



Review

Real-Time optimization as a feedback control problem – A review

Dinesh Krishnamoorthy^{a,b,*}, Sigurd Skogestad^b^a Harvard John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA^b Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

ARTICLE INFO

Article history:

Received 31 December 2020

Revised 19 October 2021

Accepted 5 February 2022

Available online 15 February 2022

Keywords:

Real-time optimization

Feedback control

Gradient estimation

Active constraint switching

ABSTRACT

Feedback optimizing control aims to achieve asymptotic optimal operation by directly manipulating the inputs using feedback controllers, without the need to solve numerical optimization problems online. The main question when designing feedback optimizing control is then *what to control* such that the economic objectives are translated into control objectives. Over the past two decades, active research has resulted in several different approaches to feedback optimizing control stemming from different application areas, leading to a rich source of literature. The different methods differ on the choice of the controlled variables, the need for detailed process models, type of measurements used, convergence speed, accuracy, ease of implementation etc. This paper aims to provide a survey of these different approaches under the unified umbrella of *feedback optimizing control*, and provides an overview and comparison of the different feedback-based real-time optimization approaches for continuous processes.

© 2022 The Authors. Published by Elsevier Ltd.

This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>)

1. Introduction

In the face of growing market competition and increased focus on sustainability, energy efficiency and safety, online optimization of process operations are becoming increasingly important. Real-time optimization (RTO) explicitly deals with the optimal economic operation of the process. A widely accepted definition of real-time optimization is that it is a work flow where the decision variables are iteratively adjusted using the system model and/or real time process measurements, in order to minimize the operational cost and satisfy constraints (Marlin and Hrymak, 1997; Chachuat et al., 2009; Krishnamoorthy, 2019).

1.1. Conventional real-time optimization (RTO)

Traditional real-time optimization (RTO) framework computes the optimal steady-state setpoints by solving a steady-state numerical optimization problem that comprises of the following components

1. Rigorous nonlinear steady-state process models,
2. Process, equipment, and environmental constraints,

3. Economic objective function that constitutes the cost of raw material, value of the products, operational costs, environmental taxes etc.

Traditional RTO implementations are thus based on steady-state nonlinear models. The model parameters are updated using data from time periods corresponding to steady-state process operation, in order to match the plant measurements and the model predictions. The updated model is then used to re-optimize for the new setpoints (Seborg et al., 2010, Chapter 19), (Marlin and Hrymak, 1997). The traditional RTO framework is thus a two-step approach:

- Step 1: Data reconciliation - Update nonlinear model using steady-state measurements.
- Step 2: Numerical optimization - Compute new optimal setpoints using the updated model.

Most of the commercially available RTO software packages are based on the repeated identification and optimization scheme using steady-state models (Cámara et al., 2016). The justification for this approach is that for most continuous processes, the economic operation of the plant often occurs at some steady-state. The exception are processes with frequent grade changes, cyclic operations, and batch processes. As such, the objective in most continuous processes is simply to find the economically optimal steady-state operating point as a function of the given operating conditions. To this end, steady-state process optimization for continuous processes will be the main focus in this paper.

* Corresponding author at: Harvard John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA.

E-mail addresses: dkrishnamoorthy@seas.harvard.edu (D. Krishnamoorthy), sigurd.skogestad@ntnu.no (S. Skogestad).

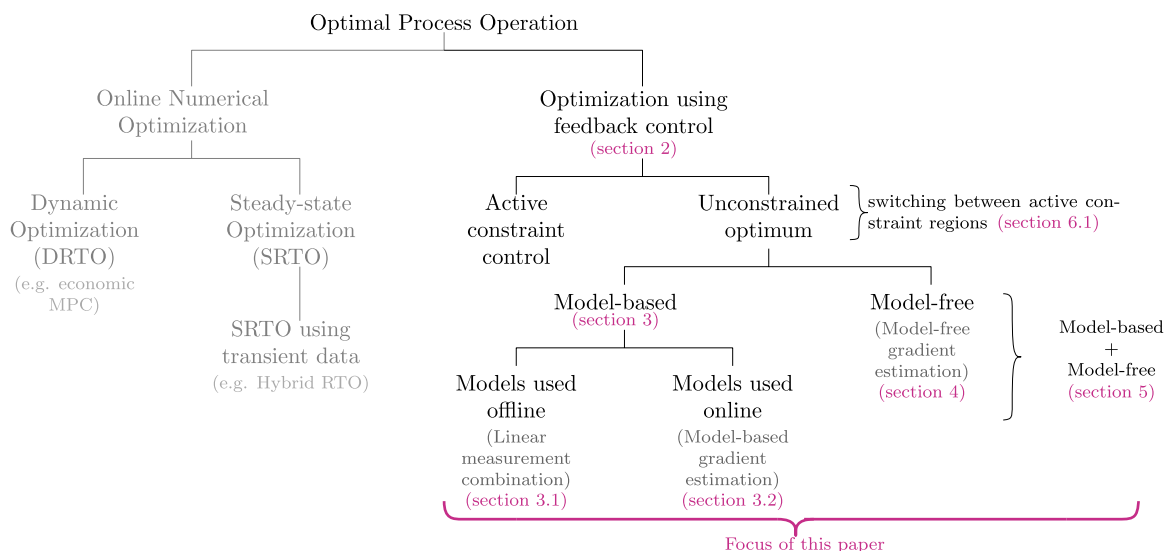


Fig. 1. Overview of online process optimization approaches, and the structure of this paper.

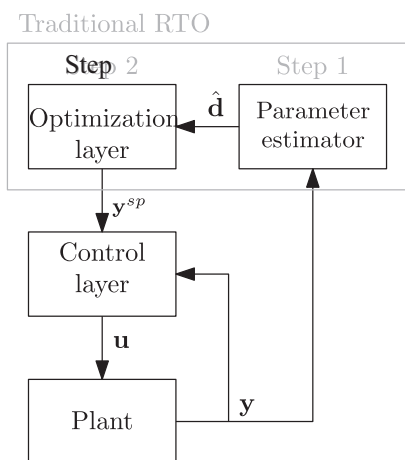


Fig. 2. Traditional hierarchical decomposition into optimization and control layers.

The decision variables for the RTO layer are typically the setpoints for the controlled variables, which are then given to a setpoint tracking control layer below. The control layer adjusts the manipulated variables in order to keep the measurements at the optimal setpoints computed by the RTO layer above. Fig. 2 shows the typical hierarchical decomposition into optimization and control layers.

Despite the economic benefits and promises, traditional real-time optimization is not widely used in practice as one would expect. Consequently, the full potential of RTO is not exploited in process industries (Darby et al., 2011; Krishnamoorthy et al., 2018a; Krishnamoorthy, 2019). The main technical challenges that limit industrial use of steady-state RTO for process operations include:

Challenge 1 Cost of developing the model (offline model development) - Developing good first principle-based models is often challenging and expensive, especially for new application areas with limited domain knowledge. In addition, lack of knowledge or model simplification leads to mismatch between the physical models used in the optimization problem and the real system. With increasing complexity of many industrial processes, simplified first-principle models are insufficient to accurately capture the system behavior.

Challenge 2 Model uncertainty, including wrong values of disturbances and parameters (online update of the model) - Since traditional RTO uses steady-state models, the model parameters must be updated using measurements that corresponds to steady-state conditions. To this end, steady-state detection algorithms are employed to detect if the process is operating at steady-state conditions, see for example Kelly and Hedengren (2013), Câmara et al. (2016) and the references therein. The time between the steady-state conditions is known as steady-state wait time. In a recent review paper on current practices of RTO, Darby et al. (2011) conclude that a fundamental limiting factor of RTO implementation is the steady-state wait-time associated with the online model update. Câmara et al. (2016) also recently pointed out some of the issues with steady-state detection routines used in real industrial RTO systems.

Challenge 3 Numerical robustness, including computational issues of solving optimization problems - Solving numerical optimization problems to compute the optimal setpoints, leads to high computational effort. Although the computational cost is considerably less for solving steady-state optimization problems than dynamic optimization problems, the optimization problem may still fail to converge for large-scale processes due to numerical robustness and convergence issues (Marlin and Hrymak, 1997).

Challenge 4 Conflicts with the planning layer - In process applications, the planning and scheduling layer (which typically sits on top of the RTO layer in the decision-making hierarchy) deals with decisions such as feed purchases, production capacity, storage, and other such operational planning over a time horizon of weeks and months, based on average plant behavior (Darby et al., 2011). RTO, on the other hand, deals with real-time decisions based on the current plant conditions. Since the economics are considered at both the RTO and planning levels, this can lead to inconsistencies, as what is optimum for a process unit or section may not be optimum when all production aspects are considered over a large horizon.

An alternative to the traditional RTO paradigm is the so-called “feedback optimizing control”, where the optimization objectives are translated into control objectives, thereby moving the optimization functionality into the control layer (Morari et al., 1980). In a recent review paper, Darby et al. (2011) also considers the question of moving the RTO functionality to the control layer (more specifically MPC) as an alternative RTO formulation, thus resulting in a one-layer solution that “performs a type of hill-climbing

optimization". Achieving optimal operation via feedback control circumvents challenge 3, since numerical optimization problems are not solved online. Feedback optimizing control also avoids the steady-state wait-time (challenge 2), since optimal operation is achieved using real time process measurements as feedback. Finally, as will be shown later, many feedback optimizing control approaches may not require detailed process models, hence addressing challenge 1.

Challenges related to human aspects In addition to the technological challenges mentioned above, human aspects such as technical competence and corporate culture are some of the main limiting factors of industrial practice. Conventional real-time optimization requires regular maintenance and monitoring. This is often provided by a team of skilled engineers and a dedicated group with expertise in modeling, control, and optimization, which the company may be lacking. Consequently, the expected benefits of such advanced tools may not be fully realized and this often leads to the application being turned off (Shook, 2006; Forbes et al., 2015; Darby et al., 2011). As Marlin and Hrymak (1997) pointed out, "the greatest demand from optimization users is a system that can be easily understood". For this reason, many traditional process industries still prefer to use simple feedback control tools available in standard digital control system (DCS) to optimize process operations. In such cases, feedback optimizing control provides a systematic procedure that enables asymptotic optimal operation using feedback control.

As shown in Fig. 1, online process optimization approaches can be broadly categorized into two classes, based on whether one needs to solve numerical optimization problems online, or not. Traditional steady-state RTO and dynamic RTO (including economic MPC) belongs to the former category, where numerical optimization problems are solved online, in order to compute the optimal operating conditions. Feedback optimizing control, on the other hand, belongs to the latter category where asymptotic optimal operation is achieved via feedback control. Here, the main idea is to achieve optimal operation by directly manipulating the inputs \mathbf{u} based on feedback measurements \mathbf{y} such that the process is operated optimally at steady-state. In other words, the objective is to eliminate the RTO layer by translating the economic objectives into control objectives. This is commonly known as "feedback optimizing control", "direct input adaptation", or "implicit RTO" (Morari et al., 1980; Chachuat et al., 2009; Srinivasan and Bonvin, 2019).

Feedback optimizing control for the unconstrained degrees of freedom is an active area of research and has been receiving significant interest over the past two decades. Different feedback-based RTO methods have been proposed not only in the process control literature, but also in electrical and mechanical engineering literature (Ariyur and Krstic, 2003; Tan et al., 2010; Hauswirth et al., 2017). The main objective of this paper is to provide an overview and understanding of the different approaches to achieving asymptotic optimal operation using feedback control, without the need to solve numerical optimization problems online. Different model-based and model-free methods, and a combination of these will be reviewed in this paper.

Much of the research in this field has been devoted to using steady-state cost gradients \mathbf{J}_u as the self-optimizing variable, with a few exceptions such as Alstad et al. (2009), Larsson and Skogestad (2000), Halvorsen et al. (2003). Naturally, this survey paper could also be seen as a review of different gradient estimation methods for real time optimization, where we also discuss and compare the performance of different approaches in terms of accuracy, convergence time, ease of implementation, scalability etc. It is important to note that the objective of this paper is not to convince that simple control structures are superior to model-based optimization routines, but rather to demonstrate that there exists a wide array of methods that can be used to asymptotically drive

a process to its optimum using feedback control, as an alternative to model-based steady-state real-time optimization.

The remainder of the paper is organized as follows: The underlying concept of feedback optimizing control relating to the first-order optimality conditions is shown in Section 2. Section 3 provides a detailed survey of the model-based approaches. More precisely, Section 3.1 reviews the methods where rigorous nonlinear process models are used offline to select a linear combination of measurements as self-optimizing variables, whereas Section 3.2 reviews model-based gradient estimation methods. Section 4 reviews the model-free gradient estimation methods, where the steady-state cost gradient is estimated directly from the cost measurement. The combination of model-based and model-free approaches are reviewed in Section 5. Section 6 deals with optimal operation with changing operating conditions. The performance of the different methods are compared using a benchmark Williams–Otto reactor example in Section 7. Detailed discussions on the key features of the different methods are provided in Section 8 before concluding the paper in Section 9.

Notational remark In the rest of the paper, we use the following notations.

- $\tilde{\mathbf{u}}$ - Physical manipulated variables (primal variables)
- $\mathbf{u} \subseteq \tilde{\mathbf{u}}$ - unconstrained degrees of freedom
- \mathbf{d} - parameters and process disturbances
- \mathbf{y}_m - vector of process measurements
- J - Cost measurement
- J_{model} - Cost predicted by model
- $\hat{\mathbf{J}}_{u,model}$ - Model-based gradient estimate
- $\hat{\mathbf{J}}_u$ - Model-free gradient estimate
- $\mathbf{C}(\tilde{\mathbf{u}}, \mathbf{d})$ - set of constraints
- $\mathbf{C}_A(\tilde{\mathbf{u}}, \mathbf{d})$ - set of active constraints
- λ - Lagrange multipliers of constraints \mathbf{C} (dual variables)
- λ_A - Lagrange multipliers of active constraint set \mathbf{C}_A

2. Feedback optimizing control

Morari et al. (1980) first introduced the concept of *feedback optimizing control*, where the main aim was to translate the economic objectives into process control objectives. This can be done by choosing the "right controlled variables", which when controlled to a constant setpoint, leads to economically optimal operation at steady-state. The most important question is then,

What variables to control, such that the economic objectives are translated into control objectives?

To answer this, consider the steady-state optimization problem,

$$\begin{aligned} \min_{\tilde{\mathbf{u}}} \quad & J(\tilde{\mathbf{u}}, \mathbf{d}) \\ \text{s.t.} \quad & \\ & \mathbf{C}(\tilde{\mathbf{u}}, \mathbf{d}) \leq 0 \end{aligned} \quad (1)$$

where $\tilde{\mathbf{u}} \in \mathbb{R}^{n_{\tilde{u}}}$ denotes the vector of manipulated variables (MV) and $\mathbf{d} \in \mathbb{R}^{n_d}$ denotes the vector of disturbances, $J: \mathbb{R}^{n_{\tilde{u}}} \times \mathbb{R}^{n_d} \rightarrow \mathbb{R}$ is the scalar cost function and $\mathbf{C}: \mathbb{R}^{n_{\tilde{u}}} \times \mathbb{R}^{n_d} \rightarrow \mathbb{R}^{n_c}$ denotes the vector of constraints. The Lagrangian of the optimization problem is given by,

$$\mathcal{L}(\lambda, \tilde{\mathbf{u}}, \mathbf{d}) := J(\tilde{\mathbf{u}}, \mathbf{d}) + \lambda^T \mathbf{C}(\tilde{\mathbf{u}}, \mathbf{d}) \quad (2)$$

where $\lambda \in \mathbb{R}^{n_c}$ are the Lagrange multipliers of the inequality constraints, and the first-order optimality conditions for this optimization problem is given by:

$$\nabla_{\tilde{\mathbf{u}}} J(\tilde{\mathbf{u}}^*, \mathbf{d}) + \lambda^{*T} \nabla_{\tilde{\mathbf{u}}} \mathbf{C}(\tilde{\mathbf{u}}^*, \mathbf{d}) = 0 \quad (3a)$$

$$\mathbf{C}(\tilde{\mathbf{u}}^*, \mathbf{d}) \leq 0 \quad (3b)$$

$$\lambda^* \geq 0 \quad (3c)$$

$$\lambda_i^* \mathbf{C}_i(\tilde{\mathbf{u}}^*, \mathbf{d}) = 0, \forall i = 1, \dots, n_c \quad (3d)$$

Assumption 1. Linear independence constraint qualification, strict second order sufficient condition and strict complementarity holds for the optimization problem (1).

The above assumption implies that the optimization problem (1) has a unique primal and dual solution $(\tilde{\mathbf{u}}^*, \lambda^*)$, and the objective is to asymptotically drive the plant to this unique local optimum by using feedback control. Note that feedback optimizing control only deals with asymptotic optimal operation and not with optimizing the transient behavior. Transient behaviour is typically determined by the controller tuning parameters.

There are two main paradigms that can be used to translate the economic objectives into control objectives:

1. **Region-based control:** In this paradigm, the idea is to first identify the different active constraint regions *a priori* and use a subset of physical manipulated variables (primal variables) to control the set of active constraints (constraints that are optimally at their limiting value) in each region. The remaining unconstrained degrees of freedom in each region are then used to satisfy the first-order optimality condition of the reduced unconstrained optimization problem. This is discussed in detail in Section 2.1.
2. **Primal-dual feedback optimizing control:** In this paradigm, both the primal and the dual variables are considered as degrees of freedom, which are adjusted using feedback control to satisfy the first-order optimality condition of the original constrained optimization problem. This approach does not require the different active constraint regions to be identified *a priori*, and is therefore more general than the region-based approach. This is discussed in detail in Section 2.2.

2.1. Region-based control

2.1.1. Active constraint control

If optimal process operation corresponds to the case when some of the constraints are active (i.e. at its limiting value), then the easiest choice for the controlled variables are the active constraints itself.

Let $n_a \leq n_c$ denote the number of active constraints $\mathbf{C}_A(\tilde{\mathbf{u}}, \mathbf{d})$ at the optimum for a given disturbance realization \mathbf{d} . Since the constraints are typically measured in the process, for each active constraint, we usually associate the constraint itself as the controlled variable, i.e. $\mathbf{CV} = \mathbf{C}_A$, which are controlled to its limiting value using feedback control.

Active constraint control is a well known idea and has been used in many examples, see Maarleveld and Rijnsdorp (1970), Arkun and Stephanopoulos (1980), Morari et al. (1980), Fisher et al. (1988), Jacobsen and Skogestad (2012), Reyes-Lúa et al. (2018), Krishnamoorthy and Skogestad (2019), Krishnamoorthy et al. (2019a) to name a few.

The first step to designing active constraint control is to identify the potential combination of active constraints that may be encountered. The rigorous approach to identifying the different active constraint regions as a function of disturbances is by offline optimization, which requires good process models (Jacobsen and Skogestad, 2012). However, if reliable models are not available (cf. Challenge 1), then one can identify the relevant active constraint combinations based on good process understanding and “engineering intuition” (Skogestad, 2004). To be systematic, one can start by writing down all the 2^{n_c} active constraint combinations, and eliminate the constraint combinations that are not feasible or not likely

(Reyes-Lúa and Skogestad, 2019b). For example, one can eliminate certain constraint combinations using the following guidelines:

- The maximum number of active constraints $n_a \leq n_{\tilde{u}}$ for the problem to be feasible.
- Certain constraint combinations are not possible, e.g. maximum and minimum constraint on the same variable cannot be active at the same time.
- Certain constraints are always active, e.g. purity constraint on most valuable product to avoid product giveaway (Govatsmark and Skogestad, 2005), (Skogestad and Postlethwaite, 2007, Chapter 10).

Note that it is sufficient to identify the expected active constraint combinations, and one does not need to know exactly where the switching occurs. It is reasonable to assume that all the constraints are measured for monitoring purposes, and one can therefore detect whether the constraints are approaching its limiting value or not.

At the same time, it is also important to note that eliminating certain active constraint combinations without solving the optimization problem offline requires good process understanding and insight, and therefore it is not guaranteed that all the relevant active constraint regions will be considered. As such, this is more suited for small-scale unit operations with few constraints, where it is possible to identify the expected active constraint combinations easily.

Due to imperfect control and noise it may be desirable to add a safety margin to the active constraint controllers. To avoid dynamic constraint violations, we may choose to implement a back-off, where the setpoint for the active constraints \mathbf{C}_A are offset by a margin ϵ . This gives rise to a loss, which increases linearly, and is quantified by the corresponding Lagrange multipliers λ_A as,

$$\text{Loss} = \lambda_A^T \epsilon \quad (4)$$

See Appendix A for the proof. Quantifying the loss due to back-off provides two important insights:

1. It tells us which constraints need to be tightly controlled.
2. It allows us to simplify the control structure design. That is, if the Lagrange multiplier for a given constraint is sufficiently small such that the loss is negligible, then we can allow a large back-off (Govatsmark and Skogestad, 2005).

To this end, by tightly controlling the active constraints at their limiting value, one then only needs to find *what to control* using the remaining $n_{\tilde{u}} - n_a$ unconstrained degrees of freedom. In other words, the total $n_{\tilde{u}}$ available degrees of freedom are partitioned into two subsets $\tilde{\mathbf{u}} = [\mathbf{u}' \quad \mathbf{u}]^T$, where $\mathbf{u}' \in \mathbb{R}^{n_a}$ is used to control the active constraints and $\mathbf{u} \in \mathbb{R}^{n_{\tilde{u}} - n_a}$ is used to control the self-optimizing variables.

2.1.2. Unconstrained optimum

For the remaining $n_u := (n_{\tilde{u}} - n_a)$ unconstrained degrees of freedom denoted by, $\mathbf{u} \subset \tilde{\mathbf{u}}$, we need to choose *what to control*. This is not obvious, and requires more knowledge about the process, either in the form of process models or direct cost measurement, in order to translate the economic objectives into feedback control objectives. This additional information can be used to identify suitable “self-optimizing” controlled variables $\mathbf{c} \in \mathbb{R}^{n_u}$, and their corresponding setpoints \mathbf{c}^{sp} .

Skogestad (2000) defined self-optimizing control as when we can achieve (near) optimal operation (acceptable loss) with constant setpoint for the self-optimizing controlled variables. This can either be a single measurement (Section 3.1.1), a linear combination of measurements (Section 3.1.2), or some other optimization specific feature such as the steady-state cost gradient, that translates the economic objectives into control objectives. The ideal self-optimizing variable is the steady-state cost gradient (Sections 3.2

and 4), which when kept at a constant setpoint of zero, satisfies the necessary conditions of optimality (Halvorsen et al., 2003; François et al., 2012; 2005). For instance, consider the gradient of the reduced system (after controlling the active constraints),

$$\min_{\mathbf{u}} J(\mathbf{u}, \mathbf{d}) \quad (5)$$

where $\mathbf{u} \in \mathbb{R}^{n_u}$ is the vector of unconstrained degrees of freedom. The KKT optimality conditions for (5) state that the necessary condition of optimality is when the reduced gradient of the cost function is zero.

$$\mathbf{J}_u(\mathbf{d}) := \nabla_{\mathbf{u}} J(\mathbf{u}^*, \mathbf{d}) = 0 \quad (6)$$

Equivalently, one can also consider the full system (1), in which case the n_u self-optimizing variables are given by a linear combination of the cost gradient w.r.t. all the manipulated variables $\tilde{\mathbf{u}}$,

$$\mathbf{J}_u(\mathbf{d}) := \nabla_{\mathbf{u}} J(\mathbf{u}, \mathbf{d}) = \mathbf{N}^T \nabla_{\tilde{\mathbf{u}}} J(\tilde{\mathbf{u}}, \mathbf{d}) = 0 \quad (7)$$

where \mathbf{N} is the nullspace of the active constraint gradients (i.e. $\mathbf{N}^T \nabla_{\tilde{\mathbf{u}}} \mathbf{C}(\tilde{\mathbf{u}}, \mathbf{d}) = 0$). This equivalent approach results in a more generalized framework that can be used for different active constraint combinations. Note that this is the same as the gradient of the reduced system (see for example Krishnamoorthy and Skogestad, 2020a; Krishnamoorthy and Skogestad, 2019; Jäschke and Skogestad, 2012; Marchetti et al., 2016; Marchetti et al., 2020 to name a few).

Theorem 1 (Generalized framework for region-based feedback optimizing control Krishnamoorthy (2019), Krishnamoorthy and Skogestad (2020a)). *Given Assumption 1, the steady-state optimization problem (1) for a given disturbance realization \mathbf{d} can be transformed into feedback control problem by controlling (in this order):*

1. Active constraints $\mathbf{C}_A(\tilde{\mathbf{u}}, \mathbf{d}) = 0$.
2. Self-optimizing variables $\mathbf{c} = \mathbf{c}^{SP}$, where

$$\mathbf{c} = \mathbf{N}^T \nabla_{\tilde{\mathbf{u}}} J(\tilde{\mathbf{u}}, \mathbf{d})$$

and $\mathbf{c}^{SP} = 0$

Proof. From Assumption 1, the Lagrangian of (1) is given by

$$J(\tilde{\mathbf{u}}, \mathbf{d}) + [\lambda_A \quad \lambda_I]^T \begin{bmatrix} \mathbf{C}_A(\tilde{\mathbf{u}}, \mathbf{d}) \\ \mathbf{C}_I(\tilde{\mathbf{u}}, \mathbf{d}) \end{bmatrix} \quad (8)$$

$$= J(\tilde{\mathbf{u}}, \mathbf{d}) + \lambda_A^T \mathbf{C}_A(\tilde{\mathbf{u}}, \mathbf{d}) \quad (9)$$

where we have partitioned the system into set of active constraints (denoted by $(\cdot)_A$) and inactive constraints (denoted by $(\cdot)_I$). The stationarity condition gives,

$$\nabla_{\tilde{\mathbf{u}}} J(\tilde{\mathbf{u}}, \mathbf{d}) + \lambda_A^T \nabla_{\tilde{\mathbf{u}}} \mathbf{C}_A(\tilde{\mathbf{u}}, \mathbf{d}) = 0 \quad (10)$$

$$\Rightarrow \nabla_{\tilde{\mathbf{u}}} J(\tilde{\mathbf{u}}, \mathbf{d}) = -\lambda_A^T \nabla_{\tilde{\mathbf{u}}} \mathbf{C}_A(\tilde{\mathbf{u}}, \mathbf{d}) \quad (11)$$

Pre-multiplying (11) by \mathbf{N}^T gives

$$\mathbf{N}^T \nabla_{\tilde{\mathbf{u}}} J(\tilde{\mathbf{u}}, \mathbf{d}) = -\mathbf{N}^T \nabla_{\tilde{\mathbf{u}}} \mathbf{C}_A(\tilde{\mathbf{u}}, \mathbf{d}) \lambda_A \quad (12)$$

and since \mathbf{N} is chosen such that $\mathbf{N}^T \nabla_{\tilde{\mathbf{u}}} \mathbf{C}_A(\tilde{\mathbf{u}}, \mathbf{d})^T = 0$, we have $\mathbf{N}^T \nabla_{\tilde{\mathbf{u}}} J(\tilde{\mathbf{u}}, \mathbf{d}) = 0$.

Therefore, keeping the active constraints $\mathbf{C}_A(\tilde{\mathbf{u}}, \mathbf{d}) = 0$ together with $\mathbf{N}^T \nabla_{\tilde{\mathbf{u}}} J(\tilde{\mathbf{u}}, \mathbf{d}) = 0$ fully specifies the first order optimality conditions. \square

This approach requires online estimation of the steady-state cost gradient $\nabla_{\tilde{\mathbf{u}}} J(\tilde{\mathbf{u}}, \mathbf{d})$, as well as the constraint gradients $\nabla_{\tilde{\mathbf{u}}} \mathbf{C}(\tilde{\mathbf{u}}, \mathbf{d})$. Any of the model-based gradient estimation method reviewed in Section 3.2 or any model-free gradient estimation method reviewed in Section 4 can be used for this purpose.

To summarize, we have thus far considered the case, where the controlled variables were chosen by partitioning the set of constraints into active and inactive sets. This resulted in different active constraint regions as a function of disturbances. In each region, n_a degrees of freedom were used to control the active constraints, and $n_{\tilde{u}} - n_a$ remaining unconstrained degrees of freedom were used to control the self-optimizing variables (such as the steady-state cost gradient of the reduced system). When the set of active constraints change, this requires us to switch the set of controllers accordingly. This can be achieved by using multivariable control such as MPC that tracks the constraints and self-optimizing variables in neighboring regions, or by using classical control elements such as selectors, split-range etc. This is discussed in detail in Section 6. In other words, by identifying and controlling the active constraints tightly, we only need to transform a simpler unconstrained optimization of the reduced system, and switch between the different controllers in each active constraint region accordingly. The general control structure under this paradigm is schematically represented in Fig. 3a.

2.2. Primal-Dual feedback optimizing control

Alternatively, one can also transform the optimization problem into a feedback control problem without partitioning the set of constraints into active and inactive sets *a priori*. This approach is especially useful if there are many active constraint regions to switch between and/or if it is not possible to identify the different active constraint regions *a priori* (e.g. due to lack of good models). The main idea here is to consider the Lagrangian of the optimization problem (2) as an unconstrained optimization problem with $n_{\tilde{u}} + n_c$ degrees of freedom. More precisely, the degrees of freedom for the feedback optimizing control are now given by both the physical manipulated variables $\tilde{\mathbf{u}}$ (which are the primal variables in (1)) as well as the Lagrange multipliers λ (which are the dual variables in (1), also known as shadow prices).

Consider the relaxed optimization problem (2), for which the stationarity condition w.r.t. to the $n_{\tilde{u}} + n_c$ degrees of freedom (denoted jointly as $[\tilde{\mathbf{u}} \quad \lambda]^T$) is given by,

$$\begin{bmatrix} \nabla_{\tilde{\mathbf{u}}} \mathcal{L}(\lambda, \tilde{\mathbf{u}}, \mathbf{d}) \\ \nabla_{\lambda} \mathcal{L}(\lambda, \tilde{\mathbf{u}}, \mathbf{d}) \end{bmatrix} = \begin{bmatrix} \nabla_{\tilde{\mathbf{u}}} J(\tilde{\mathbf{u}}^*, \mathbf{d}) + \lambda^{*T} \nabla_{\tilde{\mathbf{u}}} \mathbf{C}(\tilde{\mathbf{u}}^*, \mathbf{d}) \\ \mathbf{C}(\tilde{\mathbf{u}}^*, \mathbf{d}) \end{bmatrix} = 0 \quad (13)$$

Therefore, the $n_{\tilde{u}}$ degrees of freedom are used to control the stationary condition such that $\nabla_{\tilde{\mathbf{u}}} \mathcal{L}(\lambda, \tilde{\mathbf{u}}, \mathbf{d}) = \nabla_{\tilde{\mathbf{u}}} J(\tilde{\mathbf{u}}, \mathbf{d}) + \lambda^T \nabla_{\tilde{\mathbf{u}}} \mathbf{C}(\tilde{\mathbf{u}}, \mathbf{d}) = 0$ for any given λ . Additional n_c feedback controllers are used to update λ such that the constraints $\mathbf{C}(\tilde{\mathbf{u}}, \mathbf{d})$ are controlled to a constant setpoint of zero (i.e. active constraint control). We also have the additional dual feasibility constraint $\lambda \geq 0$ (cf. (3c)). However, at any given time, either only $\lambda \geq 0$ or $\mathbf{C}(\tilde{\mathbf{u}}, \mathbf{d}) \leq 0$ will be active, thanks to the complementary slackness condition (3d). Therefore, dual feasibility and complementary slackness is taken care by using a max-selector as shown in Fig. 3b. In other words, the ‘‘active constraint control’’ in this paradigm is achieved using the Lagrange multipliers λ (i.e. the dual degrees of freedom), instead of a subset of the primal degrees of freedom $\mathbf{u}' \subseteq \tilde{\mathbf{u}}$.

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

$$\begin{bmatrix} \dot{\tilde{\mathbf{u}}} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} \nabla_{\tilde{\mathbf{u}}} J(\tilde{\mathbf{u}}, \mathbf{d}) + \lambda^T \nabla_{\tilde{\mathbf{u}}} \mathbf{C}(\tilde{\mathbf{u}}, \mathbf{d}) \\ \max(0, \mathbf{C}(\tilde{\mathbf{u}}, \mathbf{d})) \end{bmatrix} \quad (14)$$

which is also commonly known as saddle-point flow (Venets, 1985; Feijer and Paganini, 2010).

To this end, we have $(n_{\tilde{u}} + n_c)$ feedback controllers:

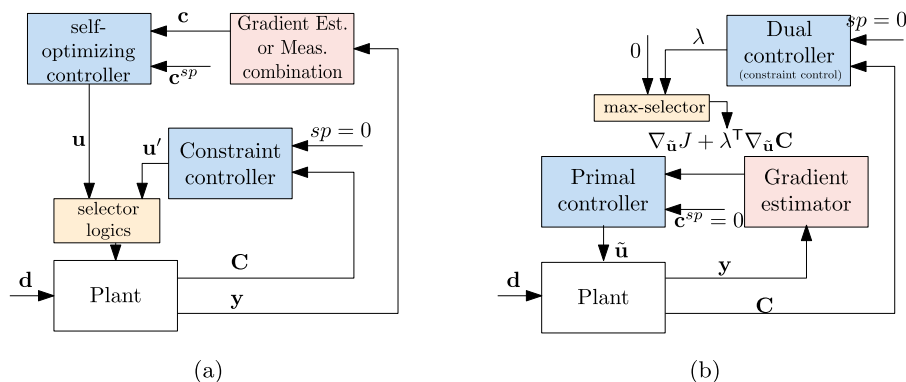


Fig. 3. Feedback Optimizing Control. (a) Paradigm 1: Region-based control, with active constraint control and self-optimizing control in each region and switching between the different active constraint regions. (b) Paradigm 2: Primal-Dual Feedback optimizing control structure.

- $n_{\tilde{u}}$ primal controllers where $\tilde{\mathbf{u}}$ is used to control the gradient of the Lagrangian $\nabla_{\tilde{\mathbf{u}}}\mathcal{L}(\lambda, \tilde{\mathbf{u}}, \mathbf{d})$ to a constant setpoint of zero thereby satisfying the stationarity condition (3a)
- n_c dual controllers where the dual variables λ are used to control the constraints $\mathbf{C}(\tilde{\mathbf{u}}, \mathbf{d})$ at their limiting value while ensuring that $\lambda \geq 0$ by using a max-selector, thereby ensuring primal feasibility (3b), dual feasibility (3c), and complementary slackness (3d).

Given Assumption 1, this ensures the satisfaction of the first-order optimality conditions by using feedback controllers. The general control structure under this paradigm is schematically represented in Fig. 3b.

Unlike the region-based approach, primal-dual control avoids the need for identifying the different active constraint regions *a priori*. Consequently this approach also avoids the need to explicitly design a switching strategy to handle changes in the active constraint regions as the operating conditions vary. The choice of MV-CV pairing is also rather straightforward for this case, where $\tilde{\mathbf{u}}$ is used to control $\nabla_{\tilde{\mathbf{u}}}\mathcal{L}(\lambda, \tilde{\mathbf{u}}, \mathbf{d})$, and the Lagrange multiplier corresponding to a constraint is used to control that constraint.

Note that the dual controllers control the constraint indirectly by updating λ , which occurs at a slower time scale relative to the primal controller (due to the cascade structure). Thus, in some cases it may be difficult to achieve tight control of the constraints and to avoid dynamic constraint violation one may need to use back-off, resulting in an economic loss, quantified by (4).

It can be seen that this approach requires online estimation of the steady-state cost gradient $\nabla_{\tilde{\mathbf{u}}}\mathcal{L}(\lambda, \tilde{\mathbf{u}}, \mathbf{d})$, as well as the constraint gradients $\nabla_{\tilde{\mathbf{u}}}\mathbf{C}(\tilde{\mathbf{u}}, \mathbf{d})$. Any of the model-based gradient estimation method reviewed in Section 3.2 or any model-free gradient estimation method reviewed in Section 4 can be used for this purpose. Note that unlike the region-based approach, primal-dual approach cannot use other self-optimizing variables such as a single measurement or a linear measurement combination. Therefore, primal-dual approach cannot be used if the cost and constraint gradient estimates are not available.

It is also important to note that the primal and dual controllers work together by gathering all the constraint and cost gradients to compute the gradient of the Lagrangian, and one is not independent of the other. This makes it less robust to faulty control loops. For example, if any control loop is broken, then this will affect the overall system, unlike in region-based control, where for example any given active constraint controller will still work perfectly fine in controlling that constraint even if some other control loop is broken. This is because in region-based control, each constraint is paired directly with an input. On the other hand, this makes region-based approach less suitable for large-scale systems than the primal-dual approach.

Distributed Feedback Optimizing Control Primal-dual approach is more amenable to distributed systems and large-scale processes that is made up of several unit operations or sub-processes, that are coupled in one form or the other. In many process applications, feedback controllers are typically designed in a decentralized fashion for small subprocesses /unit operations. If the overall process is coupled in one form or the other, then optimal operation of the local subsystems do not necessarily contribute to the overall optimal operation. The loss due to lack of coordination was also pointed out and quantified by Morari et al. (1980).

To address this issue, one can decompose the large-scale process into several smaller subprocesses or unit operations, and formulate local optimization problems with coupling constraints that couples the different subsystems together. This enables one to design feedback optimizing control structures for the local subsystems, and use a central coordinator to coordinate the different subsystems. Decomposing a large-scale system into smaller subsystems and considering the different subsystems locally makes it easier to design feedback control structures, which is also often done in practice. A distributed feedback optimizing control scheme based on the primal-dual feedback control approach was recently proposed by Krishnamoorthy (2021a), where the “optimizing” controlled variable for local subsystems are given as a function of the Lagrange multiplier of the coupling constraints. The primal controllers are designed locally for each subsystem based on the local cost and constraint gradient estimation. The dual controllers corresponding to the coupling constraints acts as the central coordinator and updates the Lagrange multipliers, as shown in Fig. 4. Together, this leads to overall system-wide asymptotic optimal operation. The coupling constraints typically arises from shared resources among the different unit operations or physical couplings (e.g.the outflow from one unit is the inflow to another). In the former case, the optimization problem is formulated as an optimal exchange problem (Boyd et al., 2011, Section 7.3), whereas in the latter case, the optimization problem is formulated as a consensus problem (Boyd et al., 2011, Section 7.1). The interested reader is referred to Krishnamoorthy (2021a) for more detailed description and analysis of the distributed feedback-based RTO scheme.

3. Model-based methods

In model-based techniques, we assume that we have access to a model of the process given by,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d}) \quad (15a)$$

$$\mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{d}) \quad (15b)$$

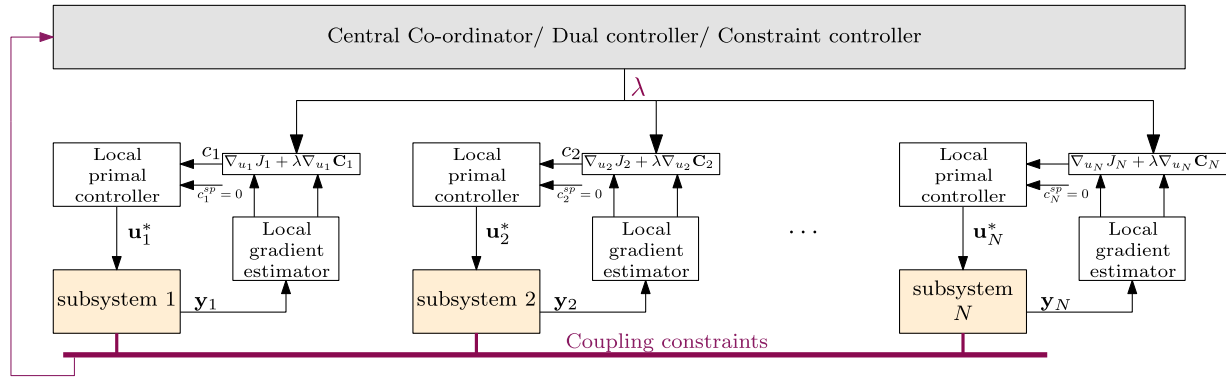


Fig. 4. A distributed feedback-based RTO framework using Lagrangian decomposition. The large-scale process is decomposed into N smaller subsystems, each optimized locally using feedback controllers. The central coordinator ensures system-wide optimal operation.

$$J_{model} = \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{d}) \tag{15c}$$

where $\mathbf{x} \in \mathbb{R}^{n_x}$ denotes the states, $\mathbf{u} \in \mathbb{R}^{n_u}$ denotes the unconstrained degrees of freedom, $\mathbf{d} \in \mathbb{R}^{n_d}$ denotes the disturbances, and $\mathbf{y} \in \mathbb{R}^{n_y}$ denotes the vector of measurements. $\mathbf{f} \in \mathcal{C}^2 : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \rightarrow \mathbb{R}^{n_x}$, $\mathbf{g} \in \mathcal{C}^2 : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \rightarrow \mathbb{R}^{n_y}$, and $\mathbf{h} \in \mathcal{C}^2 : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \rightarrow \mathbb{R}$ denotes the smooth functions that describe the system dynamics, measurements and the cost respectively. Note that the cost predicted by the model is denoted as $J_{model} = \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{d})$, whereas the true cost measurement from the plant is simply denoted as J , without any subscript. The unmeasured disturbances and model parameters are jointly denoted by \mathbf{d} . Furthermore, we assume that we have more measurements than disturbances, i.e. $n_y > n_d$.

As shown in Fig. 1, the model-based methods can be further classified into two categories, based on whether the models are used online or offline.

3.1. Models used offline for obtaining self-optimizing variables

In this class of methods, the controlled variables \mathbf{c} for the unconstrained degrees of freedom are chosen offline based on process insights extracted from the process models (15). Using offline analysis, the objective is to design the controlled variables \mathbf{c} , as well as their corresponding setpoints \mathbf{c}^{sp} .

As already mentioned, the ideal self-optimizing variable is the cost gradient $\mathbf{c} = \mathbf{J}_u$. However, obtaining an analytic expression for \mathbf{J}_u may be tedious, and even if an expression is found, one may not have measurements of all variables in the expression for \mathbf{J}_u . In addition, the cost gradient \mathbf{J}_u may also be susceptible to measurement errors. An alternative approach is therefore to start from the available measurement \mathbf{y} , and look for single measurements or measurement combinations that can be kept constant. To simplify the mathematical treatment, we usually only consider linear measurement combinations, $\mathbf{c} = \mathbf{H}\mathbf{y}$, where \mathbf{H} is a constant measurement selection matrix. Note that also the optimal setpoint \mathbf{c}^{sp} must be obtained in this case.

3.1.1. Direct loss evaluation

When deciding what to control using the unconstrained degrees of freedom, the simplest is to choose a subset of the available measurements $\mathbf{c} \subset \mathbf{y}$ as the self-optimizing controlled variables (CVs), which are controlled to some constant setpoint. The main idea here is to select all or a subset of available measurements as candidate controlled variables, and repeatedly evaluate the cost for the expected disturbance scenarios, when the different candidate controlled variables are kept constant. This analysis is performed offline for all the candidate CVs, and for all expected disturbances. Since there are n_u unconstrained degrees of freedom, a subset of

n_u measurements are evaluated as candidate CVs for expected disturbances. As such, this approach involves simulating the process several times for each set of candidate CV to evaluate the cost, and replacing the incumbent CV with the one that gives a lower cost. The candidate measurements with the lowest economic losses are then used as self-optimizing controlled variable for online control. This method is also known as *brute force approach*. Economic loss here may either be the average loss, or the worst case loss.

Note that, with this method one does not know how far one is from the true optimum, since we only evaluate the cost, and select the set of candidate CVs that provides the lowest observed cost. This is one of the earlier methods for finding the self-optimizing controlled variables for unconstrained optimization problems, that was used by Skogestad (2000), and was also studied by Larsson and Skogestad (2000) and Govatsmark and Skogestad (2005).

For processes with many candidate measurements, it may be tedious and time consuming to evaluate the economic loss for all the possible candidate controlled variables. A more numerically efficient approach is to use the “maximum-gain rule”, which uses the steady-state gain (denoted by \mathbf{G}) from the unconstrained degree of freedom to the candidate controlled variable (Halvorsen et al., 2003). Here, the gain matrix is scaled as shown below

$$\mathbf{G}' = \text{diag} \left\{ \frac{1}{\text{span}(c_i)} \right\} \mathbf{G}$$

where c_i is a set of candidate controlled variables. The maximum gain rule states that self-optimizing variables should be selected that maximizes the scaled gain matrix \mathbf{G}' , or more precisely maximize the minimum singular value of the scaled gain matrix (Halvorsen et al., 2003). The maximum gain rule can be used to select a few good alternative candidate self-optimizing variables. Brute force method can then be used to analyze the economic losses for the short-listed CV candidates. The interested reader is also referred to a recent survey paper on self-optimizing control by Jäschke et al. (2017) for a more detailed discussion of this approach.

3.1.2. Linear measurement combination

In this approach, instead of selecting a single measurement, the controlled variable is chosen as a linear combination of the available measurements

$$\mathbf{c} = \mathbf{H}\mathbf{y}$$

The objective is then to find the best measurement selection matrix \mathbf{H} using the process models offline. Most approaches are based on local linearization around some nominal optimal point.

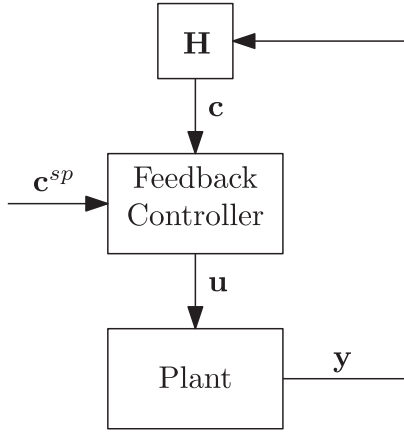


Fig. 5. Self-optimizing control using linear measurement combination as the optimal controlled variable, where the self-optimizing controlled variables are identified using the models offline (Alstad and Skogestad, 2007; Alstad et al., 2009).

The two most popular approaches to find the optimal measurement combination are the nullspace method proposed by Alstad and Skogestad (2007) and the exact local method proposed by Halvorsen et al. (2003) and further developed by Alstad et al. (2009) and Yelchuru and Skogestad (2012). Fig. 5 shows the block diagram for self-optimizing control.

Remark 1. Note that in the case of single measurements selected using direct loss evaluation (cf. Section 3.1.1) the elements of the \mathbf{H} matrix would comprise of 0s and 1s.

Nullspace method The nullspace method, is the simplest approach for finding the optimal selection matrix \mathbf{H} for the case with no implementation error. This method assumes that the number of measurements $n_y \geq n_u + n_d$.

Let the disturbance variation around some nominal point \mathbf{d}_0 be denoted as $\delta \mathbf{d} := \mathbf{d} - \mathbf{d}_0$, and the corresponding change in the optimal measurement be denoted as $\delta \mathbf{y}^* := \mathbf{y}^* - \mathbf{y}_0^*$. The optimal sensitivity matrix, which describes how the optimal value varies with the disturbance, is given by

$$\mathbf{F} = \frac{\partial \mathbf{y}^*}{\partial \mathbf{d}} \quad (16)$$

The optimal selection matrix \mathbf{H} is selected such that it is in the nullspace of the optimal sensitivity matrix \mathbf{F} , i.e.

$$\mathbf{H}\mathbf{F} = \mathbf{0} \quad (17)$$

Alstad (2005, Chapter 3) proved that computing the selection matrix \mathbf{H} such that $\mathbf{H}\mathbf{F} = \mathbf{0}$ leads to zero loss for small disturbance change $\Delta \mathbf{d}$. The setpoint for the optimal measurement combination \mathbf{c} is often computed using the nominal optimal measurement, i.e. $\mathbf{c}^{sp} = \mathbf{H}\mathbf{y}_0^*$.

Exact local method: This method is an extension of the nullspace method which takes into account the measurement and implementation error, and avoids the limitation that $n_y \geq n_u + n_d$. The starting point is to approximate the nonlinear economic optimization problem (5) with a constrained quadratic programming (QP) problem that minimizes the average loss. This is based on Taylor series expansion of the cost around the nominal optimum \mathbf{u}_0^* for a given disturbance \mathbf{d}

$$J(\mathbf{u}, \mathbf{d}) \approx J(\mathbf{u}_0^*, \mathbf{d}) + \underbrace{\mathbf{J}_u^T \delta \mathbf{u}}_{=0} + \frac{1}{2} \delta \mathbf{u}^T \mathbf{J}_{uu} \delta \mathbf{u} \quad (18)$$

from which the loss function can be written as

$$Loss = J(\mathbf{u}, \mathbf{d}) - J(\mathbf{u}_0^*, \mathbf{d}) = \frac{1}{2} \|\mathbf{z}\|^2 \quad (19)$$

where $\mathbf{z} := \mathbf{J}_{uu}^{1/2} \delta \mathbf{u}$.

Using the linearized measurement model,

$$\delta \mathbf{y} = \underbrace{\frac{\partial \mathbf{g}}{\partial \mathbf{u}}}_{\mathbf{G}^y} \delta \mathbf{u} + \underbrace{\frac{\partial \mathbf{g}}{\partial \mathbf{d}}}_{\mathbf{G}_d^y} \delta \mathbf{d} \quad (20)$$

and

$$\mathbf{c} = \mathbf{H}\mathbf{y}_m = \mathbf{H}\mathbf{y} + \mathbf{H}\mathbf{e}_y \quad (21)$$

we can express

$$\mathbf{z} = \mathbf{J}_{uu}^{1/2} (\mathbf{H}\mathbf{G}^y)^{-1} \mathbf{H} \underbrace{\begin{bmatrix} \mathbf{F}\mathbf{W}_d & \mathbf{W}_{e_y} \end{bmatrix}}_{:=\mathbf{Y}} \begin{bmatrix} -\mathbf{d}' \\ -\mathbf{e}_y' \end{bmatrix} \quad (22)$$

where $\mathbf{d} = \mathbf{d}'\mathbf{W}_d$ is the normalized disturbance, $\mathbf{e}_y = \mathbf{e}_y'\mathbf{W}_{e_y}$ is the normalized implementation error, and \mathbf{W}_d and \mathbf{W}_{e_y} are diagonal scaling matrices for the expected values of the disturbance and implementation error, respectively.

Since $Loss = \frac{1}{2} \|\mathbf{z}\|^2$, from (22), we can see that \mathbf{z} is minimized if $\mathbf{H}\mathbf{Y}$ is minimized with $\mathbf{J}_{uu}^{1/2} (\mathbf{H}\mathbf{G}^y)^{-1} = \mathbf{I}$. Therefore, the minimization of the average loss can be achieved by minimizing the following,

$$\min_{\mathbf{H}} \|\mathbf{H}\mathbf{Y}\|_F \quad (23)$$

$$\text{s.t. } \mathbf{H}\mathbf{G}^y = \mathbf{J}_{uu}^{1/2}$$

from which \mathbf{H} can be obtained analytically as,

$$\mathbf{H}^T = (\mathbf{Y}\mathbf{Y}^T)^{-1} \mathbf{G}^y \quad (24)$$

See Alstad et al. (2009) for a detailed derivation of the exact local method.

Remark 2. It is worth noting that exact local method is the only approach that explicitly takes into account measurement noise and implementation error.

Note that the optimal sensitivity matrix (16) may be determined numerically by perturbing the disturbances and re-optimizing. Alternatively, one can use the analytical expression. Assuming that we in \mathbf{y}^* include in addition to outputs \mathbf{y} also the inputs \mathbf{u} , the analytical expression for the sensitivity matrix becomes

$$\mathbf{F} = \begin{bmatrix} \frac{\partial \mathbf{y}^*}{\partial \mathbf{d}} & \frac{\partial \mathbf{u}^*}{\partial \mathbf{d}} \end{bmatrix}^T = \begin{bmatrix} \mathbf{G}_d^y - \mathbf{G}^y \mathbf{J}_{uu}^{-1} \mathbf{J}_{ud} \\ \mathbf{J}_{uu}^{-1} \mathbf{J}_{ud} \end{bmatrix} \quad (25)$$

where

$$\mathbf{J}_{uu} := \frac{\partial^2 \mathbf{h}}{\partial \mathbf{u}^2} \quad \mathbf{J}_{ud} := \frac{\partial^2 \mathbf{h}}{\partial \mathbf{u} \partial \mathbf{d}}$$

The nullspace and the exact local methods are based on local linearization around some nominal optimal point. As a result, the steady-state loss increases, as the process is operated far away from the nominal optimal conditions. In order to extend the economic performance to be globally acceptable, global self-optimizing control methods have also been proposed, see for e.g. Ye et al. (2015, 2017). Ye et al. (2012) also proposed to approximate the necessary conditions of optimality using regression techniques such as least squares or artificial neural networks, which can then be used as self-optimizing controlled variables. In recent years, there have been several developments that focus on using measurement combination as self-optimizing variables, leading to its own survey article (Jäschke et al., 2017).

3.1.3. Neighboring extremal control (NE)

In this approach, the cost gradient around the nominal optimal conditions is estimated from the variations in \mathbf{d} around the nominal operating point where the gradient is zero. The change in the cost gradient $\delta\mathbf{J}_u$ resulting from the parametric variations $\delta\mathbf{d}$ can be expressed as

$$\delta\mathbf{J}_u = \mathbf{J}_{uu}\delta\mathbf{u} + \mathbf{J}_{ud}\delta\mathbf{d} \quad (26)$$

Since the parametric variations $\delta\mathbf{d}$ may be unknown, it is inferred from the input and measurement variations $\delta\mathbf{u} := \mathbf{u} - \mathbf{u}_0$ and $\delta\mathbf{y} := \mathbf{y} - \mathbf{y}_0$ respectively. This is obtained by using the linearized measurement model (20). Hence,

$$\delta\mathbf{d} = (\mathbf{G}_d^y)^\dagger (\delta\mathbf{y} - \mathbf{G}^y\delta\mathbf{u}) \quad (27)$$

Substituting (27) in (26) results in,

$$\delta\mathbf{J}_u = \mathbf{J}_{ud}(\mathbf{G}_d^y)^\dagger \delta\mathbf{y} + [\mathbf{J}_{uu} - \mathbf{J}_{ud}(\mathbf{G}_d^y)^\dagger \mathbf{G}^y] \delta\mathbf{u} \quad (28)$$

which is the gradient at nominal conditions. The estimated gradient (at nominal conditions) is then controlled to a constant setpoint of zero.

Comparing the optimal sensitivity matrix \mathbf{F} in (25) with (28), it can be seen that the nullspace method and the neighboring extremal control approaches are equivalent (François et al., 2014). That is (28) can equivalently be written as

$$\mathbf{c} = \delta\mathbf{J}_u = \mathbf{H} \begin{bmatrix} \delta\mathbf{y} \\ \delta\mathbf{u} \end{bmatrix} \quad (29)$$

where

$$\mathbf{H} = \begin{bmatrix} \mathbf{J}_{ud}(\mathbf{G}_d^y)^\dagger & \mathbf{J}_{uu} - \mathbf{J}_{ud}(\mathbf{G}_d^y)^\dagger \mathbf{G}^y \end{bmatrix}$$

and from (25) one can see that $\mathbf{H}\mathbf{F} = \mathbf{0}$, which is the same as in the nullspace method.

Remark 3. The connection between the nullspace method and the neighboring extremal method shows that, although the nullspace method aims to find a linear measurement combination as the self-optimizing variable, it can also be seen as estimating the cost gradient (i.e. ideal-self optimizing variable). In other words, the nullspace method gives $\mathbf{J}_u = \mathbf{0}$ at nominal conditions, see also (Jäschke and Skogestad, 2011, Appendix B).

Although the standard neighboring extremal control (Gros et al., 2009) does not take into account measurement noise, variations of this approach that considers the measurement noise has also been proposed (de Oliveira et al., 2015).

3.2. Online model-based gradient estimation

3.2.1. Two-step approach

This approach is closely related to the traditional two-step RTO approach, where the steady-state cost gradient is computed analytically using the model Eq. (15). The disturbances \mathbf{d} are estimated using the measurements \mathbf{y}_{meas} using any parameter estimation algorithm.

$$\begin{bmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{d}} \end{bmatrix} = K_e(\mathbf{y}_{meas} - \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{d})) \quad (30)$$

with $\hat{\mathbf{d}}$ being the estimated parameter and K_e is the estimation gain. Once the model parameters $\hat{\mathbf{d}}_k$ are estimated, the analytical Jacobian can be evaluated using the updated model. The estimated gradient $\hat{\mathbf{J}}_{u,model}$ is then driven to a constant setpoint of zero using any feedback control law. This approach is schematically shown in Fig. 6a. The two-step approach of estimating model parameters using a dynamic model in the context of feedback optimizing control was proposed by Adetola and Guay (2007).

3.2.2. Feedback RTO using transient measurements (FRTO)

This approach proposed by Krishnamoorthy et al. (2019c) estimates the steady-state cost gradient using a nonlinear dynamic model and the process measurements \mathbf{y}_{meas} by linearizing the nonlinear dynamic model from the cost to the inputs. Any combined state and parameter estimation scheme (e.g. extended Kalman filter) may be used to estimate the states $\hat{\mathbf{x}}$ and the unmeasured disturbances $\hat{\mathbf{d}}$ using the dynamic model of the plant and the measurements \mathbf{y}_m . Once the states and unmeasured disturbances are estimated, (15a) and (15c) are linearized to obtain a local linear dynamic model from the inputs \mathbf{u} to the objective function J ,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (31)$$

$$J_{model}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$$

where $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$, $\mathbf{C} \in \mathbb{R}^{1 \times n_x}$ and $\mathbf{D} \in \mathbb{R}^{1 \times n_u}$. The system matrices are evaluated around the current estimates $\hat{\mathbf{x}}$ and $\hat{\mathbf{d}}$,

$$\begin{aligned} \mathbf{A} &= \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d})}{\partial \mathbf{x}} & \mathbf{B} &= \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d})}{\partial \mathbf{u}} \\ \mathbf{C} &= \frac{\partial \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{d})}{\partial \mathbf{x}} & \mathbf{D} &= \frac{\partial \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{d})}{\partial \mathbf{u}} \end{aligned}$$

Assuming that \mathbf{A} is invertible, the corresponding steady-state gradient is then given as

$$\hat{\mathbf{J}}_{u,model} = -\mathbf{C}\mathbf{A}^{-1}\mathbf{B} + \mathbf{D} \quad (32)$$

This approach is schematically shown in Fig. 6b. It is also worth noting that (32) was also used by Bamberger and Isermann (1978), Garcia and Morari (1981) among others to estimate the steady-state gradient from linear dynamic models (cf. Remark 4). The estimated gradient is then driven to zero using any feedback controller. The reader is referred to Krishnamoorthy et al. (2019c) for more detailed discussions, as well as comparison with the traditional RTO framework. This approach was successfully tested in a wide range of case examples, see for example Krishnamoorthy et al. (2019c), Bonnowitz et al. (2018), Krishnamoorthy et al. (2018b), Krishnamoorthy et al. (2019b).

4. Model-free methods

In this section, we review different model-free methods to achieve optimal operation using feedback control. In Section 3.2, we saw how process models can be used online to estimate the steady-state cost gradient, which can be controlled to a constant setpoint of zero. Model-free methods on the other hand involve estimating the steady-state cost gradient directly from the cost measurement without the need for detailed process models.

The underlying principle of model-free gradient estimation is via online experimentation. Simply put, the inputs are perturbed, and the corresponding change in the cost measurement is observed, which is then used to estimate the steady-state cost gradient. Therefore, all the model-free methods reviewed below assume that direct cost measurements J are available. Below we review different model-free gradient estimation, and for the sake of simplicity we consider only the unconstrained case.

4.1. Finite-difference (FD)

Finite-difference approach is probably the oldest and the most straightforward approach to estimate the steady-state cost gradient. Here, the inputs are perturbed by a small value $\Delta\mathbf{u}$ around the current operating point \mathbf{u}_k over a time period T ,

$$\mathbf{u}(t) = \begin{cases} \mathbf{u}_k & 2kT \leq t < (2k+1)T \\ \mathbf{u}_k + \Delta\mathbf{u} & (2k+1)T \leq t < (2k+2)T \end{cases} \quad (33)$$

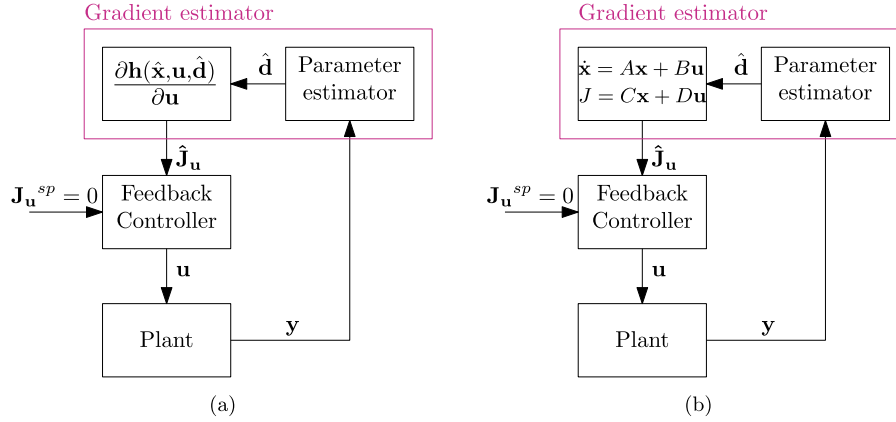


Fig. 6. Online model-based gradient-estimation methods: (a) Gradient from adapted model using the two-step approach. (b) Feedback RTO (Krishnamoorthy et al., 2019c).

The time period T is chosen such that the system reaches steady-state within the time period. The gradients are then calculated by measuring the corresponding variation in the cost measurement.

$$\hat{\mathbf{J}}_{\mathbf{u}} = \frac{\Delta J}{\Delta \mathbf{u}} = \frac{J((2k+2)T) - J((2k+1)T)}{\Delta \mathbf{u}} \quad (34)$$

Finite difference method is a popular approach due to its simplicity and ease of implementation, and has been used in several feedback optimizing control methods such as NCO-tracking control (François et al., 2005; Jäschke and Skogestad, 2011) and hill-climbing control (Kumar and Kaistha, 2014; Shinskey, 1996). Finite difference approach has also been used in other numerical optimization based methods that use cost gradients such as the integrated system optimization and parameter estimation (ISOPE) algorithm (Roberts and Williams, 1981) and modifier adaptation (Marchetti et al., 2009).

The finite-difference approach has shown to provide gradient estimates with sufficient accuracy for systems with relatively fast dynamics and high signal-to-noise ratio (SNR). However, it is known to be inefficient for processes with long settling times (which requires large T), and measurements corrupted by large noise (which requires large $\Delta \mathbf{u}$).

In addition to the classical finite difference approach mentioned above, other variants of finite-difference approximation have also been used in literature. For example, Roberts (2000), Gao and Engell (2005) and Rodger and Chachuat (2011) used Broyden's formula to estimate the steady-state gradient from current and past measurement, without the need for additional perturbations. Brdyś and Tatjewski (1995) proposed an alternative variant, where past operating points are used to calculate the gradients without additional perturbations.

4.2. Classical extremum seeking control

Extremum seeking control, as the name suggests, is a control system that is used to seek and maintain the extremum value of a static map between the input and the cost function (Ariyur and Krstic, 2003). This approach dates back to the 1950s with the work of Draper and Li (1951), where extremum seeking control was studied under the context of adaptive control. This method gained huge popularity since the work of Krstić and Wang (2000), and remains to be a popular approach to this day. Extremum seeking control has been applied to a wide array of application domains including, but not limited to, process control, aerospace, automotive, robotics, solar power, wind power, oil and gas, medical and biomedical applications etc., to name a few. Extremum seeking control has also been used in controller design such as optimal tuning of PID controllers

(Killingsworth and Krstic, 2006). Since 2000s, there has been several advancements in extremum seeking (ES) methods including, least square-based ES (Hunnkens et al., 2014), sliding-mode ES (Fu and Özgüner, 2011; Pan et al., 2003), greedy ES (Trollberg and Jacobsen, 2016), discrete-time ES (Choi et al., 2002), newton-based ES (Ghaffari et al., 2012), Lie-bracket approximation based ES (Dürr et al., 2013) to name a few.

In the classical extremum seeking approach, a slow periodic dither signal in the form of a sinusoidal wave $a \sin \omega t$ is superimposed on to the input signal,

$$u(t) = \hat{u} + a \sin \omega t \quad (35)$$

The frequency ω of the sinusoidal perturbation is chosen to be slow such that the dynamic plant appears as a static map. This induces a periodic response in the cost measurement with the same frequency ω . A high-pass filter with cut-off frequency ω_h is used to remove the static bias (also known as the DC-component) from the cost measurement,

$$\dot{\eta} = -\omega_h \eta + \omega_h J \quad (36)$$

$(J - \eta)$ is then correlated with the input perturbation and the static bias of the product of the two sinusoids is extracted using a low-pass filter with cut-off frequency ω_l ,

$$\dot{\xi} = -\omega_l \xi + \omega_l (J - \eta) a \sin \omega t \quad (37)$$

from which the cost gradient can be obtained as

$$\hat{\mathbf{J}}_{\mathbf{u}} = \frac{a^2}{2} \xi \quad (38)$$

The estimated gradient is then driven to a constant setpoint of zero using an integral controller. This is schematically represented in Fig. 7a.

As mentioned above, the dither frequency must be chosen relatively small such that the plant dynamics do not interfere with the extremum seeking scheme, and in addition the integral gain must be small such that the convergence to the optimum does not interfere with the sinusoidal perturbation. Hence this approach has a clear timescale separation between

- plant dynamics (fastest)
- sinusoidal perturbation (medium)
- convergence to the optimum (slow)

In order to extend the dither-demodulation approach to the multivariable case, each input channel is perturbed individually. The perturbation frequencies must be chosen such that the different frequency components are unique, in order to ensure that unique persistence of excitation condition is satisfied i.e. $\omega_i \neq \omega_j$,

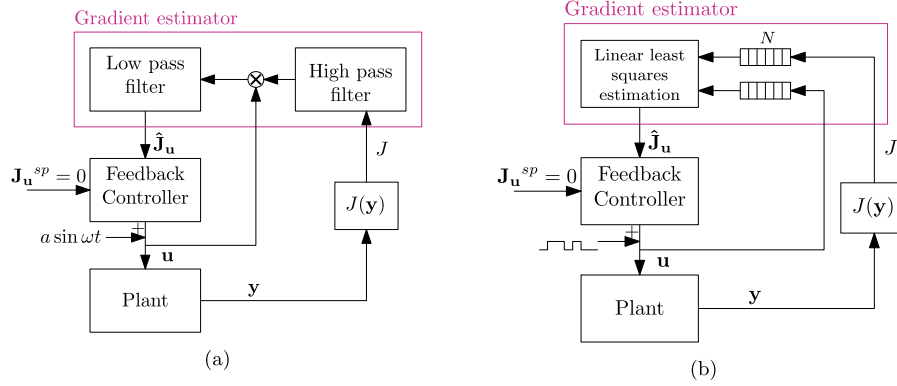


Fig. 7. Model-free gradient estimation methods: (a) classical extremum seeking control (Krstić and Wang, 2000). (b) Least squares-based extremum seeking control (Hunneken et al., 2014).

$2\omega_i \neq \omega_j$, $\omega_i + \omega_j \neq \omega_k$ for any distinct i , j , and k (Ghaffari et al., 2012; 2015).

Although extremum seeking control has gained popularity in electrical and mechanical systems, for most chemical and biochemical processes, the timescale separation requirement leads to prohibitively slow convergence to the optimum (typically convergence to the optimum in the range of several hours to days).

4.3. Least squares-based gradient estimation

In this approach, the steady-state cost gradient is estimated by using a first-order least squares fit, and was presented in the context of extremum seeking control by Hunneken et al. (2014). The last N samples of input \mathbf{u} and cost J are used to fit a local linear static model of the form,

$$J = \hat{\mathbf{J}}_{\mathbf{u}}^T \mathbf{u} + m \quad (39)$$

where m is the intercept/bias. At the current sample time k , let $\mathbf{Y} = [J_k, J_{k-1}, \dots, J_{k-N+1}]^T$ denote the vector of the past N cost measurements, and $\mathbf{U} = [\mathbf{u}_k, \dots, \mathbf{u}_{k-N+1}]^T$ be the vector of the past N samples of the input data. A moving window of fixed length N is then used to estimate the gradient using a linear least squares estimation

$$\hat{\theta} = \arg \min_{\theta} \|\mathbf{Y} - \Phi^T \theta\|_2^2 \quad (40)$$

to which the analytical solution is given by

$$\hat{\theta} = [\Phi^T \Phi]^{-1} \Phi^T \mathbf{Y} \quad (41)$$

where $\theta = [\hat{\mathbf{J}}_{\mathbf{u}}^T, m]^T$ and $\Phi = [\mathbf{U}, \mathbf{1}]^T$. This approach is schematically represented in Fig. 7b.

The application of the least squares method requires that $N > n_u$. In theory, this method does not require a constant perturbation using additional dither signal. However, a small perturbation in practice is recommended in order to track changes in the optimum and also to avoid an ill-conditioned problem in (41).

Instead of solving a linear least squares problem using (41), one can also solve a recursive least squares problem with forgetting factor to estimate $\theta = [\hat{\mathbf{J}}_{\mathbf{u}}^T, m]^T$ as described by Chioua et al. (2016), Dewasme et al. (2011).

4.4. Kalman filter-based gradient estimation

In this approach, a Kalman filter is used to estimate the steady-state cost gradient. Here, the underlying assumption is to fit a local linear static model (39) around the current operating point, which is the same as in the least squares approach (cf. Section 4.3). Instead of using least squares estimation, Henning et al. (2008) proposed to use an extended Kalman filter (EKF), where the two unknowns $x_1 = \mathbf{J}_{\mathbf{u}}$ and $x_2 = m$ are the states of the Kalman filter. A

discrete time model for the two parameters is given by,

$$\begin{bmatrix} \hat{\mathbf{J}}_{\mathbf{u}_{k+1}} \\ m_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{J}}_{\mathbf{u}_k} \\ m_k \end{bmatrix} + \mathbf{w}_k \quad (42)$$

For the system (42) to be observable, two distinct measurements of the input-cost pairs at two different times are required (Henning et al., 2008; Gelbert et al., 2012). Consequently, the measurement model is given by,

$$\begin{bmatrix} J_k \\ J_{k-i} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_k & 1 \\ \mathbf{u}_{k-i} & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{J}}_{\mathbf{u}_k} \\ m_k \end{bmatrix} + \mathbf{v}_k \quad (43)$$

where the input-cost measurement data at the current time step k and the measurement at time step $k-i$ are used, where i can be chosen by looking at the observability gramian as explained by Gelbert et al. (2012). The vectors \mathbf{w}_k and \mathbf{v}_k denote Gaussian white noises. An extended Kalman filter can now be used to estimate the "states" $\hat{\mathbf{J}}_{\mathbf{u}}$ and m using the "system model" (42) and the "measurement model" (43).

4.5. Gradient estimation using fast Fourier transform (FFT)

So far, we can see that the model-free gradient estimation approaches involve perturbing the input with additional dither signals, and the effect of the input perturbation on the cost is used to estimate the steady-state cost gradient. A natural and powerful approach to analyze the effect of periodic perturbations in any signal is to use the Fourier transform for spectral analysis, which tells us what is the contribution of each frequency component present in the signal. Therefore, one could periodically perturb the inputs and simply analyze the frequency spectrum of the cost measurement at the input perturbation frequency to obtain the cost gradient. In other words, the amplitude spectrum of the cost measurement provided by the fast Fourier transform (FFT) is equivalent to the magnitude of the cost gradient at the input perturbation frequency.

FFT is a fast, easy and robust numerical approach to extract the amplitude spectrum at different frequencies. This makes it a very favorable approach for multivariable systems, where the different inputs are perturbed with periodic signals with unique frequency components. The amplitude spectrum of the cost measurement at the different frequency components then provides the cost gradients with respect to each input. This intuitive nature of FFT has led to its use in a few engineering applications, see for example Corti et al. (2014) and Beaudoin et al. (2006). Krishnamoorthy (2021b) formalized the FFT-based extremum seeking scheme and analyzes its properties under a static map setting.

In fact, the classical extremum seeking control presented in Section 4.2 can be seen as a special case of the Fourier transform,

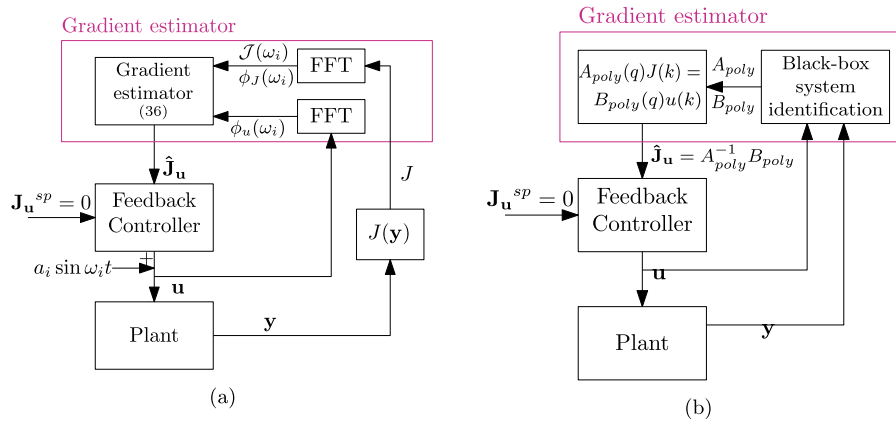


Fig. 8. Model-free gradient estimation methods: (a) Gradient estimation using Fast Fourier Transform (Krishnamoorthy, 2021b; Corti et al., 2014) (b) Gradient estimation from black-box system identification, e.g. ARX (Bamberger and Isermann, 1978; Garcia and Morari, 1981).

where the cost measurement is demodulated with a sinusoidal signal of the same frequency as the input perturbation, instead of different frequencies.

In this approach, each input is perturbed with a unique periodic sinusoidal signal $a_i \sin \omega_i t$. Consequently, the cost measurement is a function of all the sinusoidal frequencies, which is extracted by performing FFT over a sliding window of fixed length with N past data samples (similar to the least squares method presented in Section 4.3). To perform FFT, the cost measurement has to be detrended, such that it has zero mean. The detrending may be performed using any suitable method, for example the moving average filter. Note that this is analogous to using the high-pass filter in the classical extremum seeking control (cf. Section 4.2) to remove the static bias. The discrete Fourier transform then extracts the different frequency components of the detrended cost measurement J_0

$$\mathcal{J}(\omega_l) = \sum_{k=0}^{N-1} J_0(k) e^{-j \frac{2\pi}{N} l k} \quad \forall l = 0, \dots, N-1 \quad (44)$$

Note that the product of the complex exponential function and the detrended cost signal is analogous to the demodulation of the dither and the cost measurement in the classical extremum seeking control (cf. Section 4.2).

The magnitude of the gradient of the cost with respect to the i th input, perturbed using ω_i can then be obtained by looking at the amplitude spectrum $|\mathcal{J}(\omega_i)|$. Since the amplitude spectrum $|\mathcal{J}(\omega_i)| > 0$, to determine the sign of the cost gradient, the phase spectrum of the cost $\phi_J(\omega_i)$ with respect to the phase spectrum of the input signal $\phi_{u_i}(\omega_i)$ is used. This is schematically shown in Fig. 8a.

Together, the gradient of the cost w.r.t to the i th input is then estimated as follows:

$$\hat{J}_{u_i} = \frac{\widehat{\partial J}}{\partial u_i} = \frac{2}{a_i} |\mathcal{J}(\omega_i)| \operatorname{sgn} \left[\frac{\phi_J(\omega_i)}{\phi_{u_i}(\omega_i)} \right] \quad \forall i = 1, \dots, n \quad (45)$$

Alternatively, the estimated gradient can be computed directly as

$$\hat{J}_{u_i} = \frac{\widehat{\partial J}}{\partial u_i} = \frac{\operatorname{Re}(\mathcal{J}(\omega_i))}{\operatorname{Re}(\mathcal{U}_i(\omega_i))} \quad \forall i = 1, \dots, n \quad (46)$$

where $\operatorname{Re}(\cdot)$ is the real part of a complex number, and $\mathcal{U}_i(\omega)$ is the discrete Fourier transform of the i th input.

Much like any multivariable model-free gradient estimation scheme, the input perturbation frequencies must be unique, and

should not lie in the harmonics of other dither frequencies, i.e. $\omega_i \neq \omega_j$, $2\omega_i \neq \omega_j$, $\omega_i + \omega_j \neq \omega_k$ for any distinct i, j , and k .

It is also important to note that accuracy of the cost gradient at any particular frequency is sensitive to the choice of N . This is because the discrete Fourier transform treats the data window of length N as if it is periodic and produces only $l = 1, \dots, N$ discrete frequency components in (44). Therefore, if the exact perturbation frequency is not part of the discrete frequency array $l = 1, \dots, N$, this leads to inaccurate gradient estimation at the perturbation frequencies. To avoid this, the minimum length N is given by the least common multiplier of the different perturbation time periods $2\pi/\omega_i$. For a more detailed description and analysis of this approach, see Krishnamoorthy (2021b).

4.6. Gradient from multiple units

As mentioned earlier, model-free gradient estimation methods require perturbing the input in order to estimate the gradient. However, in some applications, these perturbations are not desired, especially since when the effect of the perturbations are carried over to downstream processes. For example, in continuous processes, although some perturbations may be tolerated in unit processes in order to optimize the process, the perturbations may not be accepted at the product supply to the customer.

In processes with multiple units, one can carefully design the perturbations, such that the overall perturbations cancel each other. One such approach was presented by Srinivasan (2007), where in the presence of multiple units, the inputs to the units differ by an offset Δ ,

$$\begin{aligned} u_i &= u_{i,k} + \frac{\Delta}{2} \\ u_j &= u_{j,k} - \frac{\Delta}{2} \end{aligned}$$

and the gradient is then estimated using the finite difference method (cf. Section 4.1).

$$\hat{J}_{\mathbf{u}} = \frac{J_i - J_j}{\Delta} \quad (47)$$

Similar synchronization methods for extremum seeking control were also recently presented by Pavlov et al. (2017) and Silva and Pavlov (2020).

4.7. Gradient estimation using transient measurements

A common trait in all the model-free gradient estimation methods presented so far is that it requires the assumption that the dynamic plant acts like a static map between the input and the

cost. In order for this assumption to be valid, the input perturbation must be much slower than the plant dynamics, such that the dynamic plant can be approximated as a static map. Furthermore, the controller gain to drive the steady-state gradient to zero must be sufficiently small such that the convergence to the optimum is much slower than the perturbation signal. In summary, this means that the overall convergence rate is about two orders of magnitude slower than the original plant dynamics (Krstić, 2000; Tan et al., 2010).

Although this is not a major bottleneck for many electro-mechanical systems, for many chemical and biochemical processes, the settling times are often in the range of minutes to hours. The timescale separation thus leads to prohibitively slow convergence. Despite the very appealing characteristic of not requiring a detailed model, many model-free gradient estimation methods may therefore be impractical for real-time optimization in the chemical process industry.

In order to address this issue, one potential solution is to explicitly include the plant dynamics in the model-free gradient estimation scheme. The use of measurements to repeatedly identify a black-box local linear dynamic model around the current operating point for online optimization of slow chemical processes was first proposed by Bamberger and Isermann (1978). Here, black-box system identification models such as ARX models are repeatedly identified online. The cost gradient is then estimated from the identified black-box models, which is driven to a constant set-point of zero. This approach was further analyzed by Garcia and Morari (1981), where the authors recursively identified local linear dynamic models to estimate the steady-state gradient and drive the process to its optimum using a gradient descent algorithm. Later in 1989, McFarlane and Bacon (1989) presented an empirical strategy for open-loop online optimization using black-box ARX models similar to the one proposed by Bamberger and Isermann (1978) and Garcia and Morari (1981). As such, the method proposed by Bamberger and Isermann (1978); Garcia and Morari (1981) and McFarlane and Bacon (1989) can be seen as a *dynamic* variant of extremum seeking control for Hammerstein plants, where the cost measurement is used to repeatedly identify a local linear *dynamic* model, which is then used to estimate the steady-state gradient (cf. (32)). Model-free gradient estimation using transient measurements by identifying local linear dynamic models have also recently been used in the context of modifier adaptation (Vaccari et al., 2020; Oliveira-Silva et al., 2021).

In this approach, black-box ARX model of the form

$$J(t) = -a_1 J(t-1) - \dots - a_{n_a} J(t-n_a) + b' u(t-1) + \dots + b'_{n_b} u(t-n_b) + e(t) \quad (48)$$

are repeatedly identified using process measurements. The inputs are perturbed using pseudo-random binary signal (PRBS) and the perturbation frequency could be chosen in the same time scale as the plant dynamics, leading to one order of magnitude faster convergence to the optimum compared to classical sinusoidal extremum seeking scheme that requires two orders of magnitude time scale separation.

The ARX model is identified repeatedly by solving the least squares problem

$$\hat{\theta} = \arg \min_{\theta} \|\psi - \Phi^T \theta\|_2^2 \quad (49)$$

where ψ , θ and Φ are given by the expressions

$$\psi = J$$

$$\Phi = [-J(t-1) \dots -J(t-n_a) \quad u(t-1) \dots u(t-n_b)]^T$$

$$\theta = [a_1 \quad \dots \quad a_{n_a} \quad b'_1 \quad \dots \quad b'_{n_b}]$$

Introducing the notation,

$$A_{poly}(q) = 1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}$$

and

$$B_{poly}(q) = b'_1 q^{-1} + \dots + b'_{n_b} q^{-n_b}$$

with q^{-1} being the unit delay operator, the gradient is then estimated as,

$$\hat{J}u = A_{poly}^{-1} B_{poly} \quad (50)$$

Remark 4. It is interesting to note that the identified ARX polynomials $A_{poly}(q)$ and $B_{poly}(q)$, when converted to continuous time state-space system¹ as shown below,

$$\dot{x} = Ax + Bu$$

$$J = Cx + Du$$

results in the expression

$$\hat{J}u = (-CA^{-1}B + D)$$

for the steady-state gradient, which is the same as (32) in the feedback RTO using transient measurements approach (cf. Section 3.2.2).

Alternatively, one could also repeatedly identify an ARMAX model or any other black-box model to estimate the steady-state cost gradient. For example, Golden and Ydstie (1989) used a second order Hammerstein model of the form

$$J(t) = -a_1 J(t-1) - \dots - a_{n_a} J(t-n_a) + b' u(t-1) + \dots + b'_{n_b} u(t-n_b) + c' u^2(t-1) + \dots + c'_{n_b} u^2(t-n_b) + e(t)$$

However, with such high order ARX, ARMAX models or second order Hammerstein models such as the one used by Golden and Ydstie (1989), the number of parameters that needs to be repeatedly estimated increases. If the excitation of the process is not sufficient, then all the black-box model parameters may not be estimated accurately. In the context of real-time optimization, this is especially a problem as the system approaches its optimum, where the steady-state relation between the input and the cost is typically flat. This challenge is illustrated using simple counter examples in Krishnamoorthy, 2019, Chapter 5.

One simple solution to this problem is to simply turn off the gradient estimation once the plant reaches its optimum, as done by Bamberger and Isermann (1978), McFarlane and Bacon (1989). However, in practice, it is desirable to keep estimating the gradient even after reaching the optimum. This is to ensure that the changes in the optimum are tracked.

Alternatively, if the nominal linear dynamics are known, this can be fixed, such that we only estimate the unknown local steady-state effect.

$$J = J_u \underbrace{\left(\frac{b_1 q^{-1} + \dots + b_{n_b} q^{-n_b}}{1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}} \right)}_{\text{fixed}} u \quad (51)$$

Fixing the linear dynamic part enables us to effectively use transient measurements, thereby avoiding the steady-state wait time issue (Krishnamoorthy, 2019, Chapter 5). In this case, the least squares problem (49) is solved with

$$\psi = J(t) + a_1 J(t-1) + \dots + a_{n_a} J(t-n_a)$$

$$\Phi = b_1 u(t-1) + \dots + b_{n_b} u(t-n_b)$$

¹ for example using `idss` and `d2c` command in MATLAB

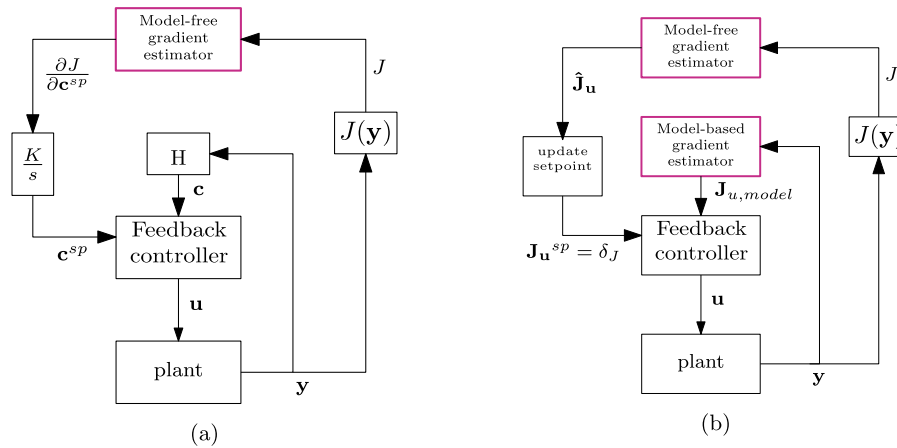


Fig. 9. Combination of model-free gradient estimation method with (a) self-optimizing control i.e. models used offline to determine \mathbf{H} (b) model-based gradient estimation, i.e. models used online.

$$\theta = J_{\mathbf{u}}$$

More recently, van Keulen et al. (2020) proposed to apply a multisine dither signal to identify the local linear dynamic model of the plant, where the perturbation frequency is in the same timescale as the plant dynamics. The transfer function of the local linear dynamic system at the multisine dither frequencies are estimated using Fourier Transform of the past N samples of input and cost measurement (much like the FFT-based gradient estimation described in Section 4.5). The local steady-state gradient is then estimated by using an online complex curve fitting, by taking a frequency domain approach to system identification.

Another approach to account for the plant dynamics is to compensate for the phase shift introduced by the plant dynamics, such that the static map assumption can be relaxed. For example, one can estimate the different harmonics using a Kalman filter, and drive the high frequency harmonics to zero when the input perturbations are in the same timescale as the plant dynamics (Atta et al., 2015; Trollberg and Jacobsen, 2016). In other words, the gradient estimation problem is replaced by a problem of estimating the harmonics, and this is used to drive the process to its steady-state optimum.

5. Combination of model-free and model-based approaches

It can be seen that the model-based and model-free methods for the unconstrained optimization problem have their own advantages and disadvantages. In short, one of main strengths of the model-based methods is that it converges faster, whereas the main weakness is the dependence on a model (making it susceptible to plant-model mismatch). On the other hand, one of the main strengths of the model-free methods is that it circumvents the need for developing models (hence is not susceptible to plant-model mismatch), whereas the main weakness is the prohibitively slow convergence to the optimum. Note that the model-based and model-free methods are complementary and not contradictory. In general the model-free methods work in the slow time scale, whereas model based methods work in the fast time scale. This time scale separation can be exploited to combine the model-based and model-free methods in a hierarchical fashion. For example, any model-based method may be used on the fast timescale, and a slow varying model-free method can be used on the on top to account for any plant-model mismatch.

5.1. Setpoint correction using model-free gradient estimation

When using the models offline to determine the self-optimizing CV (cf. Section 3.1), this is based on some nominal conditions. As mentioned earlier, as the operation drifts far away from the nominal operating condition, this leads to steady-state losses. Model-free gradient estimation approaches can be used to adjust the setpoint of the self-optimizing CVs in order to account for the steady-state losses, as shown in Fig. 9a. Such synergistic combinations of model-free methods and self-optimizing control were shown to provide an improved performance, compared to each of the method used individually. For example, Jäschke and Skogestad (2011) proposed to combine NCO-tracking with self-optimizing control and the improved performance was demonstrated using a CSTR example from Economou et al. (1986). Similarly, Straus et al. (2019) proposed to combine the least squares-based extremum seeking control with self-optimizing control, and demonstrated the performance improvement on a 3-bed ammonia reactor case example. In both the approaches, the setpoint of the self-optimizing variable \mathbf{c}^{sp} was updated by the model-free method in a cascaded fashion as shown in Fig. 9a.

Modifier adaptation is another RTO scheme that was proposed by Marchetti et al. (2009) to address the plant-model mismatch. Here, the main idea is to estimate the plant gradients directly from the measurements, and use this to correct the optimization problem. For the unconstrained optimization problem (5), the modifier adaptation scheme iteratively solves

$$\min_{\mathbf{u}} J_{model}(\mathbf{u}, \mathbf{d}) + \epsilon_j^k + \delta_j^k(\mathbf{u} - \mathbf{u}^k) \quad (52)$$

where

$$\epsilon_j = J - J_{model}$$

$$\delta_j = \hat{\mathbf{J}}_{\mathbf{u}} - \hat{\mathbf{J}}_{\mathbf{u},model}$$

are the so-called modifier terms.² Here, the plant gradient $\hat{\mathbf{J}}_{\mathbf{u}}$ is estimated using any model free gradient estimation techniques reviewed in Section 4.

Remark 5 (Modifier Adaptation). Although MA uses model-free gradient estimation algorithms, it is not a feedback optimizing control approach, since it is based on solving the numerical optimization problem (52) in real-time. To this end, Modifier Adaptation

² Note that MA is typically formulated for the full optimization problem (1), which includes additional modifier terms for the constraints as well. However for the sake of simplicity, we only consider the reduced unconstrained optimization problem (5).

can be seen as a combination of the model-free gradient estimation approach with the traditional steady-state optimization (SRTO) shown in Fig. 2.

A similar approach to modifier adaptation, but one that does not require explicitly solving (52) would be to use the modifier term $\delta_j = \hat{\mathbf{J}}_{\mathbf{u}} - \hat{\mathbf{J}}_{\mathbf{u},model}$ as the setpoint to the gradient controller as shown in Fig. 9b. By doing so, we add a bias correction to the model-based gradient estimate to account for the plant-model mismatch in the slow time scale.

5.2. Discrepancy modeling framework

Discrepancy modeling framework is an alternative approach to updating the setpoints using a model-free gradient estimation method in a cascade fashion. Here the main idea is to approximate the “unmodeled” effects using direct cost measurements. That is, the residual between the cost measured from the plant J and the cost predicted by the model J_{model} is used to estimate the cost gradient $\hat{\mathbf{J}}_{\mathbf{u}}$. For example, one can train a function approximator $\phi_\epsilon(\mathbf{u})$ that predicts the zeroth order bias correction term ϵ_j , and use the derivative of the trained function $\phi_\epsilon(\mathbf{u})$ to account for plant-model mismatch.

$$\epsilon_j = J - J_{model} \approx \phi_\epsilon(\mathbf{u}) \quad (53)$$

where $\phi_\epsilon(\mathbf{u})$ can either be parametric function approximators (e.g. neural networks), or non-parametric function approximators (e.g. Gaussian processes). The plant gradient is then estimated as the sum of the model gradient $\hat{\mathbf{J}}_{\mathbf{u},model}$ and the gradient of the function approximator.

$$\hat{\mathbf{J}}_{\mathbf{u}} = \hat{\mathbf{J}}_{\mathbf{u},model} + \frac{\partial \phi_\epsilon}{\partial \mathbf{u}} \quad (54)$$

Such an approach for estimating the plant gradient was recently used in the context of modifier adaptation by Gao et al. (2016), Matias and Jäschke (2019), del Rio Chanona et al. (2019). Although these methods were applied in modifier adaptation, the estimated gradient (54) can also be directly driven to a constant setpoint of zero using feedback control, which to the best of our knowledge, has not been shown earlier.

6. Operation with changing active constraints

The set of active constraints change with changing operating conditions. Feedback optimizing control must be able to automatically handle these changes and ensure that the process is operated optimally at steady-state over different operating conditions. Section 6.1 details how this is handled in the case of region-based control (paradigm 1), and Section 6.2 details how this is handled in the case of primal-dual feedback control (paradigm 2).

6.1. Region-based control

As mentioned earlier in Section 2.1, paradigm 1 first involves identifying the potential active constraint regions. Control structures are then designed in each active constraint region by partitioning the available degrees of freedom into constrained and unconstrained degrees of freedom (cf. Theorem 1). The set of active constraints change with changing operating conditions, which requires selecting different set of controlled variables and reconfiguration of the control loops.

The identification of the current active constraint region, may be done by tracking the constraints and self-optimizing variables in all neighboring regions, and we switch region if any of these become active (usually identified by a sign change (Manum and Skogestad, 2012)). Note that this means that the switching is based on the controlled variables (CVs). The controllers in each region

may in principle be completely different and may be multivariable (e.g. MPC).

However, a simpler approach in terms of implementation is to pair each constraint with one particular input. This is the approach studied in this paper. In this case, we implement a controller for each CV (constraints and self-optimizing variables) and the switching is done based on the controller output $\hat{\mathbf{u}}$, that is, on the manipulated variables (MVs). The advantage is that one may use the same controllers in many regions, and the switching may be performed using standard max/min-selectors. The disadvantage is that we need to pair each potential constraint with one specific input. This may not be possible in some cases with many constraint combinations, because it requires that the constraints paired with a given input are never optimally active at the same time.

6.1.1. CV-MV pairing

To this end, this requires choosing CV-MV pairing in each active constraint region. There is no systematic approach that provides a unique CV-MV pairing. However, this can generally be done in practice by using some rule of thumb.

- Pair-close rule - In order to avoid large time delays and sluggish control, it would also be wise to control a CV using an MV that is physically close to one another (Skogestad and Postlethwaite, 2007).
- Non-negative relative gain array (RGA) - The pairing must be chosen such that the steady-state RGA of the resulting transfer matrix is non-negative and close to identity matrix at crossover frequencies (Skogestad and Postlethwaite, 2007).
- Pair on large gain - One must also try to avoid pairing important CVs with MVs that quickly saturate, and instead pair such MVs with less important CVs that may be given up (Skogestad, 2004).

Additionally, the constraint grouping when using selectors to switch between the different CVs (cf. step S1 in Section 6.1.3), also influence the MV-CV pairing. Note that there may be several different possible MV-CV pairings to achieve the same objectives, and the pairing rules listed above can guide in selecting a good control structure design that would help reduce the number of logic blocks required to reconfigure the control loops.

Once the different control loops are designed for each active constraint region, we then have to design a switching strategy between the different active constraint regions. Switching between active constraint regions can be achieved by using classical advanced control elements, such as selectors, split-range control, valve position control etc. as detailed below.

6.1.2. MV-MV switching

MV-MV switching can be achieved using three alternative classical control elements, namely,

1. Split-range control (Reyes-Lúa et al., 2018; Reyes-Lúa and Skogestad, 2019b)
2. Valve position control (Reyes-Lúa and Skogestad, 2019b)
3. Individual controllers for each MV with different setpoints (Reyes-Lúa and Skogestad, 2019a)

Split-range control Split range control (SRC), dates back to 1940's (Eckman, 1945) and is very commonly used in process control industries. Here, there is a controller that produces a control signal v , typically between 0 - 100%, that is input to the split-range block, which then translates the control signal v to the physical manipulated variables u_i . A typical split range block with two MVs is shown in Fig. 10a, where u_1 is used to control the CV and u_2 is saturated when the control signal is below the split value v^* , whereas u_2 is used to control the CV and u_1 is saturated when the control signal is above the split value v^* . The reader is referred to

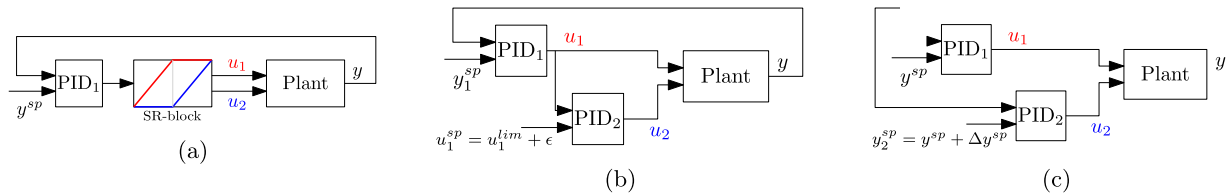


Fig. 10. MV-MV switching using (a) split range control (b) Valve position control (c) controllers with different setpoints.

Reyes-Lúa et al. (2018), Reyes-Lúa and Skogestad (2019b) for more detailed information on using split-range control for MV-MV switching.

Valve position control Valve position control (VPC), also known as input-resetting control, is often used in industrial practice to improve the dynamic performance, by allowing one MV to take care of the fast response, and another of the long-term control. However, here it is used to extend the steady-state range when implemented as shown in Fig. 10b. Here, u_1 is used to control the CV, while u_2 is not used, i.e. it is kept at its desired limiting value, e.g. $u_2^{lim} = 0$. However, when u_1 is approaching its limit, u_2 controls u_1 to a setpoint of $u_1^{lim} + \epsilon$ in order to prevent u_1 from saturating (Shinsky, 1996; Reyes-Lúa and Skogestad, 2019a).

Controllers with different setpoints Another alternative is to use independent controller for each input, and the setpoint for the different controllers vary by a small amount, as shown in Fig. 10c. For example, u_1 is used to control the CV y at its optimal setpoint y^{sp} , and u_2 is used to control the CV y at a setpoint of $y^{sp} + \Delta y^{sp}$, where Δy^{sp} is large enough such that only one controller is active at any given time. It is important to note that this approach leads to some steady-state loss, when u_2 is used to control the CV at $y^{sp} + \Delta y^{sp}$. However, it may be possible to reduce the loss of having different setpoints. The simplest is to have a master controller which slowly resets y^{sp} so that $y^{sp} + \Delta y^{sp}$ returns the truly desired setpoint. An alternative approach with individual controllers that completely avoids different setpoints, is the somewhat different “Baton strategy” (Reyes-Lúa and Skogestad, 2020), where each controller needs to identify when it has saturated. The reader is referred to Reyes-Lúa et al. (2018), Reyes-Lúa and Skogestad (2019b, 2020) for more detailed information on using controllers with different setpoints to extend the steady-state operating range.

6.1.3. CV-CV switching

A CV constraint that is optimally active, may no longer be active when a disturbance changes. In this case, a different variable (another CV constraint, or a self-optimizing CV) must be controlled in order to operate the process optimally for the new disturbance. To switch between the CVs, one can use individual controllers for each CV. The MV value that is implemented on the plant is then selected among the controller outputs using one or more selector blocks (Krishnamoorthy and Skogestad, 2019; Reyes-Lúa and Skogestad, 2019b).

To design selectors, we pair each MV with a set of constraints *a priori*, such that for a given MV u , the potential controlled variables are:

- N CV inequality constraints, denoted by y_i for all $i = 1, \dots, N$
- at most one CV y_0 with a setpoint that may be given up (i.e. self-optimizing CV)
- MV inequality constraints

A systematic design procedure for designing selector blocks was recently proposed by Krishnamoorthy and Skogestad (2020b). The main steps of the systematic design procedure of selectors for each MV u with different potential CVs can be summarized as follows: **S1 (constraint grouping)** Group the candidate CV constraints into two sets, \mathbb{Y}^+ and \mathbb{Y}^-

- The set \mathbb{Y}^+ consists of all constraints that are satisfied with a large input u . This includes a max-constraint on u (u_{max})
- The set \mathbb{Y}^- consists of all constraints that are satisfied with a small input u . This includes a min-constraint on u (u_{min})

S2 (SISO control loops) Design individual SISO control loops to compute the input for each CV constraint (u_i) and for the CV setpoint controller (u_0). Note that in this case it is important to implement anti-wind up for all the controllers, such that the controller that is not selected, does not keep integrating.

S3a (Choice of selector)

- Use a min-selector on u_i for constraints that are satisfied with a large input, $\bar{u} = \min_{\mathbb{Y}^+} u_i$
- Use a max-selector on u_i for constraints that are satisfied with a small input, $\underline{u} = \max_{\mathbb{Y}^-} u_i$

S3b (Feasibility) For the set of constraints to be consistent, that is, to have feasible operation, we need $\bar{u} > \underline{u}$ (Krishnamoorthy and Skogestad, 2020b), where \bar{u} is the output from the min-selector and \underline{u} is the output from the max-selector.

S4 (Optimality) When the problem is feasible, the optimal input is given by $u = \text{mid}(\underline{u}, u_0, \bar{u})$, where u_0 is the control input computed by the controller that controls the self-optimizing CV, e.g. steady-state cost gradient. This can be achieved using a mid-selector block in addition to the min- and max-selector blocks in step S3a to compute \bar{u} and \underline{u} , respectively as shown in Fig. 11a. Alternatively, this can also be achieved using a min-max or max-min selector block in series as shown in Fig. 11b and c, respectively. The three structures shown in Fig. 11 are equivalent when the constraints are feasible (cf. step S3b). If we only have one constraint set \mathbb{Y}^+ or \mathbb{Y}^- , then we only need a single selector block, namely, a min- or max-selector block, respectively.

S5 (Constraint Priority) If the constraints are conflicting, that is $\bar{u} < \underline{u}$, then the three structures shown in Fig. 11 are not equivalent. In this case, the constraint priority can be used to decide the appropriate selector block. If the constraints in \mathbb{Y}^+ take a higher priority than the constraint in \mathbb{Y}^- , then we can use a max-min structure (Fig. 11c). If not, we can use a min-max structure (Fig. 11b). If necessary, one can find another MV v to control the constraint given up by the original MV u . This would involve MV-MV switching as described above in Section 6.1.2.

6.1.4. Summary of switching strategies

To this end, the step-by-step procedure to design suitable switching strategies under paradigm 1 can be summarized as:

- **Step 1** - Identify the set of active constraints (CV with limits), self-optimizing CVs (CV with setpoints) and the manipulated variables (MV).
- **Step 2** - Organize the constraints in a priority list. The priority list may be used to decide which constraints may be given up, in situations where it may not be possible to control all the active constraints (e.g. when the no. of active constraints is more than the number of degrees of freedom, or if the constraints are conflicting as described in step S5 above).
- **Step 3** - Identify relevant active constraint regions.

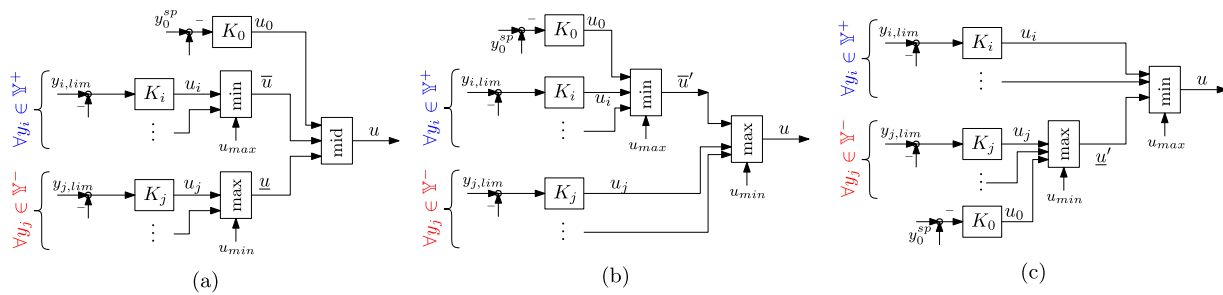


Fig. 11. CV-CV switching using (a) min-max-mid selector blocks (b) min-max selector blocks (c) max-min selector blocks.

- **Step 4** - Design the control structure for nominal operating conditions. Typically this would be the most common or important operating region.
- **Step 5** - Design the control structure of the identified active constraint regions from step 3, and design MV-MV and/or CV-CV switching strategies between the different active constraint regions, as described above. Note that it may not always be possible to find simple switching schemes based on min- and max-selectors to achieve this. In such cases, some alternative logic scheme may be needed in some constraint regions.

Recent works such as Reyes-Lúa and Skogestad (2019b), Reyes-Lúa and Skogestad (2019a), Reyes-Lúa et al. (2018), Krishnamoorthy and Skogestad (2019, 2020b) provides a detailed discussion on switching between active constraint regions along with a wide range of application examples in the context of feedback optimizing control. It is important to note that explicitly designing the switching strategies requires one to identify all the different possible active constraint switching *a priori* and carefully design the switching logic to account for all the possible switching. As such, this is more suited for small-scale processes with only a few active constraint regions that can be easily managed using such switching logics.

6.2. Primal-dual feedback optimizing control

In the case of primal-dual feedback controllers as described in Section 2.2, the CV-MV pairing is rather straightforward, where for the dual controllers, it is natural to pair the Lagrange multiplier with its corresponding constraint control, and for the primal controllers, it is natural to pair diagonally (i.e. pair u_i with $\nabla_{u_i} \mathcal{L}$) as long as the determinant of the Hessian $\det(\nabla_{\mathbf{u}\mathbf{u}}^2 \mathcal{L})$ does not change sign. More precisely, here, we pair the physical manipulated MVs \mathbf{u} to drive the controlled variables $\mathbf{c} = \nabla_{\mathbf{u}} \mathcal{L}(\lambda, \mathbf{u}, \mathbf{d})$ to zero, and the Lagrange multipliers (i.e. the dual variables) must be paired to their corresponding constraints. By doing so, when an active constraint becomes inactive as the disturbance changes, the max-operator in (14) would make the corresponding Lagrange multiplier $\lambda = 0$ and hence changes in active constraints are handled automatically without the need for additional switching strategies.

7. Illustrative example - Williams-Otto reactor

In this section, we use a benchmark Williams-Otto reactor example (Williams and Otto, 1960) to provide an illustrative comparison of the different model-based and model-free feedback optimizing control approaches discussed above. The Williams-Otto reactor converts the raw materials A and B to useful products P and E, along with by-products C and G, through a series of reactions

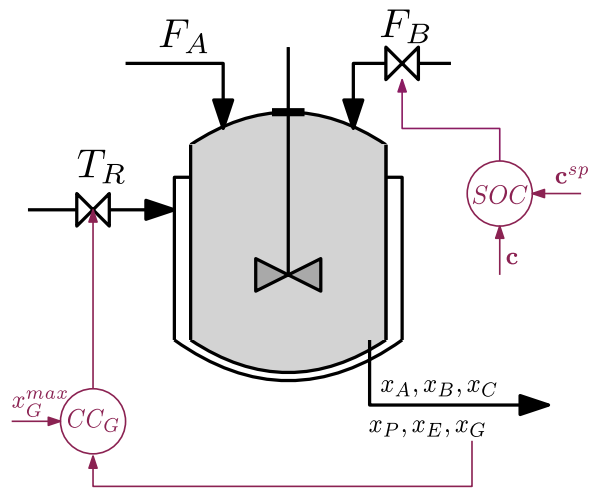
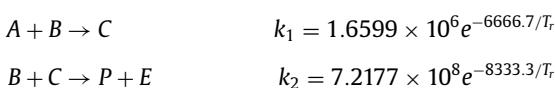


Fig. 12. Proposed control structure used in Sections 7.1 and 7.2 for achieving optimal operation of a Williams-Otto reactor using feedback control.

$$C + P \rightarrow G \quad k_3 = 2.6745 \times 10^{12} e^{-11111/T_r}$$

The process is controlled using the feed stream F_B with pure B component and the reactor temperature T_r , i.e. the process has two physical manipulated variables. The feed stream F_A with pure A component is a disturbance to the process. The objective is to maximize the profits from the valuable products P and E, subject to purity constraints on G and A in the product stream. the steady-state optimization problem is formulated as,

$$\begin{aligned}
 \min_{T_r, F_B} & -1043.38x_P(F_A + F_B) - 20.92x_E(F_A + F_B) \\
 & + 79.23F_A + 118.34F_B \\
 \text{s.t. } & x_G \leq 0.08, \\
 & x_A \leq 0.12
 \end{aligned}$$

and the objective is to asymptotically drive the system to its optimal operating point using feedback control.

We will first only consider the case where the constraint $x_G = 0.08$ is the only active constraint, and is controlled using the reactor temperature T_r as shown in Fig. 12. We then have one unconstrained degree of freedom, namely feed stream F_B , for which we compare the performance of different model-based and model-free approaches detailed in Sections 3 and 4. This is schematically represented in Fig. 12, where the self-optimizing CV \mathbf{c} is given by the different methods. The simulation results shown in this section were performed using MATLAB v2020b³.

³ The MATLAB codes that were used to generate the plots in this section can be found in <https://github.com/dinesh-krishnamoorthy/Feedback-based-RTO/tree/master/OverviewFeedbackRTO>

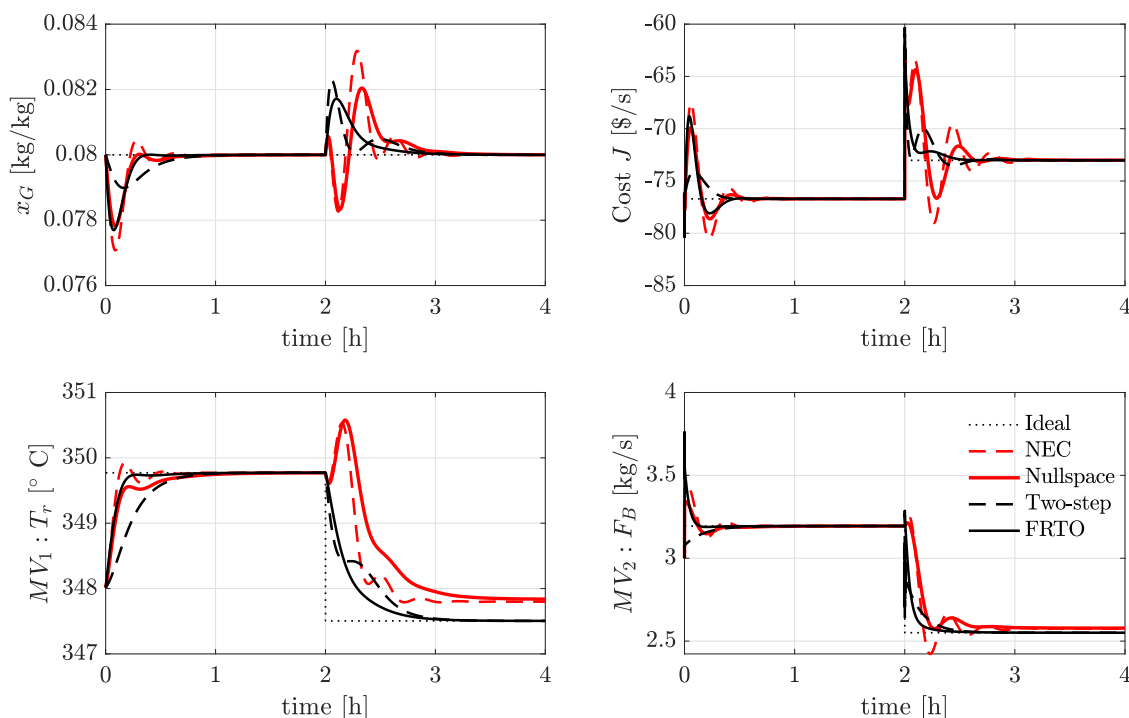


Fig. 13. Comparison of four different model-based feedback optimizing control approaches on the benchmark Williams-Otto reactor example.

Remark 6. Note that the focus of the simulation results throughout this paper is the asymptotic behavior of the different approaches, and not the transient behavior. As such, one may find alternative tuning parameters for the controllers that provides a different transient behavior.

7.1. Model-based approaches

In this section, we consider the case with no plant-model mismatch. The model used in this section is based on the three reaction model shown in Appendix B.1. For the unconstrained degree of freedom, F_B , we compare the use of two offline methods namely, nullspace method (Section 3.1.2) and neighboring extremal control (Section 3.1.3), and two online methods; two-step approach (Section 3.2.1) and Feedback RTO (Section 3.2.2).

The nullspace method and neighboring extremal control were designed around the nominal operating conditions of $F_A = 1.3$ kg/s. Fig. 13 shows the simulation results using the different model-based approaches. The true steady-state optimum is shown in black dotted lines.

The simulation starts at the nominal condition of $F_A = 1.3$ kg/s, and one can see that all the four methods asymptotically converge to the optimum. At time $t = 2$ h, the disturbance changes to $F_A = 1$ kg/s. As the process is driven away from the nominal operating conditions, self-optimizing control and neighboring extremal control leads to some steady-state offset from the true optimum, since these are based on local linear approximation around the nominal operating conditions. The online model-based methods, on the other hand, are able to asymptotically drive the process to its optimum, since these are based on local linear approximation around the current operating point. In general, it can be seen that the model-based approaches converge to the optimum in the same time scale as the plant dynamics.

7.2. Model-free approaches

In this section, we now compare the model-free approaches for the unconstrained optimum. Fig. 14 shows the simulation re-

sults where the steady-state cost gradient is estimated using four different approaches, namely, the classical extremum seeking control (Section 4.2), least squares estimation approach (Section 4.3), Fast Fourier transform (Section 4.5) and the Kalman filter approach (Section 4.4). It can be clearly seen that all the four methods are able to asymptotically drive the process to its optimum without the need for rigorous nonlinear process models. However, the convergence to the optimum is significantly slower compared to the model-based approaches shown in Fig. 13.

7.3. Combination of model-based and model-free approaches

We now consider the case with plant-model structural mismatch by considering a model with only two reactions and five components, as shown in Appendix B.2, whereas the plant is given by the three reaction system as shown in Appendix B.1.

We first use the model-based gradient estimation method (from Section 3.2.2), where the gradient is estimated using the structurally wrong model without any additional correction. The simulation results are shown in Fig. 15 (red solid), where it can be clearly seen that the true plant optimum (shown in black dotted) is not reached due to the model structural mismatch.

We now consider two approaches to account for the steady-state error induced by the structural mismatch. The first approach is based on the setpoint correction using the modifier term δ_j which is computed based on the plant gradient and the model gradient (cf. Section 5.1). The simulation results using the modified setpoint is shown in Fig. 15 (black solid), where it can be seen that by modifying the setpoint using the estimated plant gradient, we are able to asymptotically drive the system to its true optimum (shown in black dotted) despite using a structurally wrong model.

The second approach that is used here is based on the discrepancy modeling framework (cf. Section 5.2). Here we use a Radial Basis Function network (RBFN) as the function approximator $\phi_\epsilon(\mathbf{u})$ to account for the discrepancy between the model predicted cost and the measured cost $\epsilon_j = J - J_{model}$, as done by Matias and Jäschke (2019). The estimated gradient is given by (54) as described in Section 5.2. The simulation results using this approach

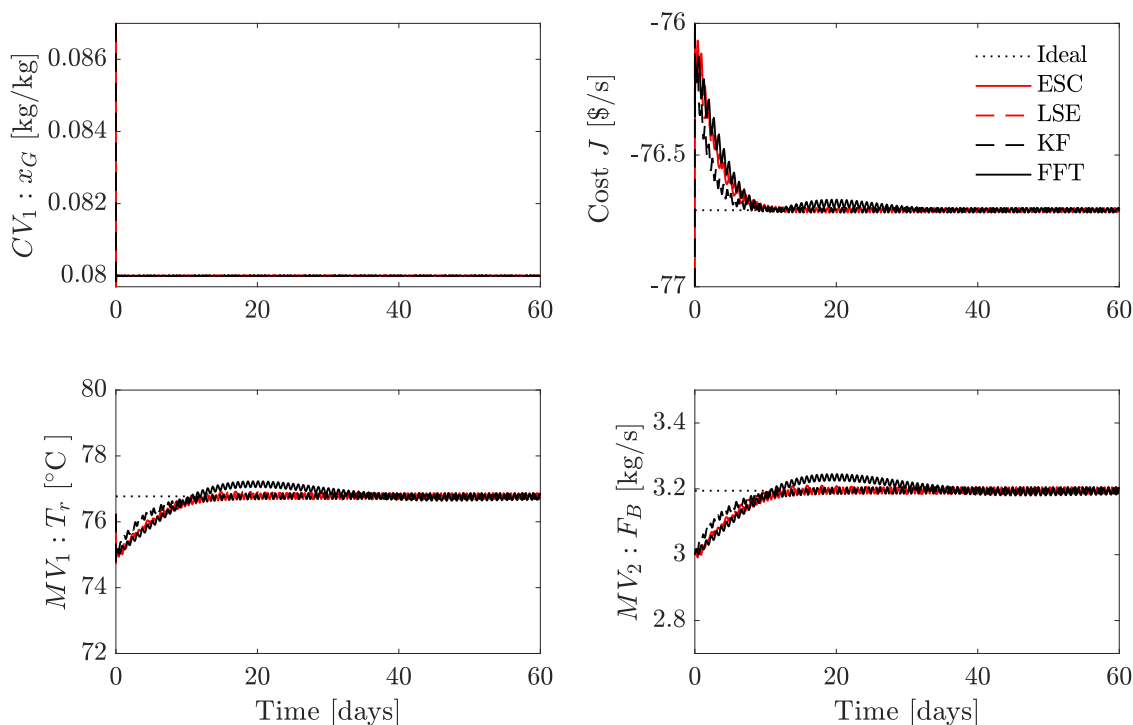


Fig. 14. Comparison of four different model-free feedback optimizing control approaches on the benchmark Williams-Otto reactor example.

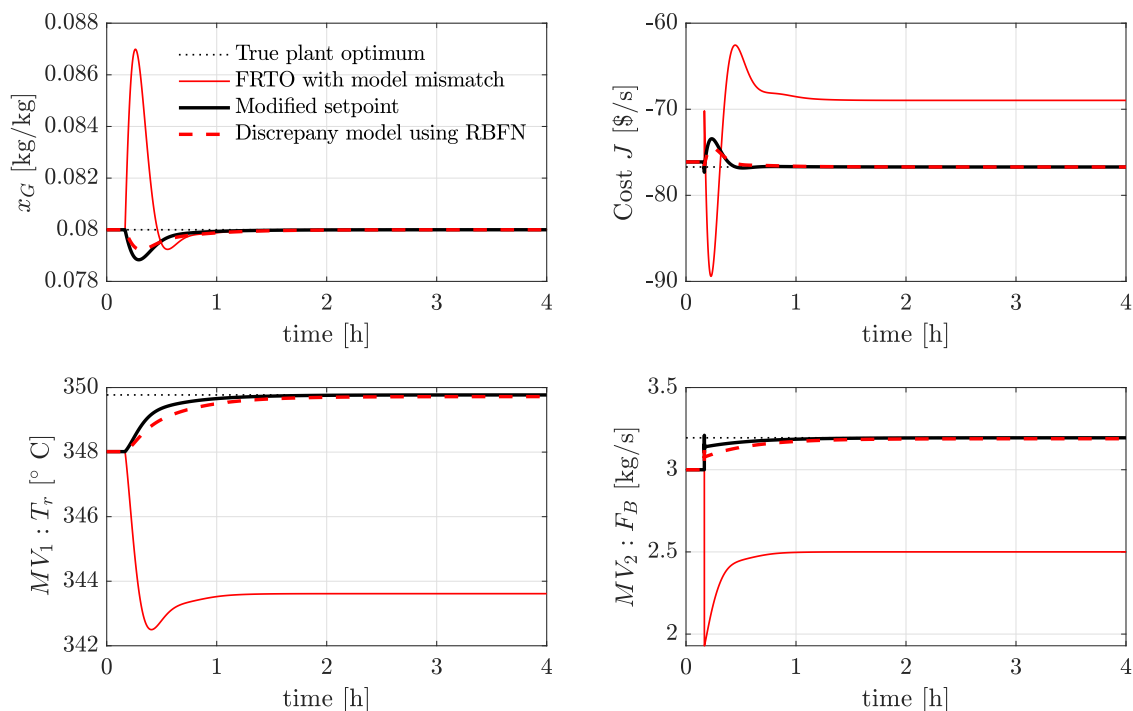


Fig. 15. Simulation results showing the effect of plant-model mismatch, and comparison of two approaches that combines model-based and model-free methods to handle structural mismatch.

is shown in Fig. 15 (red dashed), where it can be seen that the plant is asymptotically driven to its true optimum (shown in black dotted) despite using a structurally wrong model.

7.4. Handling changes in active constraint sets

So far, we considered a case where the set of active constraints remained the same. We now consider a case, where

the disturbances varies as shown in the bottom right subplot in Fig. 17 which causes the set of active constraints to change. The two paradigms (cf. Sections 2.1 and 2.2) to handle the changes in the active constraint sets are compared in this section.

7.4.1. Region-based control

In this approach, we have to predetermine the different active constraint regions and design control structures for each active

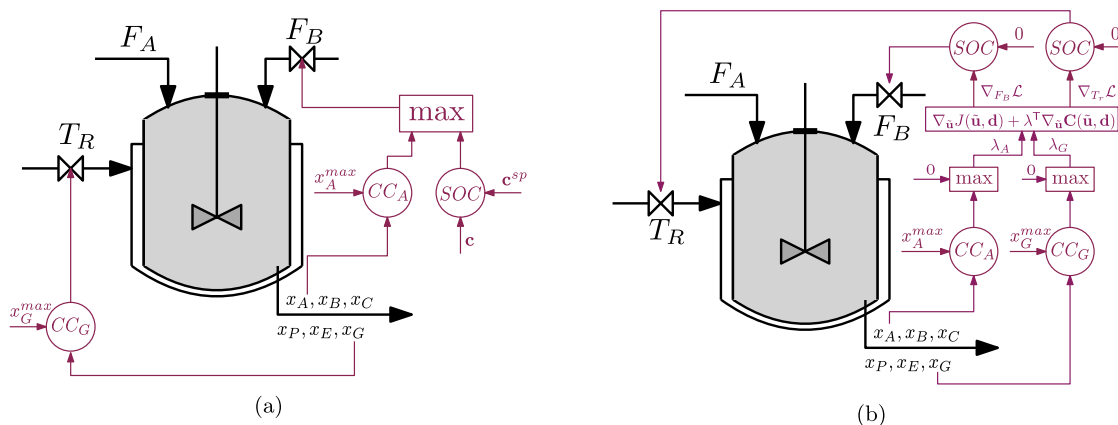


Fig. 16. Control structure design (a) Region-based control with selectors to switch between the active constraint regions (cf. paradigm 1 in Fig. 3a) (b) Primal-dual feedback control approach (cf. paradigm 2 in Fig. 3b).

constraint region and switch between the different regions using classical control elements as detailed in Section 6.1.

Here we show how to identify the different active constraint regions without using process models. Since there are two constraints, we can at most have $2^2 = 4$ possible active constraint regions, namely,

1. x_G and x_A active
2. only x_G active
3. only x_A active
4. no active constraints

However, the purity constraint on x_G is very low such that the one can expect the constraint on x_G to be always active. Therefore, we can eliminate regions 3 and 4, and design control structures for regions 1 and 2 only, and design a suitable switching strategy between these two regions. It is important to note that eliminating active constraint regions 3 and 4 without solving the optimization problem offline requires good process understanding and insight.

In region 1, optimal operation occurs when the purity constraints on x_G and x_A are active. Therefore, this region corresponds to active constraint control. One possible control structure design here is to control x_G at its limit of 0.08 kg/kg using the reactor temperature T_R and control x_A at its limit of 0.12 kg/kg using the feed rate F_B .

In region 2, optimal operation occurs when the purity constraints on x_G is active, which is controlled using the reactor temperature, and we have one unconstrained degree of freedom, namely F_B , which should be used to control a self-optimizing variable. Here we use the steady-state cost gradient computed using the model (assuming no structural mismatch).

Automatic switching between the two regions can be achieved by using a max selector. Fig. 16a shows the proposed control structure design using this approach. The simulation results are shown in Fig. 17 (red dashed), where it can be seen that as the disturbance changes from $F_{a1} = 1.4$ kg/s to $F_{a2} = 1.9$ kg/s at time $t = 3$ h, the max constraint on x_A becomes active, and the max-selector switches from the gradient controller to the x_A composition controller, hence achieving asymptotic optimal operation, as the operating condition changes. Similarly, at time $t = 6$ h, the disturbance reduces to $F_{a3} = 1.5$ kg/s, and the selector switches from the x_A composition controller to the gradient controller.

7.4.2. Primal-dual feedback optimizing control

Alternatively, one can use the primal-dual feedback controllers as detailed in Section 2.2, where we consider the dual variables λ_{x_A} and λ_{x_G} as additional degrees of freedom which are used to control the constraints to its limit, while ensuring that these are non-

negative using a max-selector. The primal MVs F_B and T_R are used to control the gradient of the Lagrangian function to a constant setpoint of zero. The proposed control structure using the primal and dual variables is shown in Fig. 16b. Here, the cost and constraint gradients are estimated using a model (assuming no structural mismatch). The simulation results are shown in Fig. 17 (black solid), where it can be seen that as the disturbance changes from $F_{a1} = 1.4$ kg/s to $F_{a2} = 1.9$ kg/s at time $t = 3$ h, and again at time $t = 6$ h to $F_{a3} = 1.5$ kg/s, the proposed control structure is able to achieve asymptotic optimal operation without the need to explicitly design additional switching strategies. However, this approach is less robust to faulty control loops since the constraints are controlled indirectly by updating the Lagrange multipliers. For example, we simulated a case where F_B saturates at 4 kg/s for the same disturbance profile. In the primal-dual approach, the constraints on x_G as well as x_A were violated, since the constraints are controlled indirectly by updating λ . Whereas in region-based approach, the constraint on x_G was still feasible, since this was directly controlled by T_R and is therefore not affected when F_B saturates. This is as expected and the simulation results are not shown here for the sake of brevity.

8. Discussion

In this section, we discuss the main distinguishing properties of the different feedback optimizing control approaches.

8.1. Rigorous process models

Model-free methods, as the name suggests, does not need rigorous nonlinear models, circumventing the modeling related challenges. Rigorous nonlinear process models are required only by the model-based methods. The offline model-based approaches reviewed in Section 3.1 requires steady-state process models to choose the best measurement combination as self-optimizing controlled variables. Models are used online to estimate the cost gradient, which are then controlled to a constant setpoint of zero. If the steady-state models are used online, then the models can be updated only using the steady-state process measurements. This leads to steady-state wait time issues (Darby et al., 2011). Therefore, in order to use transient measurements, it is recommended to use dynamic models for online model-based methods (Krishnamoorthy et al., 2019c).

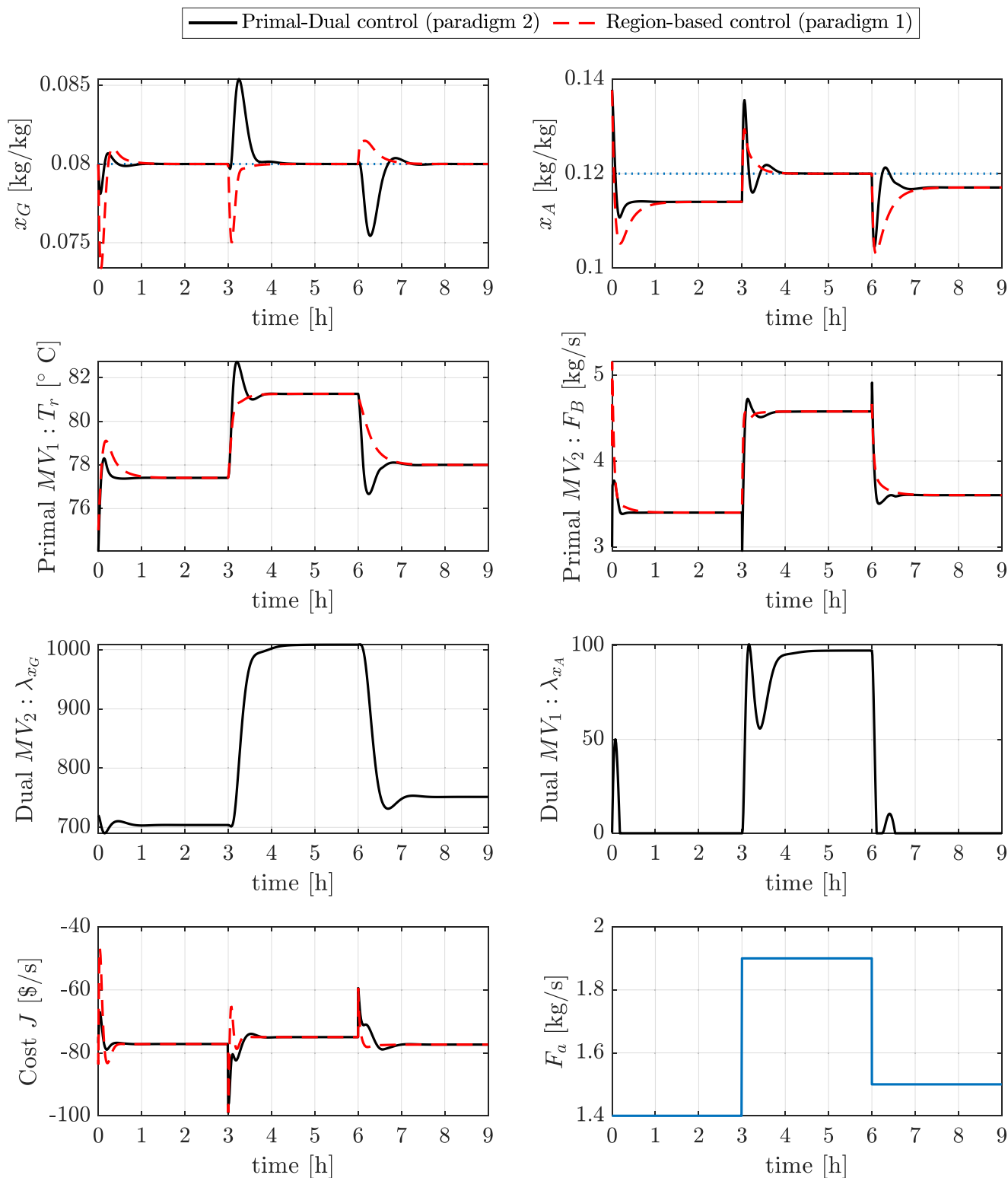


Fig. 17. Comparison of region-based control with selectors (cf. paradigm 1 in Fig. 3a) and the primal-dual feedback control approach (cf. paradigm 2 in Fig. 3b) on the benchmark Williams-Otto reactor example.

8.2. Measurement requirements

Since optimal operation is achieved using feedback control, the choice of measurements that are available affects the applicability of the different methods.

For active constraint control, direct measurements of the constraints are required. It is reasonable to assume that the constraints

are typically measured in most applications, since these are required for monitoring the process operations anyway.

For the unconstrained optimum, models can be used offline to select a linear combination of a subset of the available measurements (cf. Section 3.1). Online model-based gradient estimation methods, that relies on the use of state and parameter estimators, requires sufficient measurements such that the states and the pa-

parameters being estimated are observable. Since the model maps the measurements to the cost, direct cost measurements are not required for model-based methods (cf. Section 3.2).

On the other hand, the model-free methods require the cost to be directly measured. However, note that in many chemical processes, direct measurement of the cost is often not available, especially, if the cost function is comprised of several terms. A typical cost function of a chemical process has the form

$$J = p_Q Q + p_F F - p_1 P_1 - p_2 P_2 + \dots \quad (55)$$

where Q , F , P_1 and P_2 are the flow rates (in [kg/s]) of utility, feed, products 1 and 2 respectively, and p_Q , p_F , p_1 and p_2 are the corresponding prices (in [\$/kg]). This means that accurate flow measurements of all the components comprising of the cost function is required, in order to employ model-free methods.

8.3. Accuracy

The offline model-based methods are based on local linearization around some nominal optimal point. Consequently, controlling a linear measurement combination leads to steady-state losses if the operation drifts far away from the nominal optimal point, even if there is no plant-model mismatch.

Furthermore, in the presence of plant-model mismatch, the estimated gradient differs from the actual plant gradient, leading to suboptimal operation. In the presence of structural mismatch, parameter estimators (such as the one used in the two-step method and the feedback RTO method) are not sufficient to close the optimality gap, as clearly demonstrated by Roberts and Williams (1981), Chachuat et al. (2009) to name a few. This was also demonstrated in Fig. 15. Consequently, in online model-based methods, where the cost gradient is estimated using the model, plant-model mismatch implies that the plant may not be driven to its true optimum.

Since model-free methods estimate the plant-gradient directly from the cost measurement, the estimated gradient in theory reflects the true plant gradient under the assumption that the gradient estimation algorithm is properly implemented. If not, this can lead to estimation errors, leading to suboptimal operation. For example, poor choice of the tuning parameters in extremum seeking control has been shown to lead to inaccurate gradient estimation (Krstić, 2000). This important caveat is often overlooked, and there is a clear need to better understand the sensitivity of the tuning parameters on the gradient estimation accuracy in many model-free gradient estimation methods.

Furthermore, as mentioned earlier, a typical cost function in a chemical process plant may comprise of several flowrate terms. Often, the operational profit is made by shifting smaller amounts of feed to the most valuable product, and very accurate flow measurements are required in order to capture this. In practice, data reconciliation using nonlinear process models may then be needed to get accurate flow estimates. In such cases, it is important to keep in mind that the model-free methods will not truly be “model-free” and may suffer from structural mismatch, since a nonlinear process model is needed to get a measurement/estimate of the cost.

8.4. Disturbance rejection

Model-based methods can handle unmeasured, but expected disturbances, i.e. the model captures the effect of the anticipated disturbances. Whereas, model-free methods can handle unexpected and unmeasured disturbances. However, it has been shown that changes in any disturbance may affect the accuracy of the gradient estimation in model-free methods. This is because, the effect of the disturbances on the cost measurement is often not

explicitly accounted for in the model-free gradient estimation routines. That is, the model-free methods typically assume that any change observed in the cost measurement is induced by the input. For example, abrupt changes in disturbances may temporarily affect the gradient estimate, before converging to the true gradient.

Disturbance measurements, if available, may be used to improve the gradient estimation, by explicitly accounting for the change in the cost measurements caused by the disturbances, (Krishnamoorthy et al., 2016). Alternatively, one can also use an event-based supervisory control to halt the gradient estimation temporarily following abrupt/abnormal changes in the cost measurement, similar to the steady-state detection used in traditional RTO. This approach has been used in the context of extremum seeking control by Marinkov et al. (2014). It is also important to note that disturbances occurring in the same frequency as the input perturbation leads to inaccurate gradient estimation since this makes it impossible to differentiate the effect of the disturbance and the input on the observed cost measurement.

8.5. Convergence time

The process knowledge available in the form of rigorous process models enables model-based methods to converge significantly faster than their model-free counterparts. In model-based methods, the convergence time is predominantly determined by the PID controller tuning.

Model-free methods presented in Sections 4.1–4.6 assume that the dynamic plant behaves like a static map. The time scale separation required for this assumption to be valid, makes the convergence of model-free methods to be very slow. In the chemical process industry, the settling times are typically in the range of minutes to hours, which often leads to prohibitively slow convergence, typically in the range of hours to even days. This is one of the main reasons why model-free methods are not used widely in process industries, despite the popularity of such methods in other fields. Addressing this challenge can broaden the applicability of model-free gradient estimation approaches in the chemical process industry.

8.6. Perturbation

Model-based gradient estimation methods in general do not require additional perturbation signals to estimate the steady-state cost gradient, whereas all model-free gradient estimation methods require additional perturbations in order to estimate the steady cost gradient accurately. As a rule of thumb, the amplitude of the perturbations must be such that the signal-to-noise ratio $SNR > 1$. For multivariable processes, it is also important that the different inputs are perturbed with unique frequencies in order to be able to extract the gradient from each input channel. Although some dither-free (Hunnekens et al., 2014) or diminishing dither approaches (Wang et al., 2016) have been studied in the literature, in practice some kind of additional perturbation is always required in order to track changes in the optimum. In many process industries however, the additional perturbations may not be desirable, since this can affect the product flow and quality specifications, and degrade the process equipment such as valves and pumps leading to frequent maintenance and production shutdown. This potentially limits the applicability of model-free gradient estimation methods in some process applications.

8.7. Ease of implementation and tuning

Development of rigorous nonlinear models is a major bottleneck for implementing model-based methods. Once the models are available, the implementation is straightforward. Meth-

ods such as self-optimizing control using linear gradient combination is perhaps the easiest to implement, since it uses only standard controllers such as PID, that have been in use for several decades. Standard PID tuning rules, such as the SIMC tuning rules (Skogestad, 2003) can be used to tune the PID controllers.

Model-free methods, on the other hand, circumvents the challenge of developing rigorous models. However, model-free gradient estimation algorithms may require several tuning parameters, such as the controller gain, perturbation frequency and amplitude. In addition to these, the different methods have their own tuning parameters. For example, the classical extremum seeking control requires tuning the cut-off frequencies for the high-pass and low-pass filters. The finite difference method has the time period T as a tuning parameter. The least squares method, FFT-based gradient estimation, and the dynamic model identification requires tuning the size of the moving window length N . The dynamic model identification also requires choosing the model structure and the model order, which may not be trivial. All these tuning parameters affect the accuracy and robustness of the gradient estimation, and the speed of convergence to the optimum. The model-free gradient estimation methods are predominantly tuned using trial and error method, and the lack of tuning guidelines can make the implementation non-trivial.

8.8. Handling nonconvexity

So far, we assumed that the stationary point \mathbf{u}^* of the unconstrained optimization problem (5) is also the local minimum (cf. Assumption 1). It may happen that this assumption may not hold for some processes. In this case, when we estimate the steady-state cost gradient and drive it to a constant setpoint of zero, this does not always correspond to the optimal operating point. By driving the cost gradient to zero, the system in principle only converges to a stationary point. This means that if Assumption 1 does not hold, then the stationary point may be a saddle point or even a local maximum.

Bayesian optimization is an alternative model-free approach that can be used to drive the processes to the global minimum without the need for detailed process models. Bayesian optimization is a black-box optimization approach where the real-time cost measurement is used to update a Gaussian process (GP) model. That is, it builds a surrogate model for the objective function, which is updated online by conditioning on the observed cost measurements. An acquisition function defined using the GP surrogate model is then optimized to compute the next control input (Mockus, 2012; Jones et al., 1998; Frazier, 2018; Shahriari et al., 2015). The acquisition function is chosen to trade-off exploitation vs. exploration in order to find the global minimum. Bayesian optimization can be seen as a model-free approach that drives the processes to its global minimum using only the real-time cost measurement. However, it is not strictly a feedback control problem, since it involves optimizing the acquisition function at each time step to compute the next input.

8.9. Scalability to large-scale systems

The scalability of feedback-optimization in general is rather limited compared to traditional RTO framework, and there are different facets to the scalability issue, as discussed below:

Complicated control structure design-With large multivariable plants with several inputs and constraints, the number of relevant active constraint regions increases. This implies to poor scalability for region-based control (paradigm 1) discussed in Section 2.1. Firstly, identifying all the relevant active constraint regions *a priori* can be challenging. Secondly, designing control structures for each active constraint region and designing switching strategies

between the different control configurations can lead to a complicated and messy control structure design. Furthermore, it may also happen that it may not be possible to find a pairing for the selectors that is unique in all constraint regions. This makes maintenance and monitoring of the control loops more challenging, and perhaps one would then be better off with the traditional RTO framework.

Alternatively, the scalability issues can be handled by using the primal-dual feedback control paradigm detailed in Section 2.2. This is because, this approach does not require identifying the active constraint regions *a priori*. Secondly, the CV-MV pairing is rather straightforward in this approach as discussed in Section 6.2. Furthermore, the controlled variables for both the dual controllers and the primal controllers remain the same throughout (cf. (14)). This was also one of the main motivations behind the distributed feedback optimizing control structure proposed by Krishnamoorthy (2021a)

Multivariable gradient estimation - In most model-free gradient estimation methods, the gradient estimation gets more challenging as the number of inputs increases. This is due to the unique perturbation frequency required to extract the effect of the input on the cost measurement. Due to the time scale separation requirement, the integral gain is limited by the slowest perturbation frequency, affecting the convergence speed. Furthermore, in large-scale processes, the cost may be measured several units downstream from the input perturbation. This causes additional time delays in the process dynamics, which subsequently affects the convergence speed due to the time scale separation requirement. In such large scale processes, it is also important to ensure that the regulatory control loops that does not have an impact on steady-state economics (such as level control or pressure control), does not attenuate or amplify the perturbations, as this can lead to erroneous gradient estimation. Therefore, the model-free gradient estimation algorithms typically are more suited for unit operations, where the cost is measured locally.

9. Summary

In this paper we provided a comprehensive overview of the different approaches that aims to achieve asymptotic optimal operation using feedback control without the need to solve numerical optimization problems online. We firstly conclude this review by pointing out that there exists a wide array of different methods that can be used to achieve asymptotic optimal process operation using feedback control, as an alternative to traditional steady-state RTO. Secondly, we showed that the different methods have their own advantages and disadvantages, and differ on key properties ranging from the choice of controlled variables, the degrees of freedom, type of measurements used, the need for process models, convergence speed, accuracy, ease of implementation, scalability etc. Understanding the key distinguishing properties of the different feedback optimizing control methods is important to choose the right tool for the right problem, such that the benefits of online process optimization can be exploited in a wide range of applications.

Declaration of Competing Interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further con-

firm that the order of authors listed in the manuscript has been approved by all of us.

We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.

We understand that the Corresponding Author is the sole contact for the Editorial process (including Editorial Manager and direct communications with the office). He/she is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs.

Acknowledgment

DK would like to thank the European Federation of Chemical Engineers (EFCE) Computer-Aided Process Engineering (CAPE) working party for the invitation to submit a guest paper in connection to the 2020 EFCE Excellence in CAPE PhD Thesis Award.

The authors gratefully acknowledge financial support from the Research Council of Norway through the IKTPLUSS program (project number 299585) and SFI SUBPRO (project number 237893).

Appendix A. Loss due to back-off

Consider the optimization problem (1), but with the set of active constraints given by additional back-off

$$\mathbf{C}_A(\tilde{\mathbf{u}}, \mathbf{d}) + \boldsymbol{\varepsilon} = 0$$

The Lagrangian function is then given as

$$\mathcal{L}(\lambda, \tilde{\mathbf{u}}, \mathbf{d}, \boldsymbol{\varepsilon}) = J(\tilde{\mathbf{u}}, \mathbf{d}) + \lambda_A^\top (\mathbf{C}_A(\tilde{\mathbf{u}}, \mathbf{d}) + \boldsymbol{\varepsilon}) \quad (\text{A.1})$$

Assuming there exists a unique solution for each $\boldsymbol{\varepsilon}$, the necessary conditions of optimality

$$\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{u}}} = \frac{\partial J}{\partial \tilde{\mathbf{u}}} + \lambda_A^\top \frac{\partial \mathbf{C}_A}{\partial \tilde{\mathbf{u}}} = 0 \quad (\text{A.2a})$$

$$\mathbf{C}_A(\tilde{\mathbf{u}}, \mathbf{d}) + \boldsymbol{\varepsilon} = 0 \quad (\text{A.2b})$$

determines the optimal primal and dual variables $\tilde{\mathbf{u}}^*(\boldsymbol{\varepsilon})$ and $\lambda_A^*(\boldsymbol{\varepsilon})$ respectively, as a function of the back-off parameter $\boldsymbol{\varepsilon}$.

Since \mathbf{C}_A depends on $\boldsymbol{\varepsilon}$ through $\tilde{\mathbf{u}}$, differentiating (A.2b) w.r.t $\boldsymbol{\varepsilon}$ gives

$$\frac{\partial \mathbf{C}_A}{\partial \tilde{\mathbf{u}}} \frac{\partial \tilde{\mathbf{u}}^\top}{\partial \boldsymbol{\varepsilon}} + 1 = 0 \quad (\text{A.3})$$

Let the optimal value function be denoted as $J^*(\boldsymbol{\varepsilon}) = J(\tilde{\mathbf{u}}^*(\boldsymbol{\varepsilon}), \mathbf{d})$, and the sensitivity of the optimal value function w.r.t $\boldsymbol{\varepsilon}$ can be expressed as

$$\frac{\partial J^*(\boldsymbol{\varepsilon})}{\partial \boldsymbol{\varepsilon}} = \frac{\partial J}{\partial \tilde{\mathbf{u}}} \frac{\partial \tilde{\mathbf{u}}^\top}{\partial \boldsymbol{\varepsilon}} \quad (\text{A.4})$$

From (A.2a) and (A.3), this can be rewritten as,

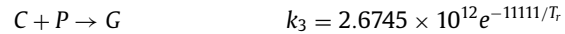
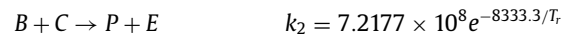
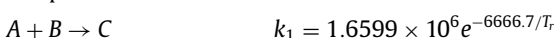
$$\frac{\partial J^*(\boldsymbol{\varepsilon})}{\partial \boldsymbol{\varepsilon}} = -\lambda_A^\top \frac{\partial \mathbf{C}}{\partial \tilde{\mathbf{u}}} \frac{\partial \tilde{\mathbf{u}}^\top}{\partial \boldsymbol{\varepsilon}} = \lambda_A^\top \quad (\text{A.5})$$

Therefore it can be seen that the sensitivity of the optimal value function is given by the Lagrange multiplier.

Appendix B. Williams-Otto reactor model

B1. Williams-Otto reactor model with 3 reactions

The Williams-Otto reactor model with three reactions and six components:



and is given by the following set of ordinary differential equations:

$$\dot{x}_A = \frac{1}{m_r} (F_a - (F_a + F_b)x_A) - x_A x_B k_1 \quad (\text{B.1})$$

$$\dot{x}_B = \frac{1}{m_r} (F_b - (F_a + F_b)x_B) - x_A x_B k_1 - x_B x_C k_2 \quad (\text{B.2})$$

$$\dot{x}_C = -\frac{1}{m_r} (F_a + F_b)x_C + 2x_A x_B k_1 - 2x_B x_C k_2 - x_C x_P k_3 \quad (\text{B.3})$$

$$\dot{x}_P = -\frac{1}{m_r} (F_a + F_b)x_P + x_B x_C k_2 - 0.5x_P x_C k_3 \quad (\text{B.4})$$

$$\dot{x}_E = -\frac{1}{m_r} (F_a + F_b)x_E + 2x_B x_C k_2 \quad (\text{B.5})$$

$$\dot{x}_G = -\frac{1}{m_r} (F_a + F_b)x_G + 1.5x_P x_C k_3 \quad (\text{B.6})$$

where $m_r = 2105$ kg is the reactor holdup and $x_{(\cdot)}$ are the component mass fractions. This set of equations is used to simulate the plant through out this paper, and is also used as the model in Sections 7.1 and 7.4.

B2. Approximate Williams-Otto reactor model with 2 reactions

Plant-model mismatch is demonstrated by using a two-reaction approximation of the Williams-Otto reactor, which considers the following two reactions with five components:



and is given by the following set of ordinary differential equations:

$$\dot{x}_A = \frac{1}{m_r} (F_a - (F_a + F_b)x_A) - x_A x_B^2 k_1 - x_A x_B x_P k_2 \quad (\text{B.7})$$

$$\dot{x}_B = \frac{1}{m_r} (F_b - (F_a + F_b)x_B) - 2x_A x_B^2 k_1 - x_B x_B x_P k_2 \quad (\text{B.8})$$

$$\dot{x}_P = -\frac{1}{m_r} (F_a + F_b)x_P + x_A x_B^2 k_1 - x_A x_B x_P k_2 \quad (\text{B.9})$$

$$\dot{x}_E = -\frac{1}{m_r} (F_a + F_b)x_E + 2x_A x_B^2 k_2 \quad (\text{B.10})$$

$$\dot{x}_G = -\frac{1}{m_r} (F_a + F_b)x_G + 3x_A x_B x_P k_2 \quad (\text{B.11})$$

This set of equations is used as the model in Section 7.3, whereas the plant is still given by the three reaction model.

References

- Adetola, V., Guay, M., 2007. Parameter convergence in adaptive extremum-seeking control. *Automatica* 43 (1), 105–110.
- Alstad, V., 2005. Studies on selection of controlled variables PhD thesis.
- Alstad, V., Skogestad, S., 2007. Null space method for selecting optimal measurement combinations as controlled variables. *Ind. Eng. Chem. Res.* 46 (3), 846–853.
- Alstad, V., Skogestad, S., Hori, E.S., 2009. Optimal measurement combinations as controlled variables. *J. Process Control* 19 (1), 138–148.

- Ariyur, K.B., Krstic, M., 2003. *Real-Time Optimization by Extremum-Seeking Control*. John Wiley & Sons.
- Arkun, Y., Stephanopoulos, G., 1980. Studies in the synthesis of control structures for chemical processes: Part IV. design of steady-state optimizing control structures for chemical process units. *AIChE J.* 26 (6), 975–991.
- Atta, K.T., Johansson, A., Gustafsson, T., 2015. Extremum seeking control based on phasor estimation. *Syst. Control Lett.* 85, 37–45.
- Bamberger, W., Isermann, R., 1978. Adaptive on-line steady-state optimization of slow dynamic processes. *Automatica* 14 (3), 223–230.
- Beaudoin, J.-F., Cadot, O., Aider, J.-L., Wesfreid, J.E., 2006. Bluff-body drag reduction by extremum-seeking control. *J. Fluids Struct.* 22 (6–7), 973–978.
- Bonnowitz, H., Straus, J., Krishnamoorthy, D., Jahanshahi, E., Skogestad, S., 2018. Control of the steady-state gradient of an ammonia reactor using transient measurements. *Comput.-Aided Chem. Eng.* 43, 1111–1116.
- Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations Trends® Mach. Learn.* 3 (1), 1–122.
- Brdýš, M., Tatjewski, P., 1995. An algorithm for steady-state optimizing dual control of uncertain plants. In: *New Trends in Design of Control Systems 1994*. Elsevier, pp. 215–220.
- Câmara, M.M., Quelhas, A.D., Pinto, J.C., 2016. Performance evaluation of real industrial RTO systems. *Processes* 4 (4), 44.
- Chachuat, B., Srinivasan, B., Bonvin, D., 2009. Adaptation strategies for real-time optimization. *Comput. Chem. Eng.* 33 (10), 1557–1567.
- Chioua, M., Srinivasan, B., Guay, M., Perrier, M., 2016. Performance improvement of extremum seeking control using recursive least square estimation with forgetting factor. *IFAC-PapersOnLine* 49 (7), 424–429.
- Choi, J., Krstic, M., Ariyur, K., Lee, J., 2002. Extremum seeking control for discrete-time systems. *IEEE Trans. Automat. Control* 47 (2), 318–323.
- Corti, E., Forte, C., Mancini, G., Moro, D., 2014. Automatic combustion phase calibration with extremum seeking approach. *J. Eng. Gas Turbine. Power* 136 (9).
- Darby, M.L., Nikolaou, M., Jones, J., Nicholson, D., 2011. RTO: an overview and assessment of current practice. *J. Process Control* 21 (6), 874–884.
- Dewasme, L., Srinivasan, B., Perrier, M., Wouwer, A.V., 2011. Extremum-seeking algorithm design for fed-batch cultures of microorganisms with overflow metabolism. *J. Process Control* 21 (7), 1092–1104.
- Draper, C.S., Li, Y.T., 1951. *Principles of Optimizing Control Systems and An Application to the Internal Combustion Engine*. American Society of Mechanical Engineers.
- Dürr, H., Stanković, M., Ebenbauer, C., Johansson, K., 2013. Lie bracket approximation of extremum seeking systems. *Automatica* 49 (6), 1538–1552.
- Eckman, D. P., 1945. *Principles of industrial process control*.
- Economou, C.G., Morari, M., Palsson, B.O., 1986. Internal model control: extension to nonlinear system. *Ind. Eng. Chem. Process Des. Dev.* 25 (2), 403–411.
- Fejjer, D., Paganini, F., 2010. Stability of primal–dual gradient dynamics and applications to network optimization. *Automatica* 46 (12), 1974–1981.
- Fisher, W.R., Doherty, M.F., Douglas, J.M., 1988. The interface between design and control. 3. Selecting a set of controlled variables. *Ind. Eng. Chem. Res.* 27 (4), 611–615.
- Forbes, M.G., Patwardhan, R.S., Hamadah, H., Gopaluni, R.B., 2015. Model predictive control in industry: challenges and opportunities. *IFAC-PapersOnLine* 48 (8), 531–538.
- François, G., Srinivasan, B., Bonvin, D., 2005. Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty. *J. Process Control* 15 (6), 701–712.
- François, G., Srinivasan, B., Bonvin, D., 2012. Comparison of six implicit real-time optimization schemes. *Journal Européen des Systèmes Automatisés* 46 (EPFL-ARTICLE-170545), 291–305.
- François, G., Srinivasan, B., Bonvin, D., 2014. Equivalence between neighboring-extremal control and self-optimizing control for the steady-state optimization of dynamical systems. *Ind. Eng. Chem. Res.* 53 (18), 7470–7478.
- Frazier, P. I., 2018. *A tutorial on Bayesian optimization*. arXiv preprint arXiv:1807.02811.
- Fu, L., Özgüner, Ü., 2011. Extremum seeking with sliding mode gradient estimation and asymptotic regulation for a class of nonlinear systems. *Automatica* 47 (12), 2595–2603.
- Gao, W., Engell, S., 2005. Iterative set-point optimization of batch chromatography. *Comput. Chem. Eng.* 29 (6), 1401–1409.
- Gao, W., Wenzel, S., Engell, S., 2016. A reliable modifier-adaptation strategy for real-time optimization. *Comput. Chem. Eng.* 91, 318–328.
- Garcia, C.E., Morari, M., 1981. Optimal operation of integrated processing systems. Part I: open-loop on-line optimizing control. *AIChE J.* 27 (6), 960–968.
- Gelbert, G., Moeck, J.P., Paschereit, C.O., King, R., 2012. Advanced algorithms for gradient estimation in one-and two-parameter extremum seeking controllers. *J. Process Control* 22 (4), 700–709.
- Ghaffari, A., Krstić, M., Nešić, D., 2012. Multivariable newton-based extremum seeking. *Automatica* 48 (8), 1759–1767.
- Ghaffari, A., Seshagiri, S., Krstić, M., 2015. Multivariable maximum power point tracking for photovoltaic micro-converters using extremum seeking. *Control Eng. Pract.* 35, 83–91.
- Golden, M.P., Ydstie, B.E., 1989. Adaptive extremum control using approximate process models. *AIChE J.* 35 (7), 1157–1169.
- Govatsmark, M.S., Skogestad, S., 2005. Selection of controlled variables and robust setpoints. *Ind. Eng. Chem. Res.* 44 (7), 2207–2217.
- Gros, S., Srinivasan, B., Bonvin, D., 2009. Optimizing control based on output feedback. *Comput. Chem. Eng.* 33 (1), 191–198.
- Halvorsen, I.J., Skogestad, S., Morud, J.C., Alstad, V., 2003. Optimal selection of controlled variables. *Ind. Eng. Chem. Res.* 42 (14), 3273–3284.
- Hauswirth, A., Zanardi, A., Bolognani, S., Dörfler, F., Hug, G., 2017. Online optimization in closed loop on the power flow manifold. In: *2017 IEEE Manchester PowerTech*. IEEE, pp. 1–6.
- Henning, L., Becker, R., Feuerbach, G., Muminovic, R., King, R., Brunn, A., Nitsche, W., 2008. Extensions of adaptive slope-seeking for active flow control. *Proc. Inst. Mech. Eng. Part I* 222 (5), 309–322.
- Hunnekens, B., Haring, M., van de Wouw, N., Nijmeijer, H., 2014. A dither-free extremum-seeking control approach using 1st-order least-squares fits for gradient estimation. In: *53rd IEEE Conference on Decision and Control*. IEEE, pp. 2679–2684.
- Jacobsen, M.G., Skogestad, S., 2012. Active constraint regions for optimal operation of distillation columns. *Ind. Eng. Chem. Res.* 51 (7), 2963–2973.
- Jäschke, J., Cao, Y., Kariwala, V., 2017. Self-optimizing control—a survey. *Annu. Rev. Control*.
- Jäschke, J., Skogestad, S., 2011. NCO tracking and self-optimizing control in the context of real-time optimization. *J. Process Control* 21 (10), 1407–1416.
- Jäschke, J., Skogestad, S., 2012. Optimal controlled variables for polynomial systems. *J. Process Control* 22 (1), 167–179.
- Jones, D.R., Schonlau, M., Welch, W.J., 1998. Efficient global optimization of expensive black-box functions. *J. Global Optim.* 13 (4), 455–492.
- Kelly, J.D., Hedengren, J.D., 2013. A steady-state detection (SSD) algorithm to detect non-stationary drifts in processes. *J. Process Control* 23 (3), 326–331.
- van Keulen, T., van der Weijst, R., Oomen, T., 2020. Fast extremum seeking using multisine dither and online complex curve fitting. *IFAC-PapersOnLine* 53 (2), 5362–5367.
- Killingsworth, N.J., Krstic, M., 2006. PID tuning using extremum seeking: online, model-free performance optimization. *IEEE Control Syst. Mag.* 26 (1), 70–79.
- Krishnamoorthy, D., 2019. *Novel approaches to online process optimization under uncertainty*. Norwegian University of Science Technology Ph.D. thesis.
- Krishnamoorthy, D., 2021. A distributed feedback-based online process optimization framework for optimal resource sharing. *J. Process Control* 97, 72–83.
- Krishnamoorthy, D., 2021b. *On the design and analysis of multivariable extremum seeking control using fast fourier transform*. arXiv preprint arXiv:2104.14365.
- Krishnamoorthy, D., Fjalestad, K., Skogestad, S., 2019. Optimal operation of oil and gas production using simple feedback control structures. *Control Eng. Pract.* xx, xxx–xxx.
- Krishnamoorthy, D., Foss, B., Skogestad, S., 2018. Steady-state real-time optimization using transient measurements. *Comput. Chem. Eng.* 115, 34–45.
- Krishnamoorthy, D., Jahanshahi, E., Skogestad, S., 2018. Gas-lift optimization by controlling marginal gas-oil ratio using transient measurements. *IFAC-PapersOnLine* 51 (8), 19–24.
- Krishnamoorthy, D., Jahanshahi, E., Skogestad, S., 2019. Control of the steady-state gradient of an evaporator process using transient measurements. *PSE Asia*. (In-press)
- Krishnamoorthy, D., Jahanshahi, E., Skogestad, S., 2019. A feedback real time optimization strategy using a novel steady-state gradient estimate and transient measurements. *Ind. Eng. Chem. Res.* 58, 207–216.
- Krishnamoorthy, D., Pavlov, A., Li, Q., 2016. Robust extremum seeking control with application to gas lifted oil wells. *IFAC-PapersOnLine* 49 (13), 205–210.
- Krishnamoorthy, D., Skogestad, S., 2019. Online process optimization with active constraint set changes using simple control structures. *Ind. Eng. Chem. Res.*
- Krishnamoorthy, D., Skogestad, S., 2020. Linear gradient combination as self-optimizing variables. *Comput. Aided Chem. Eng.*. In-Press
- Krishnamoorthy, D., Skogestad, S., 2020. Systematic design of active constraint switching using selectors. *Comput. Chem. Eng.* 143, 107106.
- Krstić, M., 2000. Performance improvement and limitations in extremum seeking control. *Syst. Control Lett.* 39 (5), 313–326.
- Krstić, M., Wang, H.-H., 2000. Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica* 36 (4), 595–601.
- Kumar, V., Kaistha, N., 2014. Hill-climbing for plantwide control to economic optimum. *Ind. Eng. Chem. Res.* 53 (42), 16465–16475.
- Larsson, T., Skogestad, S., 2000. Plantwide control—a review and a new design procedure. *Model. Identif. Control* 21 (4), 209.
- Maarleveld, A., Rijnsdorp, J., 1970. Constraint control on distillation columns. *Automatica* 6 (1), 51–58.
- Manum, H., Skogestad, S., 2012. Self-optimizing control with active set changes. *J. Process Control* 22 (5), 873–883.
- Marchetti, A., de Avila Ferreira, T., Costello, S., Bonvin, D., 2020. Modifier adaptation as a feedback control scheme. *Ind. Eng. Chem. Res.* 59 (6), 2261–2274.
- Marchetti, A., Chachuat, B., Bonvin, D., 2009. Modifier-adaptation methodology for real-time optimization. *Ind. Eng. Chem. Res.* 48 (13), 6022–6033.
- Marchetti, A.G., François, G., Faulwasser, T., Bonvin, D., 2016. Modifier adaptation for real-time optimization—methods and applications. *Processes* 4 (4), 55.
- Marinkov, S., de Jager, B., Steinbuch, M., 2014. Extremum seeking control with data-based disturbance feedforward. In: *2014 American Control Conference*, pp. 3627–3632. doi:10.1109/ACC.2014.6858832.
- Marlin, T.E., Hrymak, A.N., 1997. Real-time operations optimization of continuous processes. In: *AIChE Symposium Series*, vol. 93. New York, NY: American Institute of Chemical Engineers, 1971–c2002., pp. 156–164.
- Matias, J. O., Jäschke, J., 2019. Using a neural network for estimating plant gradients in real-time optimization with modifier adaptation.
- McFarlane, R., Bacon, D., 1989. Empirical strategies for open-loop on-line optimization. *Can. J. Chem. Eng.* 67 (4), 665–677.

- Mockus, J., 2012. Bayesian Approach to Global Optimization: Theory and Applications, vol. 37. Springer Science & Business Media.
- Morari, M., Arkun, Y., Stephanopoulos, G., 1980. Studies in the synthesis of control structures for chemical processes: Part I: formulation of the problem. Process decomposition and the classification of the control tasks. Analysis of the optimizing control structures. *AIChE J.* 26 (2), 220–232.
- de Oliveira, V., Jaschke, J., Skogestad, S., 2015. Neighbouring-extremal control for steady-state optimization using noisy measurements. *IFAC-PapersOnLine* 48 (8), 698–703.
- Oliveira-Silva, E., de Prada, C., Navia, D., 2021. Economic MPC with modifier adaptation using transient measurements. In: *Computer Aided Chemical Engineering*, Vol. 50. Elsevier, pp. 1253–1258.
- Pan, Y., Özgüner, Ü., Acarman, T., 2003. Stability and performance improvement of extremum seeking control with sliding mode. *Int J Control* 76 (9–10), 968–985.
- Pavlov, A., Haring, M., Fjalestad, K., 2017. Practical extremum-seeking control for gas-lifted oil production. In: *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. IEEE, pp. 2102–2107.
- Reyes-Lúa, A., Skogestad, S., 2019. Multiple-input single-output control for extending the steady-state operating range—use of controllers with different setpoints. *Processes* 7 (12), 941.
- Reyes-Lúa, A., Skogestad, S., 2019. Systematic design of active constraint switching using classical advanced control structures. *Ind. Eng. Chem. Res.*
- Reyes-Lúa, A., Skogestad, S., 2020. Multi-input single-output control for extending the operating range: generalized split range control using the baton strategy. *J. Process Control* 91, 1–11.
- Reyes-Lúa, A., Zotică, C., Das, T., Krishnamoorthy, D., Skogestad, S., 2018. Changing between active constraint regions for optimal operation: classical advanced control versus model predictive control. In: *Computer Aided Chemical Engineering*, vol. 43. Elsevier, pp. 1015–1020.
- del Rio Chanona, E. A., Alves Graciano, J. E., Bradford, E., Chachuat, B., 2019. Modifier-adaptation schemes employing gaussian processes and trust regions for real-time optimization.
- Roberts, P., 2000. Broyden derivative approximation in ISOPE optimising and optimal control algorithms. *IFAC Proc.* Vol. 33 (16), 293–298.
- Roberts, P.D., Williams, T., 1981. On an algorithm for combined system optimisation and parameter estimation. *Automatica* 17 (1), 199–209.
- Rodger, E.A., Chachuat, B., 2011. Design methodology of modifier adaptation for on-line optimization of uncertain processes. *IFAC Proc.* Vol. 44 (1), 4113–4118.
- Seborg, D.E., Mellichamp, D.A., Edgar, T.F., Doyle III, F.J., 2010. *Process Dynamics and Control*. John Wiley & Sons.
- Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., De Freitas, N., 2015. Taking the human out of the loop: a review of Bayesian optimization. *Proc. IEEE* 104 (1), 148–175.
- Shinskey, F.G., 1996. *Process Control Systems: Application, Design, and Tuning*, vol. 4. McGraw-Hill New York.
- Shook, D., 2006. Best practices improve control system performance. *Oil Gas J.* 104 (38).
- Silva, T.L., Pavlov, A., 2020. Dither signal optimization for multi-agent extremum seeking control. In: *2020 European Control Conference (ECC)*. IEEE, pp. 1230–1237.
- Skogestad, S., 2000. Plantwide control: the search for the self-optimizing control structure. *J. Process Control* 10 (5), 487–507.
- Skogestad, S., 2003. Simple analytic rules for model reduction and PID controller tuning. *J. Process Control* 13 (4), 291–309.
- Skogestad, S., 2004. Control structure design for complete chemical plants. *Comput. Chem. Eng.* 28 (1), 219–234.
- Skogestad, S., Postlethwaite, I., 2007. *Multivariable Feedback Control: Analysis and Design*, 2 ed. Wiley New York.
- Srinivasan, B., 2007. Real-time optimization of dynamic systems using multiple units. *Int. J. Robust Nonlinear Control* 17 (13), 1183–1193.
- Srinivasan, B., Bonvin, D., 2019. 110th Anniversary: a feature-based analysis of static real-time optimization schemes. *Ind. Eng. Chem. Res.* 58 (31), 14227–14238.
- Straus, J., Krishnamoorthy, D., Skogestad, S., 2019. Combining self-optimizing control and extremum seeking control - applied to ammonia reactor case study. *J. Process Control* 78, 78–87.
- Tan, Y., Moase, W., Manzie, C., Nešić, D., Mareels, I., 2010. Extremum seeking from 1922 to 2010. In: *Control Conference (CCC), 2010 29th Chinese*. IEEE, pp. 14–26.
- Trollberg, O., Jacobsen, E.W., 2016. Greedy extremum seeking control with applications to biochemical processes. *IFAC-PapersOnLine (DYCOPS-CAB)* 49 (7), 109–114.
- Vaccari, M., Pelagagge, F., Bonvin, D., Pannocchia, G., 2020. Estimation technique for offset-free economic MPC based on modifier adaptation. *IFAC-PapersOnLine* 53 (2), 11251–11256.
- Venets, V., 1985. Continuous algorithms for solution of convex optimization problems and finding saddle points of convex-concave functions with the use of projection operations. *Optimization* 16 (4), 519–533.
- Wang, L., Chen, S., Ma, K., 2016. On stability and application of extremum seeking control without steady-state oscillation. *Automatica* 68, 18–26.
- Williams, T.J., Otto, R.E., 1960. A generalized chemical processing model for the investigation of computer control. *Trans. Am. Inst. Electr. Eng. Part I* 79 (5), 458–473.
- Ye, L., Cao, Y., Li, Y., Song, Z., 2012. Approximating necessary conditions of optimality as controlled variables. *Ind. Eng. Chem. Res.* 52 (2), 798–808.
- Ye, L., Cao, Y., Skogestad, S., 2017. Global self-optimizing control for uncertain constrained process systems. *IFAC-PapersOnLine* 50 (1), 4672–4677.
- Ye, L., Cao, Y., Yuan, X., 2015. Global approximation of self-optimizing controlled variables with average loss minimization. *Ind. Eng. Chem. Res.* 54 (48), 12040–12053.
- Yelchuru, R., Skogestad, S., 2012. Convex formulations for optimal selection of controlled variables and measurements using mixed integer quadratic programming. *J. Process Control* 22 (6), 995–1007.