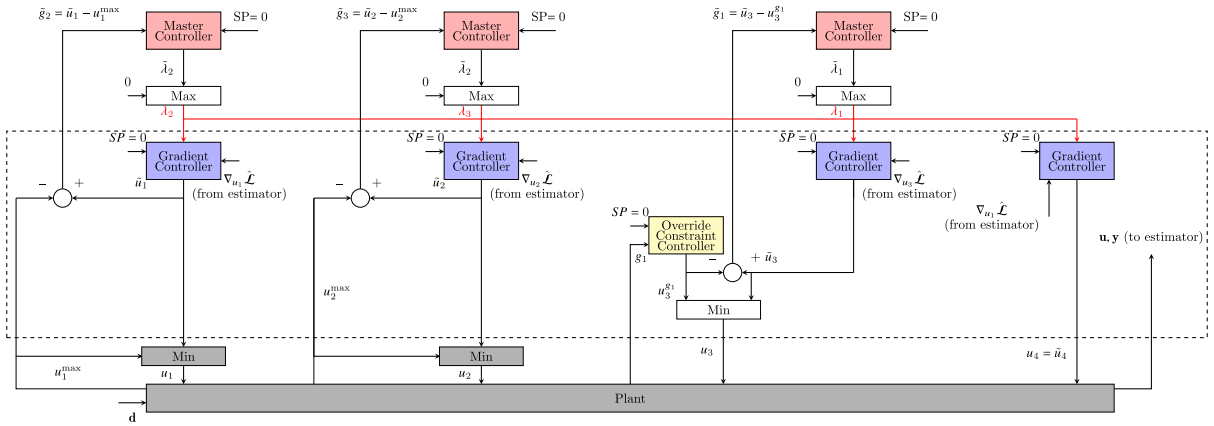


Fig. 10. Proposed primal–dual control scheme for the case study. For the case without override control, $\tilde{g}_1$ is replaced by the measured constraint g_1. The red lines indicate that all three Lagrange multipliers are distributed to each of the four gradient controllers.

valve openings are implicitly hard constraints, so we have $n_u = 4$ and $n_{g,o} = 5$. Fortunately, this is not a problem for our case study because two of the input constraints (on inputs 3 and 4) are never active for the assumed disturbances.

In these simulations, we consider the two disturbances, GOR_2, and q_g^{max} and they are varied over time. At $t = 5$ h, GOR_2 gradually decreases over 5 min from 0.1200 to 0.0360, rebounding to 0.1240 in 5 min at $t = 21$ h. Meanwhile, q_g^{max} gradually decreases in 5 min from 933.12 to 743.04 m^3/day at $t = 1$ h, gradually increases in 5 min to 915.84 m^3/day at $t = 5$ h, further gradually increases in 5 min to 1114.56 m^3/day at $t = 13$ h, and finally drops in 5 min to 864.00 m^3/day at $t = 21$ h.

4.3. Simplified problem description

Since $n_g = 5$ and $n_u = 4$, the primal–dual feedback-optimizing control ideally requires 5 master controllers and 4 gradient controllers. However, in the considered disturbance scenario (Fig. 9), constraints $g^{u_3^{\max}}$ and $g^{u_4^{\max}}$ are never active, implying that their associated Lagrange multipliers λ_i are always zero. Hence, the number of required master controllers can be reduced to three. In the remaining sections of the paper, we consider the following three constraints:

$$\mathbf{g} = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} g_1 \\ g^{u_1^{\max}} \\ g^{u_2^{\max}} \end{bmatrix} = \begin{bmatrix} q_g - q_g^{max} \\ u_1 - u_1^{\max} \\ u_2 - u_2^{\max} \end{bmatrix} \tag{17}$$

The associated Lagrange multipliers are denoted λ_1, λ_2, and λ_3, respectively.

The proposed control structure is shown in Fig. 10. All controllers are SIMC (Simple Internal Model Control)-tuned PI-controllers [27], with the closed-loop time constant τ_c as the tuning parameter. Note that we need a time scale separation between the fast override controller in the lower layer, the intermediate gradient controllers in the middle layer and the slower master constraint controllers in the upper layer. Typically, we need a time scale separation in the range 4 to 10 between each layer.

In the simulations, we will consider two cases: one with the override controller on the coupling constraint g_1 and one without.

4.4. Gradient controllers

For tuning the gradient controllers, we consider the square steady-state gain matrix G from the inputs (u) to the gradients ($\nabla_u \mathcal{L}$). Linearizing in region R-IV gives:

$$\mathbf{G} = \begin{bmatrix} 0.4092 & -0.0069 & -0.0139 & -0.0159 \\ -0.0068 & 0.5106 & -0.0150 & -0.0171 \\ -0.0131 & -0.0142 & 1.8861 & -0.0327 \\ -0.0149 & -0.0161 & -0.0325 & 2.3963 \end{bmatrix} \tag{18}$$

We notice from the nonzero off-diagonal elements that there are some interactions between the wells, as expected. However, the interactions are small, and single-loop (decentralized) gradient controllers will work well.

The four gradient controllers were tuned with a closed-loop time constant (τ_c) of 1.5 min. The resulting controllers tuning are available as supplementary material.

4.5. Master constraint controllers

With the four lower-layer gradient controllers tuned and functioning properly, we tune the three master constraint controllers. Note that the coupling constraint g_1 is a common constraint, whereas $g_2 = g^{u_1^{max}}$ and $g_3 = g^{u_2^{max}}$ represent local input constraints.

The three master constraint controllers were tuned with a closed-loop time constant (τ_c) of 7.5 min, corresponding to a time scale separation of 5 relative to the four gradient controllers. The resulting controllers tuning are available as supplementary material.

4.6. Override constraint controller

Inputs u_3 and u_4 never saturate and are therefore candidates for override control for the coupling constraint g_1. For the case study, we chose to use u_3. Inputs u_1 and u_2 may saturate, but the physical valve provides implicit override for these inputs.

The override controller for g_1 was tuned with a closed-loop time constant (τ_c) of 0.3 min, which again gives a time scale separation of 5 relative to the gradient controllers with $\tau_c = 1.5$ min. The resulting controllers tuning are available as supplementary material.

4.7. Gradient estimator

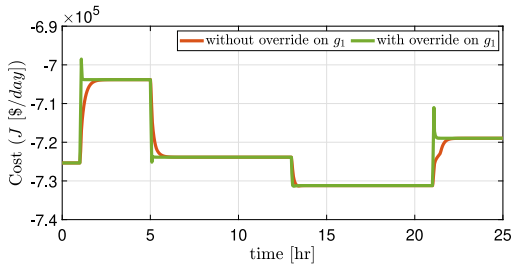
To evaluate the gradient of the Lagrange function with respect to the inputs, see (6), it is necessary to estimate the steady-state cost gradient and steady-state constraint gradient. In this case study, we follow [28] and linearize at each sample time the nonlinear model to obtain a linear model from u to the states x,

$$\dot{\mathbf{x}} = \mathbf{A}\Delta\mathbf{x} + \mathbf{B}\Delta\mathbf{u}$$

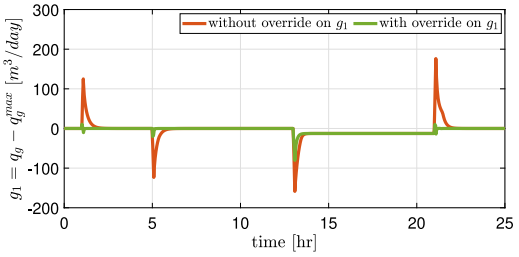
(in deviation variables). The static model for the cost $J(x, u, d)$ is linearized in a similar way to obtain

$$\Delta J = \mathbf{C}\Delta\mathbf{x} + \mathbf{D}\Delta\mathbf{u}$$

We do not need to include the dependency of $\dot{\mathbf{x}}$ and J on the disturbances d, because the disturbances are assumed constant into the future ($\Delta d = 0$), when we follow the "trick" in [28] of setting $\dot{\mathbf{x}} = \mathbf{0}$ to



(a) Cost J [$/day].



(b) Coupling constraint g_1.

Fig. 11. Simulation results with override (green) and without override (red). Without override we get constraint violation with $g_1 > 0$.

Table 1
Violation of constraint g_1 for case study.

	Without override	With override
Max. constraint violation [m^3/day]	176.2760	10.6436
Flared gas during 25 h [m^3]	2.9667	0.0444
Average flared gas [m^3/h]	0.1187	0.0018

eliminate $\Delta\mathbf{x}$ and obtain $\Delta J = \nabla_{\mathbf{u}}\mathbf{J}\Delta u$. The estimate of the steady-state cost gradient then becomes

$$\nabla_{\mathbf{u}}\hat{\mathbf{J}} = -\mathbf{C}\mathbf{A}^{-1}\mathbf{B} + \mathbf{D} \tag{19}$$

In this paper, we use an extended Kalman filter (EKF) to estimate the states and disturbances and based on these estimates, we relinearize the system at each sample time to obtain the matrices A, B, C and D.

The same approach is used to estimate the constraint gradient $\nabla_{u_i}\mathbf{g}_j\,(\mathbf{u}, \mathbf{d})$.

The gradient estimates can be achieved using any model-based or model-free method and the choice of method is not important for the results for this case study. For various gradient estimation techniques for RTO, refer to [29,30]. If unmeasurable state variables are present in model-based estimation, a novel piecewise fuzzy affine observer was proposed [31].

4.8. Simulation results

We present simulation results for the proposed primal–dual structure in Fig. 10, both with and without override control.

Fig. 11(a) shows that we reach the steady-state optimal operating points both with and without override. Fig. 11(b) shows that the override gives much better control of the coupling constraint (g_1). Table 1 shows this in more detail. We note that the amount of flared gas is reduced by about a factor 67 (from 2.9967 to 0.0444 m^3 over 25 h), and the maximum violation of the constraint is reduced by about a factor 17. Fig. 12 shows the four inputs (valve positions) for the case with override. It shows that the proposed approach is able to move correctly to all the active constraints.

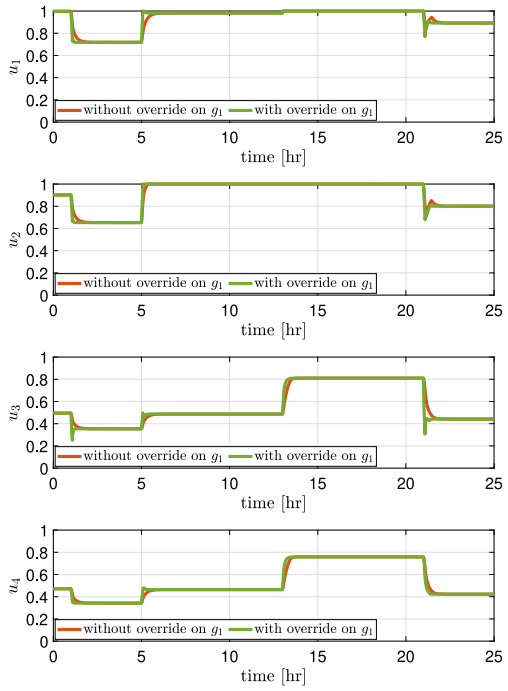


Fig. 12. Inputs u_i (gas lift valve positions) with override (green) and without override (red).

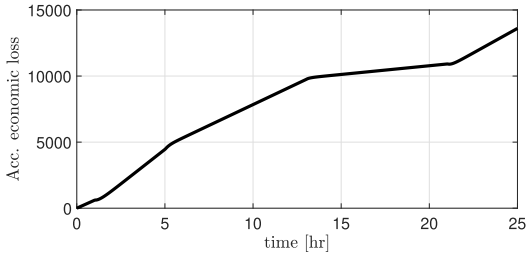


Fig. 13. Accumulated economic loss [$] caused by "back-off" for the case without override (the back-off is chosen to achieve the same constraint violation as with override).

4.9. Discussion of example

4.9.1. "Back-off"

For the present case study, we assume that the small amount of flared gas (0.0444 m^3 in 25 h) obtained with the override is acceptable, so we introduce a back-off on the case without override to get a similar constraint violation (flaring). We find that the required back-off on the export gas is $b_1 = 137.9$ m^3/day. In practice, this is implemented by changing the setpoint for g_1 from SP = 0 to SP = $-b_1 = -137.9$. The resulting economic loss by adding this back-off is about 14 000 $ (over 25 h) as seen from Fig. 13 where we plot as a function of time the accumulated economic loss.

Fig. 14 shows the value of the Lagrange multiplier λ_1 for the coupling constraint g_1 as a function of time. Note that the magnitude is correlated with the slope of economic loss in Fig. 13. Recall from (10) that the loss imposed by back-off is equal to $|\lambda_1 b_1|$ [$/s]. In our case, we have $|b_1| = 137.9$ m^3/day $= 0.0016$m^3/s and from the plot of the Lagrange multiplier in Fig. 14 (yellow line), we find that λ_1 is 125.2 [$/m^3] on average. The average loss is then approximately $|\lambda_1 b_1| = 0.20$ [$/s] and the accumulated loss over 25 h is then $0.20 \cdot 25 \cdot 3600 \approx 18000$ $ which agrees well with the final value of about 14 000 $ in Fig. 13. The 4000 $ discrepancy can be attributed mainly to the unconstrained operation between $t = 13$ h and $t = 21$ h.

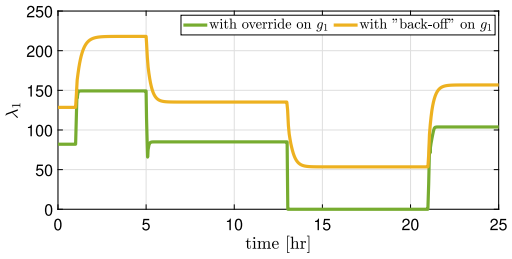


Fig. 14. The Lagrange multiplier λ_1 with "back-off" (and without override) (yellow line) is seen to be larger than with override (green line).

Furthermore, from Fig. 11(a) the ideal average profit is 722,000 $/day, which means that implementing override saves approximately 2% compared to a "back-off" strategy.

4.9.2. Necessity of using auxiliary input constraints

To emphasize the need to use the auxiliary input constraint (14) in the master controller, we compare in Fig. 15 the use of auxiliary constraints (green lines) with the incorrect use of the original input constraint (13) (purple lines). In the latter case, the Lagrange gradient does not go 0 at steady state as it should for optimality (Fig. 15(a)). We see from Fig. 15(b) that the reason is that without the auxiliary constraint, the inputs $\tilde{u}_1$ and $\tilde{u}_2$ (purple lines) do not violate their constraints and therefore there is no update of the corresponding Lagrange multiplier. Fig. 15(c) confirms that with the auxiliary constraint (green lines) the Lagrange multipliers λ_2 (associated with u_1) and λ_3 (associated with u_2) are positive (as they should) when the inputs become constrained.

5. Discussion

5.1. Comparison with reduced gradient approach

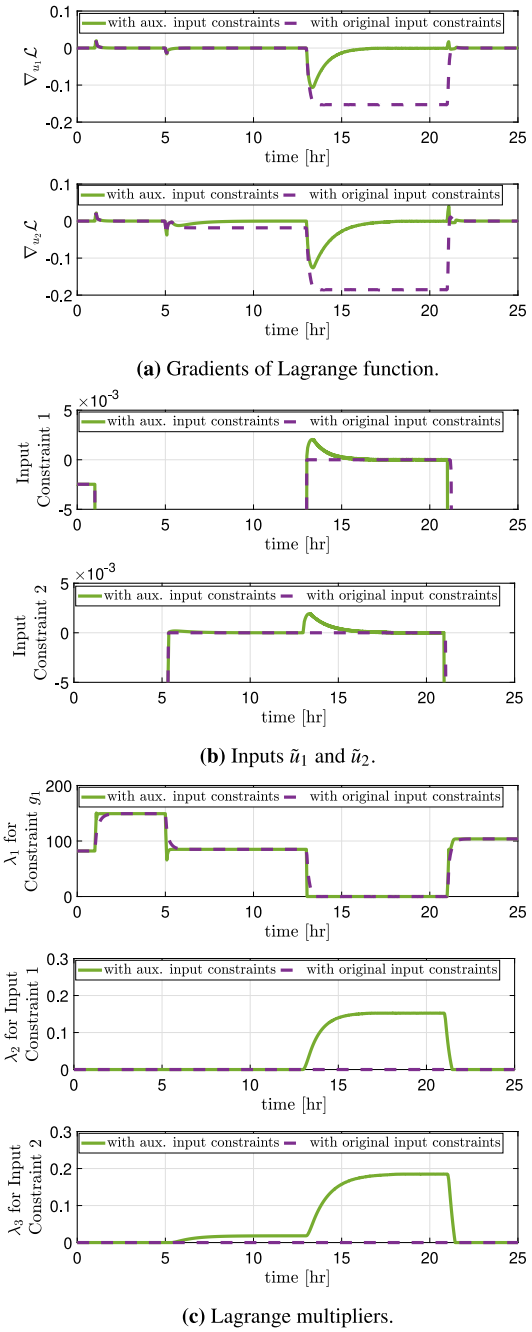
An alternative feedback-optimizing approach that allows for constraint control on a fast time scale, is the region-based approach using the reduced gradient [32]. However, this requires the user to identify a different control structure in each possible active constraints region. In our example, with five constraints ($n_{\mathbf{g}} = 5$), we may need to identify up to $2^{n_{\mathbf{g}}} = 32$ control structures, which is problematic in practice. On the other hand, for the primal–dual approach studied in this paper, the max-selectors in the master constraint controller provide an automatic switching between the active constraint regions. The disadvantage is that the master controllers operate on a slow time scale, which may lead to temporary (dynamic) constraint violations. The focus of the present paper has therefore been to show how this can be avoided with fast override control.

5.2. Comparison with numerical solver-based RTO

Compared to the standard numerical solver-based RTO, the proposed approach offers a lower level of complexity, and the constraints are measured and controlled in a *transparent* manner (without relying on a model for the constraints) in the upper layer. Additionally, when solving Eq. (3a) using feedback controllers, the computation time requirements are much less, as it uses only PID controllers.

5.3. Additional advantage of feedback implementation

In some cases, the decision variables u do not appear explicitly in the optimal stationary condition (3a) (Lagrange gradient = 0). For example, this may happen if the constraints are linear in the decision variables. In such cases a mathematical or numerical solver cannot be used. However, for primal–dual feedback-optimizing control, this limitation can be overcome by using feedback as a "trick" to satisfy (3a), see Fig. 2. In this feedback approach, the primal variables u are "indirectly" adjusted to satisfy (3a).



(a) Gradients of Lagrange function.

(b) Inputs $\tilde{u}_1$ and $\tilde{u}_2$.

(c) Lagrange multipliers.

Fig. 15. Illustration of necessity of using auxiliary input constraints (14) (green lines) rather then the original input constraints (13) (purple lines) in the Master controller.

5.4. Possible extension of method of multipliers

Another alternative for reducing the constraint violation is the method of multipliers (MoM) or augmented Lagrangian [33]. It involves incorporating penalty terms in the Lagrangian cost $\mathcal{L}$ to enforce constraint satisfaction during the (numerical) minimization of $\mathcal{L}$

However, in the implementation of MoM (discretized representation), the master controller (equipped with a `max` selector and weight (ρ) on the constraint) is limited by the time scale separation concept [14,22], rendering the penalty terms insignificant if the constraint is controlled in a slow time scale.

$$\lambda^{k+1} = \max\left[0, \lambda^k + \rho \mathbf{g}\right] \tag{20}$$

Table 2
Summary of the benefits of the proposed method.

No	List of benefit
1	Constraints are controlled transparently by the master controller.
2	Override control of constraints in the fast time scale when necessary.
3	Automatic active constraint switching.
4	Allows for local optimizations having different time scales.
5	Drives the system to the optimal steady-state operating point for varying disturbances
6	Uses feedback rather then models whenever possible.
7	No limit on the number of constraints that can be handled optimally at steady state.

Even with an appropriate (and enforced) value of ρ for the input constraint case, incorporating "back-off" remains necessary to minimize constraint violation further. This suggests that this strategy is inferior to the proposed override control with and "auxiliary" constraints. This is expected because the selector in the proposed strategy immediately eliminates the term from the "upper layer" in the calculation of the controlled variables ($\mathbf{c}(\lambda) = \mathbf{g}(\mathbf{u}, \mathbf{y})$).

5.5. Decomposing interactive systems

In the case of a non-interactive system, the decomposition and pairing of subsystems are relatively straightforward [9,13]. Here, coordination between the subsystems can achieve steady-state optimal performance effectively. However, in the case of a weakly-interactive system, the decomposition and pairing remain apparent, but the coordination between subsystems may only reach steady-state performance with a bounded loss [34]. One way to achieve optimal performance is to reformulate the problem (16); any interaction, such as input, from different subsystems, is modeled as additional coupling constraints in each subsystem.

5.6. Main benefit of decomposition using subsystems

In decomposed systems, the proposed approach allows for individual subproblems to have different tunings, closed-loop time constants, and sampling intervals. This is particularly beneficial in large-scale industrial processes where different subprocesses may have varying dynamics and measurement delays.

5.7. Comparison with primal decomposition

Yet another alternative to dual decomposition is primal decomposition where the constraint control is moved to the faster layer, which avoids the need for override. The unconstrained optimization is moved to a slower layer which contains a compensator subsystem to ensure primal feasibility (3b) [35]. However, in the fast constraint control layer, each subsystem requires an input (primal variable) to handle each coupling constraint, meaning that number of inputs (degrees of freedom) in each subsystem must at least be equal to the number of coupling constraints. This is often not satisfied, for example, it would not be satisfied for our case study if we added a constraint on the total gas lift in addition to existing constraint on total gas handling. On the other hand, dual decomposition does not have this serious limitation. This is a major advantage of the proposed primal–dual approach studied in this paper.

6. Conclusion

This paper uses the method of primal–dual feedback-optimizing control for real-time optimization, which gives optimal switching between active constraint regions. However, a significant drawback is that the constraint control is on a slow time scale, which is unacceptable for certain critical constraints, as it either results in unacceptable dynamic violations or results in economic loss because of constraint back-off. The main contribution of this paper is the incorporation of override constraint control on a fast time scale and auxiliary constraints on a slow time scale.

To demonstrate the effectiveness of the proposed method, we applied it to a case study involving the optimization of gas-lifted oil production. The results showcased the success of the proposed approach.

Table 2 provides a summary of the benefits of the proposed method. In conclusion, the proposed scheme in Fig. 5 offers a flexible and effective feedback alternative to real-time optimization, making it well-suited for industrial applications.

As future work, it is interesting to investigate further the strategy of decomposing highly interactive systems as outlined in Section 5.5.

CRediT authorship contribution statement

Risvan Dirza: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Data curation, Conceptualization. **Sigurd Skogestad:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization.

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.

Data availability

Data will be made available on request.

Appendix A. Supplementary data

The model equations and data for the case study is available as supplementary materials.

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jprocont.2024.103208.

References

[1] C. Cutler, R. Perry, Real time optimization with multivariable control is required to maximize profits, Comput. Chem. Eng. 7 (5) (1983) 663–667.
[2] M.L. Darby, M. Nikolaou, J. Jones, D. Nicholson, RTO: An overview and assessment of current practice, J. Process Control 21 (6) (2011) 874–884, http://dx.doi.org/10.1016/j.jprocont.2011.03.009.
[3] D. Krishnamoorthy, S. Skogestad, Real-time optimization as a feedback control problem - A review, Comput. Chem. Eng. (2022) 107723, http://dx.doi.org/10.1016/j.compchemeng.2022.107723.
[4] M. Campos, H. Teixeira, F. Liporace, M. Gomes, Challenges and problems with advanced control and optimization technologies, IFAC Proc. Vol. 42 (11) (2009) 1–8, http://dx.doi.org/10.3182/20090712-4-TR-2008.00003.
[5] M.M. Câmara, A.D. Quelhas, J.C. Pinto, Performance evaluation of real industrial RTO systems, Processes 4 (4) (2016) http://dx.doi.org/10.3390/pr4040044.
[6] B.J.T. Binder, D.K.M. Kufoalor, A. Pavlov, T.A. Johansen, Embedded model predictive control for an electric submersible pump on a programmable logic controller, in: 2014 IEEE Conference on Control Applications, CCA, 2014, pp. 579–585, http://dx.doi.org/10.1109/CCA.2014.6981402.

[7] M. Morari, Y. Arkun, G. Stephanopoulos, 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 J. 26 (2) (1980) 220–232, http://dx.doi.org/10.1002/aic.690260205.

[8] S. Boyd, L. Xiao, A. Mutapcic, J. Mattingley, Notes on decomposition methods, 2008, pp. 1–36.

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

[10] P. Wolfe, A duality theorem for non-linear programming, Quaterly Appl. Math. 19 (161) 239–244.

[11] A. Bernstein, E. Dall'Anese, A. Simonetto, Online primal-dual methods with measurement feedback for time-varying convex optimization, IEEE Trans. Signal Process. 67 (8) (2019) 1978–1991, http://dx.doi.org/10.1109/TSP.2019.2896112.

[12] A. Hauswirth, S. Bolognani, G. Hug, F. Dorfler, Timescale separation in autonomous optimization, IEEE Trans. Automat. Control 66 (2) (2021) 611–624, http://dx.doi.org/10.1109/TAC.2020.2989274.

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

[14] R. Dirza, J. Matias, S. Skogestad, D. Krishnamoorthy, Experimental validation of distributed feedback-based real-time optimization in a gas-lifted oil well rig, Control Eng. Pract. 126 (2022) 105253, http://dx.doi.org/10.1016/j.conengprac.2022.105253.

[15] R. Dirza, D. Krishnamoorthy, S. Skogestad, Primal-dual feedback-optimizing control with direct constraint control, in: Y. Yamashita, M. Kano (Eds.), 14th International Symposium on Process Systems Engineering, in: Computer Aided Chemical Engineering, vol. 49, Elsevier, 2022, pp. 1153–1158, http://dx.doi.org/10.1016/B978-0-323-85159-6.50192-5.

[16] R. Dirza, S. Skogestad, Online feedback-based optimization with multi-input direct constraint control, IFAC-PapersOnline 55 (7) (2022) 149–154, http://dx.doi.org/10.1016/j.ifacol.2022.07.436.

[17] R. Dirza, S. Skogestad, Systematic pairing selection for economic-oriented constraint control, in: L. Montastruc, S. Negny (Eds.), 32nd European Symposium on Computer Aided Process Engineering, in: Computer Aided Chemical Engineering, vol. 51, Elsevier, 2022, pp. 1249–1254.

[18] J. Nocedal, S.J. Wright, Numerical Optimization, second ed., Springer, New York, NY, USA, 2006.

[19] K. Arrow, L. Hurwicz, H. Uzawa, Studies in Linear and Non-Linear Programming, Stanford University Press, 1958.

[20] D. Krishnamoorthy, S. Skogestad, Systematic design of active constraint switching using selectors, Comput. Chem. Eng. 143 (2020) 107106, http://dx.doi.org/10.1016/j.compchemeng.2020.107106.

[21] S. Skogestad, Advanced control using decomposition and simple elements, Annu. Rev. Control 56 (2023) 107106, http://dx.doi.org/10.1016/j.arcontrol.2023.100903.

[22] S. Skogestad, I. Postlethwaite, Multivariable Feedback Control, John Wiley and Sons, New York, NY, USA, 2005, pp. 82–91.

[23] M. Fetkovich, The isochronal testing of oil wells, in: 1973 SPE Annual Technical Conference and Exhibition, in: SPE Annual Technical Conference and Exhibition, vol. All Days, 1973, pp. SPE–4529–MS.

[24] F. Elldakli, Gas lift system, Petrol. Petrochem. Eng. J. 1 (2017) http://dx.doi.org/10.23880/PPEJ-16000121.

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

[26] A. Reyes-Lúa, S. Skogestad, Systematic design of active constraint switching using classical advanced control structures, Ind. Eng. Chem. Res. 59 (6) (2020) 2229–2241, http://dx.doi.org/10.1021/acs.iecr.9b04511.

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

[28] D. Krishnamoorthy, E. Jahanshahi, S. Skogestad, Feedback real-time optimization strategy using a novel steady-state gradient estimate and transient measurements, Ind. Eng. Chem. Res. 58 (1) (2019) 207–216, http://dx.doi.org/10.1021/acs.iecr.8b03137.

[29] B. Srinivasan, G. François, D. Bonvin, Comparison of gradient estimation methods for real-time optimization, in: 21st European Symposium on Computer Aided Process Engineering, in: Computer Aided Chemical Engineering, vol. 29, Elsevier, 2011, pp. 607–611, http://dx.doi.org/10.1016/B978-0-444-53711-9.50122-X.

[30] L.F. Bernardino, S. Skogestad, Optimal measurement-based estimate of the cost gradient for real-time optimization, J. Process Control (2024) submitted for publication. (correction: submitted to Comp.Chem.Eng. instead)

[31] J. Qiu, W. Ji, H.-K. Lam, A new design of fuzzy affine model-based output feedback control for discrete-time nonlinear systems, IEEE Trans. Fuzzy Syst. 31 (5) (2023) 1434–1444, http://dx.doi.org/10.1109/TFUZZ.2022.3202360.

[32] J. Jäschke, S. Skogestad, Optimal controlled variables for polynomial systems, J. Process Control 22 (1) (2012) 167–179, http://dx.doi.org/10.1016/j.jprocont.2011.09.005.

[33] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends® Mach. Learn. 3 (1) (2010) 1–122, http://dx.doi.org/10.1561/2200000016.

[34] A. Rantzer, Dynamic dual decomposition for distributed control, in: 2009 American Control Conference, 2009, pp. 884–888, http://dx.doi.org/10.1109/ACC.2009.5160224.

[35] R. Dirza, M. Rizwan, S. Skogestad, D. Krishnamoorthy, Real-time optimal resource allocation using online primal decomposition, IFAC-PapersOnLine 55 (21) (2022) 31–36, http://dx.doi.org/10.1016/j.ifacol.2022.09.239, 19th IFAC Symposium on Control, Optimization and Automation in Mining, Mineral and Metal Processing MMM 2022.