Real-Time Optimization as a Feedback Control Problem - A Review

Dinesh Krishnamoorthy*, Sigurd Skogestad

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

Abstract

Feedback optimizing control aims to achieve optimal process operation by directly manipulating the inputs using feedback controllers, without the need to solve numerical optimization problems online. The main question here is *what to control* such that the economic objectives are translated into control objectives. If some of the constraints are optimally at their limiting values, then the simplest and obvious choice is to control the constraints itself, known as active constraint control. For the remaining unconstrained degrees of freedom, it is not obvious what to control, and one requires additional knowledge either in the form of process models or optimization-specific measurements. Feedback optimizing control also requires switching between different active constraint regions as the operating conditions change. Over the past two decades, active research has resulted in several different methods stemming from different application domains, 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, convergence speed and 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.

Keywords: Real-time optimization, feedback control, gradient estimation, active constraint switching

1. Introduction

In the face of growing market competition and the increased focus on sustainability, energy efficiency and safety, online optimization of the process operation is becoming increasingly important. Real-time optimization 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 [1, 2, 3].

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 steady-state time periods, in order to match the plant measurements and the model predictions. The updated model is then used to re-optimize for the new setpoints [4, Chapter 19],[2]. 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.

The repeated identification and optimization scheme using steady-state models is used in most of the current commercial RTO software packages [5]. The justification for this approach is that for most continuous processes, the economic operation of the plant often occurs at steady-state. The exception is processes with frequent grade changes, cyclic operations etc. As such, the objective in most continuous processes is simply to find the economically optimal steady-state operating point as a function of the disturbances. Steady-state process optimization for continuous processes will be the focus in this paper.

The decision variables for the RTO layer are typically the setpoints for the controlled variables $\mathbf{y}^{sp}$, which are then given to a setpoint tracking control layer below. The control layer adjusts the manipulated variables $\mathbf{u}$ in order to keep the measurements $\mathbf{y}$ at the optimal setpoints $\mathbf{y}^{sp}$ computed by the RTO layer. Fig. 2 shows the typical

*Corresponding author

Email addresses: dinesh.krishnamoorthy@ntnu.no (Dinesh Krishnamoorthy), sigurd.skogestad@ntnu.no (Sigurd Skogestad)

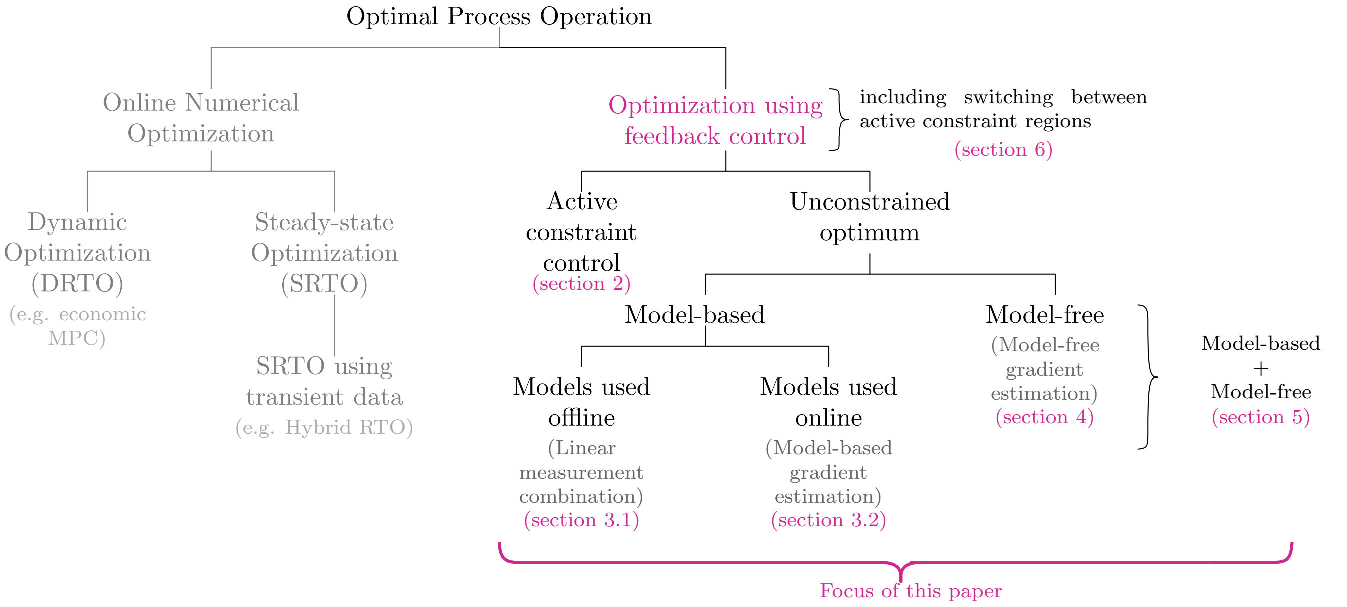

Optimal Process Operation
Online Numerical Optimization
Optimization using feedback control
including switching between active constraint regions
(section 6)
Dynamic Optimization (DRTO)
(e.g. economic MPC)
Steady-state Optimization (SRTO)
SRTO using transient data
(e.g. Hybrid RTO)
Active constraint control
(section 2)
Unconstrained optimum
Model-based
Model-free
(Model-free gradient estimation)
(section 4)
Model-based + Model-free
(section 5)
Models used offline
(Linear measurement combination)
(section 3.1)
Models used online
(Model-based gradient estimation)
(section 3.2)
Focus of this paper

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

hierarchical decomposition into optimization and control layers.

Despite the economic benefits and promises, traditional real-time optimization is not used commonly in practice. Consequently, the full potential of RTO is not exploited in process industries [6, 1, 7]. The main technical challenges which limits the 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 lead to mismatch between the physical models used in the optimizer 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 adaptation step must be carried out using measurements that corresponds to steady-state operation. A steady-state detection algorithm is used to detect if the process is operating at steady-state conditions. 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. [7] concludes that a fundamental limiting factor of RTO implementation is the steady-state wait-time associated with the online update of the model. Câmara et al. [5] 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 [2].

In light of these challenges, feedback optimizing control has been seen as a promising alternative to the traditional RTO paradigm. Achieving optimal operation via feedback 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 simple feedback controllers. Finally, as will be shown later, many feedback optimizing control approaches such as active constraint control, extremum seeking control etc. do 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 is one 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, 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 [8, 9]. As Marlin and Hrymak [2] pointed out, “the greatest demand from optimization users is a system that can be easily understood”. For this reason, many process industries may prefer to use simple

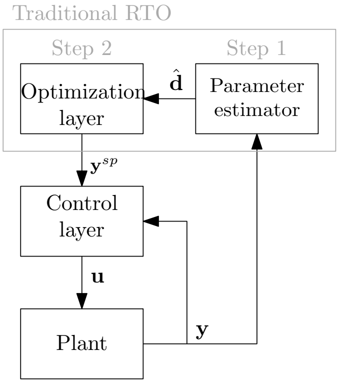


Figure 2: Traditional hierarchical decomposition into optimization and control layers

feedback control tools available in standard digital control system (DCS) to optimize process operations [10]. In such cases, feedback optimizing control provides a systematic procedure that enables optimal process operation using simple feedback control loops.

As shown in Fig. 1, online process optimization approaches can be categorized into two classes, based on whether one needs to solve numerical optimization problems online, or not. Traditional steady-state RTO and dynamic RTO explicitly solves static and dynamic optimization problems, respectively, in order to compute the optimal solution. In order to circumvent the need for solving numerical optimization problems online, several different algorithms have been developed, where the numerical optimization problem is converted into a feedback control problem. The main idea is to achieve optimal operation by manipulating the inputs $\mathbf{u}$ based on feedback measurements $\mathbf{y}$. 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*" [11, 3, 12], which is the focus of this article.

2. Feedback optimizing control

In 1980, Morari et al. [11] first introduced the concept of *feedback optimizing control*, where the main objective 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. The most important question is then,

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

Consider the steady-state optimization problem,

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

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, $J : \mathbb{R}^{n_{\tilde{u}}} \times \mathbb{R}^{n_d} \to \mathbb{R}$ is the scalar cost function and $\mathbf{C} : \mathbb{R}^{n_{\tilde{u}}} \times \mathbb{R}^{n_d} \to \mathbb{R}^{n_c}$ denotes the vector of constraints.

Since we have $n_{\tilde{u}}$ degrees of freedom, the main idea behind feedback optimizing control is to find $n_{\tilde{u}}$ suitable controlled variables (CV), and their setpoints at which to control them.

The choice of the optimal controlled variables generally depends on the set of active constraints. If the process is operated optimally, when some of the constraints are active (i.e. at its limiting value), then the easiest choice for the controlled variables are the constraints itself. By tightly controlling the constraints to their limits, the process can be driven to its optimum, which is known as "active constraint control" [13].

Active constraint control. Let $n_a \leq n_c$ denote the number of active constraints $\mathbf{C}_{\mathbb{A}}(\tilde{\mathbf{u}}, \mathbf{d})$ 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. $CV = \mathbf{C}_{\mathbb{A}}$, which are controlled to its limiting value using simple feedback controllers. Note that it is reasonable to assume that the constraints are typically measured for monitoring purposes, and one can detect if the constraints are active or not.

Active constraint control approach is a well known idea and has been used in many examples, see [14, 15, 16, 17, 11, 18, 19, 20] to name a few. With active constraint control, one can easily achieve optimal operation without the need for a model for optimization (online or offline), since the constraints can be maintained at their limits using simple feedback controllers, thereby circumventing all the three challenges mentioned above [19, 13].

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 $\mathbf{C}_{\mathbb{A}}$ are offset by a margin $\Delta\mathbf{C}$. This gives rise to a loss, which increases linearly, and is quantified by the corresponding Lagrange multipliers $\lambda_{\mathbb{A}}$ as,

$$Loss = \lambda_{\mathbb{A}} |\Delta\mathbf{C}| \tag{2}$$

See for example [21, 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 [22].

Unconstrained optimum. A more challenging case is when the solution to the optimization problem at hand is unconstrained. 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 $\mathbf{c} \in \mathbb{R}^{n_c - n_a}$, which can be controlled using a feedback control law.

Skogestad [13] defined self-optimizing control as when we can achieve (near) optimal operation (acceptable loss) with constant setpoint for the self-optimizing controlled variables. This may be a single measurement [13], a linear combination of measurements [23], 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, which when kept at a constant setpoint of zero, satisfies the necessary conditions of optimality [24, 25, 26]. For instance, consider the gradient of the reduced system,

$$\min_{\mathbf{u}} \quad J(\mathbf{u}, \mathbf{d}) \tag{3}$$

where $\mathbf{u} \in \mathbb{R}^{n_u}$ is the vector of unconstrained degrees of freedom. The KKT optimality conditions for (3) state that the necessary condition of optimality is when the gradient of the cost function is zero.

$$\mathbf{J}_u(\mathbf{d}) := \nabla_{\mathbf{u}} J(\mathbf{u}^*, \mathbf{d}) = 0 \tag{4}$$

Assumption 1. *The cost function $J(\mathbf{u}, \mathbf{d})$ has a unique KKT point at $\mathbf{u}^*$ such that (4) holds. Furthermore, we assume that strict second order sufficient condition holds at $\mathbf{u}^*$, that is $\mathbf{u}^*$ is an unique local minimum.*

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 [27] (which is the same as the gradient of the reduced system)

$$\mathbf{J}_u(\mathbf{d}) := \nabla_{\mathbf{u}} J(\mathbf{u}, \mathbf{d}) = \mathbf{N}^T \nabla_{\tilde{\mathbf{u}}} J(\tilde{\mathbf{u}}, \mathbf{d}) = 0 \tag{5}$$

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. See, [27, 19] for detailed description of this approach. For ease of notation, we will use $J_{\mathbf{u}}$ to denote the steady-state gradient of J with respect to the unconstrained degrees of freedom $\mathbf{u}$.

Theorem 1 (Generalized framework for feedback optimizing control [1, 27])**.** *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 $\mathbf{C}_{\mathbb{A}}(\mathbf{d}) = 0$.*
2. *Self-optimizing variables $\mathbf{c} = \mathbf{c}^{sp}$, where*

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

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

Proof. Providing just the sketch of the proof here, it can be seen that enforcing $\mathbf{C}_{\mathbb{A}}(\mathbf{d}) = 0$ leads to feasible operation, and enforcing $\mathbf{c} = \mathbf{J}_u(\mathbf{d}) = \mathbf{N}^T \nabla_{\tilde{\mathbf{u}}} J(\tilde{\mathbf{u}}, \mathbf{d}) = 0$ ensures that the necessary conditions of optimality are satisfied. See [27] for detailed proof. □

Therefore, feedback optimizing control can be classified into two categories, namely, active constraint control and unconstrained optimum (cf. Fig. 1).

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 the electrical and mechanical engineering literature, predominantly in the context of extremum seeking control [28, 29, 30]. The main objective of this paper is to provide an overview and understanding of the different approaches to achieving optimal operation using the unconstrained degrees of freedom, 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 [31, 32, 24]. 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.

A constraint that is active for a given disturbance, may no longer be active when the disturbance changes. This requires reconfiguration of the control loops with a new set of controlled variables. Feedback optimizing control should ideally handle changes in the active constraint set without the need for solving numerical optimization problems. This can be achieved using classical advanced control elements which will also be reviewed in this paper, thus providing a comprehensive overview of feedback optimizing control.

The reminder of the paper is organized as follows: 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 the method where the models are used online to estimate the steady-state cost gradients. Section 4 reviews the model-free methods, where the steady-state cost gradient is estimated directly from the cost measurement, which is then controlled to a constant setpoint of zero to achieve optimal operation. The combination of model-based and model-free approaches are briefly reviewed in Section 5. The use of classical advanced control elements to handle active constraint switching is reviewed in Section 6. 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.
$\mathbf{u}$ - unconstrained degrees of freedom
$\mathbf{y}_m$ - vector of process measurements
J - Cost measurement
J_{model} - Cost predicted by model
$\hat{\mathbf{J}}_{\mathbf{u},model}$ - Model based gradient estimate
$\hat{\mathbf{J}}_{\mathbf{u}}$ - Model-free gradient estimate
CV - Controlled Variable
MV - Manipulated Variable

3. Model-based methods

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

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d}) \tag{6a}$$

$$\mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{d}) \tag{6b}$$

$$J_{model} = \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{d}) \tag{6c}$$

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

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 (6). 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 not be the truly optimal variable to keep constant if we take into account measurement error. An alternative approach is therefore to start from the available measurement $\mathbf{y}$, and look for single measurements or measurement combinations to keep constant. To simplify the mathematical treatment, we usually only consider linear measurement combinations, $\mathbf{c} = \mathbf{Hy}$, 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 [13], and was also studied by Larsson and Skogestad [32] and Govatsmark and Skogestad [22].

For processes with many candidate measurements, it may be tedious and time consuming to evaluate the economic loss for the 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 [24]. Here, the gain matrix is scaled as shown below

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

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 [24]. 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 [33] for amore detailed discussion of this approach.

3.1.2. Linear measurement combination

In this approach, instead of a 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 optimal selection matrix $\mathbf{H}$ using the process models offline. Most approaches are based on local linearization around some nominal optimal point. The two most popular approaches to find the optimal measurement combination are the nullspace method proposed by Alstad and Skogestad [23] and the exact local method proposed by Halvorsen et al. [24] and further developed by Alstad et al. [31] and Yelchuru and Skogestad [34]. Fig. 3 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

$$\mathbf{F} = \begin{bmatrix} \frac{\partial \mathbf{y}^*}{\partial \mathbf{d}} & \frac{\partial \mathbf{u}^*}{\partial \mathbf{d}} \end{bmatrix}^\mathsf{T} \tag{7}$$

Note that we explicitly show the input $\mathbf{u}$ as well, since it is reasonable to assume that it also a measured quantity that can be used to design the self-optimizing variable.

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

$$\mathbf{HF} = 0 \tag{8}$$

Alstad [35, Chapter 3] proved that computing the selection matrix $\mathbf{H}$ such that $\mathbf{HF} = 0$ 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. $\mathbf{c}^{sp} = \mathbf{H}\mathbf{y}_0^*$.

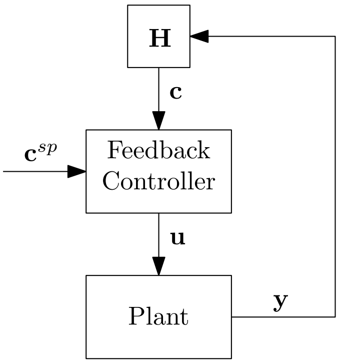


Figure 3: Self-optimizing control using linear measurement combination as the optimal controlled variable, where the self-optimizing controlled variables are identified using the models offline.

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 (3) 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 $\mathbf{d}$

$$J(\mathbf{u},\mathbf{d}) \approx J(\mathbf{u}_0^*,\mathbf{d}) + \underbrace{\mathbf{J_u}^\mathsf{T}\delta\mathbf{u}}_{=0} + \frac{1}{2}\delta\mathbf{u}^\mathsf{T}\mathbf{J_{uu}}\delta\mathbf{u} \tag{9}$$

from which the loss function can be written as

$$Loss = J(\mathbf{u},\mathbf{d}) - J(\mathbf{u}_0^*,\mathbf{d}) = \frac{1}{2}\|\mathbf{z}\|^2 \tag{10}$$

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

Using the linearized measurement model,

$$\delta\mathbf{y} = \underbrace{\frac{\partial \mathbf{g}}{\partial \mathbf{u}}}_{\mathbf{G^y}}\delta\mathbf{u} + \underbrace{\frac{\partial \mathbf{g}}{\partial \mathbf{d}}}_{\mathbf{G_d^y}}\delta\mathbf{d} \tag{11}$$

and

$$\mathbf{c} = \mathbf{H}\mathbf{y}_m = \mathbf{H}\mathbf{y} + \mathbf{H}\mathbf{e^y} \tag{12}$$

we can express

$$\mathbf{z} = \mathbf{J_{uu}}^{1/2}(\mathbf{HG^y})^{-1}\mathbf{H}\underbrace{\begin{bmatrix}\mathbf{FW_d} & \mathbf{W_{e^y}}\end{bmatrix}}_{:=\mathbf{Y}}\begin{bmatrix}-\mathbf{d}' \\ -\mathbf{e}^{y'}\end{bmatrix} \tag{13}$$

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.

The minimization of the average loss can be formulated as,

$$\begin{aligned} \min_{\mathbf{H}} \quad & \|\mathbf{HY}\|_F \\ \text{s.t.} \quad & \mathbf{HG^y} = \mathbf{J_{uu}}^{1/2} \end{aligned} \tag{14}$$

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

$$\mathbf{H}^{\mathsf{T}} = (\mathbf{Y}\mathbf{Y}^{\mathsf{T}})^{-1}\mathbf{G}^{\mathbf{y}} \tag{15}$$

See [31, 33] 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 (7)may be determined numerically by perturbing the disturbances and re-optimizing. Alternatively, one can use the analytical expression given by,

$$\mathbf{F} = \begin{bmatrix} \mathbf{G}^{\mathbf{y}}_{\mathbf{d}} - \mathbf{G}^{\mathbf{y}}\mathbf{J}_{\mathbf{uu}}{}^{-1}\mathbf{J}_{\mathbf{ud}} \\ \mathbf{J}_{\mathbf{uu}}{}^{-1}\mathbf{J}_{\mathbf{ud}} \end{bmatrix} \tag{16}$$

where

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

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. [36, 37]. Ye et al. [38] 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 within self-optimizing control, leading to a survey article focusing on self-optimizing control by Jäschke et al. [33].

3.1.3. Neighboring extremal (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 $\delta J_{\mathbf{u}}$ resulting from the parametric variations $\delta\mathbf{d}$ can be expressed as

$$\delta\mathbf{J}_{\mathbf{u}} = \mathbf{J}_{\mathbf{uu}}\delta\mathbf{u} + \mathbf{J}_{\mathbf{ud}}\delta\mathbf{d} \tag{17}$$

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 the measurement model (11). Hence,

$$\delta\mathbf{d} = (\mathbf{G}^{\mathbf{y}}_{\mathbf{d}})^{\dagger}(\delta\mathbf{y} - \mathbf{G}^{\mathbf{y}}\delta\mathbf{u}) \tag{18}$$

Substituting (18) in (17) results in,

$$\hat{\mathbf{J}}_{\mathbf{u},model} = \mathbf{J}_{\mathbf{ud}}(\mathbf{G}^{\mathbf{y}}_{\mathbf{d}})^{\dagger}\delta\mathbf{y} + \left[\mathbf{J}_{\mathbf{uu}} - \mathbf{J}_{\mathbf{ud}}(\mathbf{G}^{\mathbf{y}}_{\mathbf{d}})^{\dagger}\mathbf{G}^{\mathbf{y}}\right]\delta\mathbf{u} \tag{19}$$

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 (16) with (19), it can be seen that the nullspace method and the neighboring extremal control approaches are equivalent [39]. That is (19) can equivalently be written as

$$\mathbf{c} = \hat{\mathbf{J}}_{\mathbf{u},model} = \mathbf{H}\begin{bmatrix}\delta\mathbf{y} \\ \delta\mathbf{u}\end{bmatrix} \tag{20}$$

where

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

and from (16) one can see that $\mathbf{HF} = 0$, which is the same as in the nullspace method.

Remark 3. *The equivalence 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* $\mathbf{J}_{\mathbf{u}} = 0$ *at nominal conditions, see also [40, Appendix B].*

Although the standard neighboring extremal control [41] does not take into account measurement noise, variations of this approach that considers the measurement noise has been proposed in [42].

3.2. Online model-based gradient estimation

3.2.1. Two-step approach

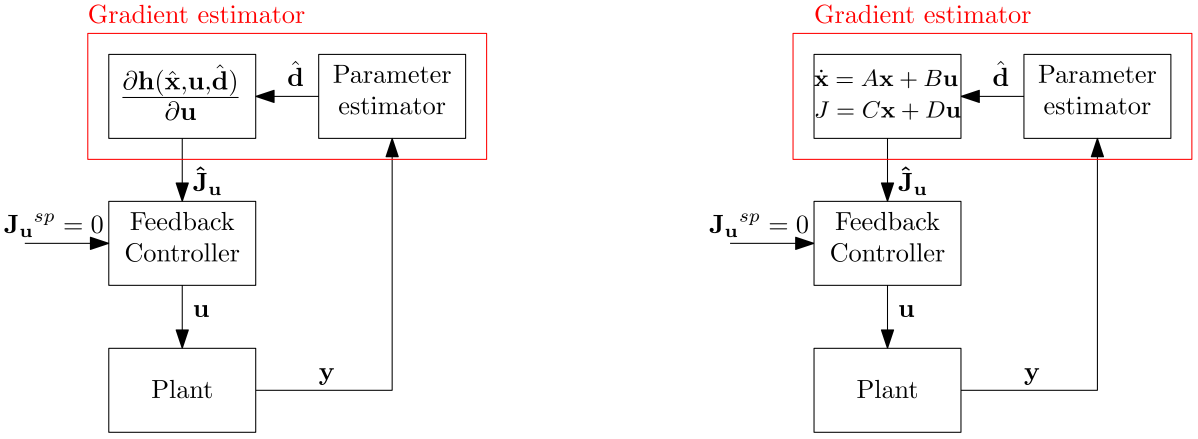
This approach is closely related to the traditional two-step RTO approach, where the steady-state cost gradient is computed analytically using the model equations (6). The disturbances $\mathbf{d}$ are estimated using the measurements $\mathbf{y}_{meas}$ using any parameter estimation algorithm.

$$\begin{bmatrix}\dot{\hat{\mathbf{x}}} \\ \dot{\hat{\mathbf{d}}}\end{bmatrix} = K_e(\mathbf{y}_{meas} - \mathbf{g}(\mathbf{x},\mathbf{u},\mathbf{d})) \tag{21}$$

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 $\hat{\mathbf{J}}_{\mathbf{u},model}$ is then driven to a constant setpoint of zero using any feedback control law. This approach is schematically shown in Fig. 4a. 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 [43].

3.2.2. Feedback RTO using transient measurements (FRTO)

This approach proposed in [44] estimates the steady-state cost gradient using a nonlinear dynamic model and the process measurements 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



(a) (b)

Figure 4: Online model-based gradient-estimation methods: (a) Gradient from adapted model using the two-step approach. (b) Feedback RTO [44]

states and unmeasured disturbances are estimated, (6a) and (6c) are linearized to obtain a *local linear dynamic model* from the inputs $\mathbf{u}$ to the objective function J,

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x}(t) + B\mathbf{u}(t) \\ J_{model}(t) &= C\mathbf{x}(t) + D\mathbf{u}(t) \end{aligned} \tag{22}$$

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

$$A = \left. \frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u},\mathbf{d})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(t_0),\mathbf{d}=\hat{\mathbf{d}}(t_0)}$$

$$B = \left. \frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u},\mathbf{d})}{\partial \mathbf{u}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(t_0),\mathbf{d}=\hat{\mathbf{d}}(t_0)}$$

$$C = \left. \frac{\partial \mathbf{h}(\mathbf{x},\mathbf{u},\mathbf{d})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(t_0),\mathbf{d}=\hat{\mathbf{d}}(t_0)}$$

$$D = \left. \frac{\partial \mathbf{h}(\mathbf{x},\mathbf{u},\mathbf{d})}{\partial \mathbf{u}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(t_0),\mathbf{d}=\hat{\mathbf{d}}(t_0)}$$

assuming that A is invertible, the corresponding steady-state gradient is then given as

$$\hat{\mathbf{J}}_{\mathbf{u},model} = -CA^{-1}B + D \tag{23}$$

This approach is schematically shown in Fig. 4b. Eq. (23) was also used by Bamberger and Isermann [45], Garcia and Morari [46] among others to estimate the steady-state gradient from linear dynamic models. The estimated gradient is then driven to zero using any feedback controller. The reader is referred to [44] 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 [44, 47, 48, 49].

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.

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,

$$\mathbf{u}(t) = \begin{cases} \mathbf{u}_k & 2kT \leq t < (2k+1)T \\ \mathbf{u}_k + \Delta\mathbf{u} & (2k+1)T \leq t < (2k+2)T \end{cases} \tag{24}$$

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.

$$\hat{\mathbf{J}}_{\mathbf{u}} = \frac{\Delta J}{\Delta\mathbf{u}} = \frac{J((2k+2)T) - J((2k+1)T)}{\Delta\mathbf{u}} \tag{25}$$

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 [26, 40] and hill-climbing control [50, 51]. 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 [52] and modifier adaptation [53].

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 inefficent 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 [54], Gao and Engell [55] and Rodger and Chachuat [56] used Broyden's formula to estimate the steady-state gradient from current and past measurement, without the need for additional perturbations. Brdyś and Tatjewski [57] 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 [28]. This approach dates back to the 1950s with the work of Draper and Li [58], where extremum seeking control was studied under the context of adaptive control. This method gained huge popularity since the work of Krstić and Wang [59], 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 [60]. Since 2000s, there has been several advancements in extremum seeking (ES) methods including, least square-based ES [61], sliding-mode ES [62, 63], greedy ES [64], discrete-time ES[65], newton-based ES [66], Lie-bracket approximation based ES [67] 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,

$$u(t) = \hat{u} + a\sin\omega t \tag{26}$$

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 period 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,

$$\dot{\eta} = -\omega_h\eta + \omega_h J \tag{27}$$

$(J-\eta)$ 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,

$$\dot{\xi} = -\omega_l\xi + \omega_l(J-\eta)a\sin\omega t \tag{28}$$

from which the cost gradient can be obtained as

$$\hat{\mathbf{J}}_{\mathbf{u}} = \frac{a^2}{2}\xi \tag{29}$$

The estimated gradient is then driven to a constant setpoint of zero using an integral controller.

As mentioned above, the dither frequency must be chosen relatively small such that the plant dynamics do no 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. $\omega_i \neq \omega_j$, $2\omega_i \neq \omega_j$, $\omega_i + \omega_j \neq \omega_k$ for any distinct i, j, and k [66, 68].

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 Hunnekens et al. [61]. The last N samples of input $\mathbf{u}$ and cost J measurement data is used to fit a local linear static model of the form,

$$J = \hat{\mathbf{J}}_{\mathbf{u}}^T\mathbf{u} + m \tag{30}$$

where m is the bias. At the current sample time k, let $\mathbf{Y} = [J_k, J_{k-1}, \ldots, J_{k-N+1}]^T$ denote the vector of the past N cost measurements, and $\mathbf{U} = [\mathbf{u}_k, \ldots, \mathbf{u}_{k-N+1}]^T$ be the vector of the past N samples of the input data. A moving

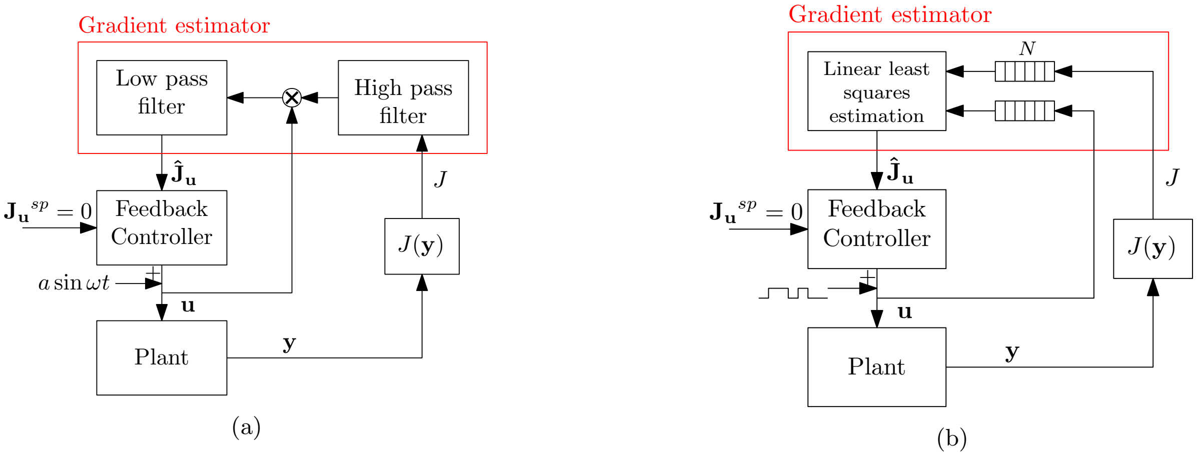


Figure 5: Model-free gradient estimation methods: (a) classical extremum seeking control [59]. (b) Least squares-based extremum seeking control [61]

window of fixed length N is then used to estimate the gradient using a linear least squares estimation

$$\hat{\theta} = \arg\min_{\theta} \|\mathbf{Y} - \Phi^T \theta\|_2^2 \tag{31}$$

to which the analytical solution is given by

$$\hat{\theta} = [\Phi^T \Phi]^{-1} \Phi^T \mathbf{Y} \tag{32}$$

where $\theta = [\hat{\mathbf{J}}_{\mathbf{u}}^T, m]^T$ and $\Phi = [\mathbf{U}, \mathbb{1}]^T$.

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 (32).

Instead of solving a linear least squares problem using (32), 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 in [69, 70].

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 (30) 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. [71] proposed to use an extended Kalman filter (EKF), where the two unknowns $x_1 = \mathbf{J_u}$ and $x_2 = m$ are the states of the Kalman filter. A discrete time model for the two parameters is given by,

$$\begin{bmatrix} \hat{\mathbf{J}}_{\mathbf{u_{k+1}}} \\ m_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{J}}_{\mathbf{u_k}} \\ m_k \end{bmatrix} + \mathbf{w}_k \tag{33}$$

For the system (33) to be observable, two distinct measurements of the input-cost pairs at two different times are required [71, 72]. Consequently, the measurement model is given by,

$$\begin{bmatrix} J_k \\ J_{k-i} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_k & 1 \\ \mathbf{u}_{k-i} & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{J}}_{\mathbf{u_k}} \\ m_k \end{bmatrix} + \mathbf{v}_k \tag{34}$$

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 in [72]. 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 (33) and the measurement model (34).

4.5. Gradients from 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 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 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.

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. Although the use of FFT to estimate the steady-state cost gradient remains to be studied, it has been used in a few application oriented papers by Corti et al. [73] and Beaudoin et al. [74]. In fact, the classical extremum seeking control presented in Section 4.2 can be seen as a special case of the Fourier transform, 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 the 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, and 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

$$\mathcal{J}(\omega_l) = \sum_{k=0}^{N-1} J_0(k) e^{-j\frac{2\pi}{N}l\,k} \qquad \forall l = 0, \dots, N-1 \tag{35}$$

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. 6a.

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

$$\hat{\mathbf{J}}_{u_i} = \widehat{\frac{\partial J}{\partial u_i}} = \frac{2}{a_i} |\mathcal{J}(\omega_i)| \operatorname{sgn}\left[\phi_J(\omega_i) \cdot \phi_{u_i}(\omega_i)\right] \qquad \forall i = 1, \dots, n \tag{36}$$

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 (35). 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 [75].

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 out each other. One such approach was presented by [76], where in the presence of multiple units, the inputs to the units differ by an offset Δ,

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

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

$$\hat{\mathbf{J}}_{\mathbf{u}} = \frac{J_i - J_j}{\Delta} \tag{37}$$

Similar synchronization method for extremum seeking control were also recently presented in [77, 78].

4.7. Model-free gradient estimation using transient measurements

A common trait in all the model-free gradient estimation methods presented in this section is that it requires the assumption that the dynamic plant acts likes 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 [79, 29].

Although this is not a major bottleneck for many mechanical and electro-mechanical applications, for many chemical and biochemical processes, the settling times are often in the range of minutes to hours, thus leading 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 process industry.

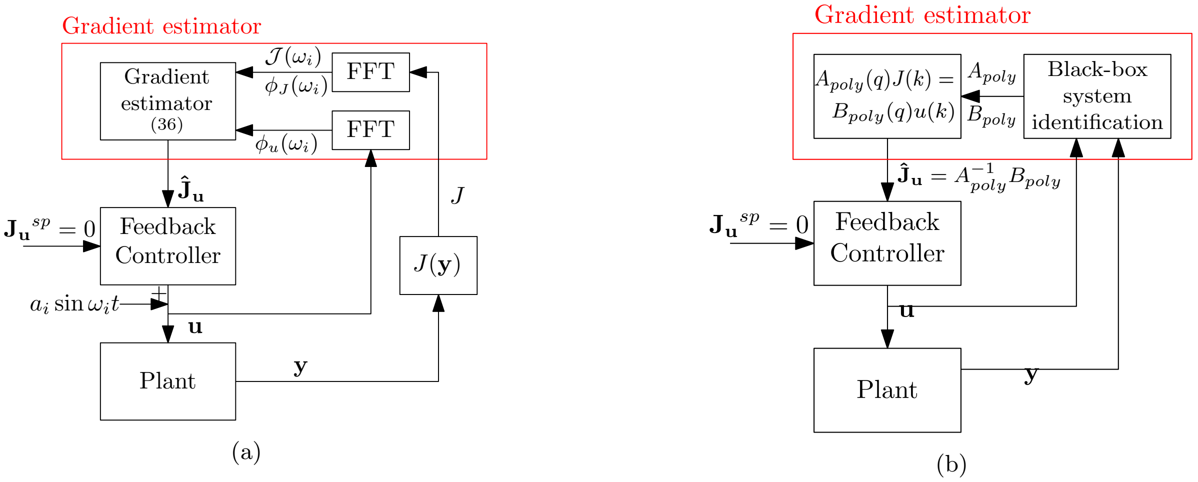


Figure 6: Model-free gradient estimation methods: (a) Gradient estimation using Fast Fourier Transform[74, 73] (b) Gradient estimation from black-box system identification, e.g. ARX [45, 46]

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 [45]. 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 setpoint of zero. This approach was further analyzed by Garcia and Morari [46], 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 [80] presented an empirical strategy for open-loop online optimization using black-box ARX models similar to the one proposed in [45] and [46]. The method proposed in [45, 46] and [80] 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 used to estimate the steady-state gradient

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

$$\begin{aligned} J(t) = &- a_1 J(t-1) - \cdots - a_{n_a} J(t-n_a) \\ &+ b' u(t-1) + \cdots + b'_{n_b} u(t-n_b) + e(t) \end{aligned} \tag{38}$$

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

$$\hat{\theta} = arg\ \min_{\theta} \|\psi - \Phi^{\mathsf{T}}\theta\|_2^2 \tag{39}$$

where ψ, θ and Φ are given by the expressions

$$\psi = J$$

$$\Phi = [-J(t-1)\ \ldots\ -J(t-n_a)\quad u(t-1)\ \ldots\ u(t-n_b)]^{\mathsf{T}}$$

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

Introducing the notation,

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

and

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

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

$$\hat{\mathbf{J}}_{\mathbf{u}} = A^{-1}_{poly} B_{poly} \tag{40}$$

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 system*[1] *as shown below,*

$$\begin{aligned} \dot{x} &= Ax + Bu \\ J &= Cx + Du \end{aligned}$$

results in the expression

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

for the steady-state gradient, which is identical to (23) *in the feedback RTO using transient measurements approach (cf. Section 3.2.2).*

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

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 [81] used a second order Hammerstein model of the form

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

However, with high order ARX, ARMAX models or second order Hammerstein models such as the one used in [81], 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 [1, Chapter 5].

One simple solution to this problem is to simply turn off the gradient estimation once the plant reaches its optimum, as done in [45, 80]. 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, for Hammerstein systems, if the nominal linear dynamics are known, this can be fixed, such that we only estimate the unknown local steady-state effect.

$$J = J_u \underbrace{\left(\frac{b_1 q^{-1} + \cdots + b_{n_b} q^{-n_b}}{1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a}} \right)}_{\text{fixed}} u \tag{41}$$

Fixing the linear dynamic part enables us to effectively use transient measurements, thereby avoiding the steady-state wait time issue. In this case, the least squares problem (39) is solved with

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

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

$$\theta = J_u$$

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 the 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, and hence is not susceptible to plant-model mismatch, whereas the main weakness is the prohibitively slow convergence to the optimum due to the time scale separation. However, 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 then 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, 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. 7a. 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 [40] proposed to combine NCO-tracking with self-optimizing control and the improved performance was demonstrated using a CSTR example from [82]. Similarly, Straus et al. [83] proposed to combine the least square based extremum seeking control with self-optimizing control, and demonstrated the performance improvement on a 3-bed ammonia reactor case example.

Modifier adaptation is another RTO scheme that was proposed in [53] to address the plant-model mismatch. Here, the main idea is to estimate the gradients directly from the measurements, and use this to correct the optimization problem. For the unconstrained optimization problem (3), the modifier adaptation scheme iteratively solves

$$\min_{\mathbf{u}} \quad J_{model}(\mathbf{u}, \mathbf{d}) + \epsilon_J^k + \delta_J^k (\tilde{\mathbf{u}} - \mathbf{u}^k) \tag{42}$$

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.

[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 (3).

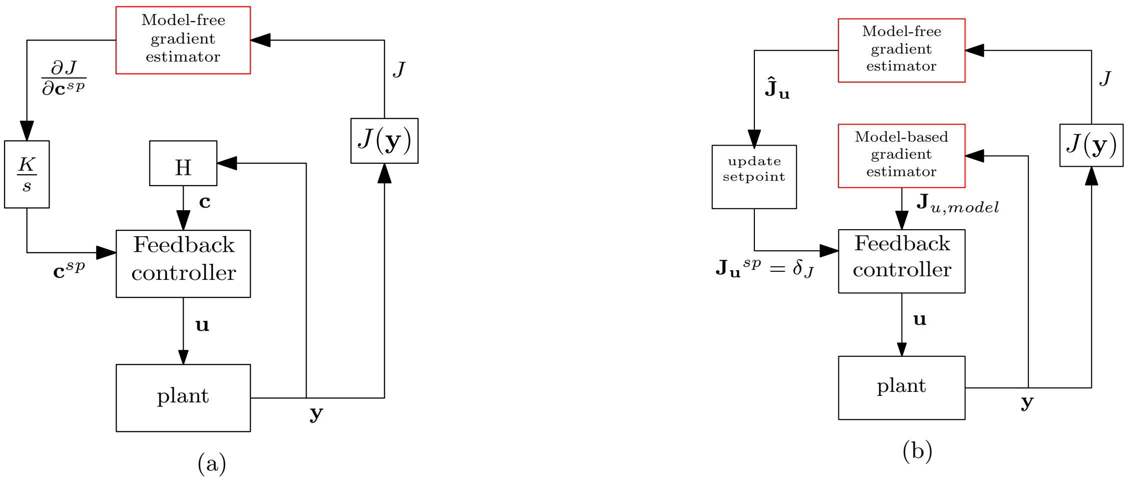


Figure 7: Combination of model-free gradient estimation method with (a) self-optimizing control i.e. models used offline to determine **H** (b) model-based gradient estimation, i.e. models used online

Remark 5 (Modifier Adaptation)**.** *Although MA uses model-free gradient estimation algorithms, it is not a feedback optimizing control approach, since it requires the solution of the numerical optimization problem* (42) *online. To this end, Modifier Adaptation can be seen as a combination of the model-free gradient estimation approach with the traditional steady-state optimization (SRTO), as shown in Fig. 1.*

A similar approach to modifier adaptation, but one that does not require explicitly solving (42) 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. 7b. 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 modelling framework

An alternative approach to using both, the model and the cost measurement is the so-called discrepancy modelling framework. Here the main idea is to approximate the “unmodelled” 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 $\hat{\mathbf{J}}_{\mathbf{u}}$. In other words, the idea here is to train a parametric function approximator $\phi_e(\mathbf{u}, \mathbf{p})$ (e.g. neural networks) that predicts the zeroth order bias correction term ϵ_J, and use the derivative of the trained function $\phi_e(\mathbf{u}, \mathbf{p})$ to account for plant-model mismatch

$$\epsilon_J = J - J_{model} \approx \phi_\epsilon(\mathbf{u}, \mathbf{p}) \tag{43}$$

where $\mathbf{p}$ is the set of parameters that characterizes the function approximator, e.g. weights and biases in a neural network. The plant gradient is then estimated as the sum of the model gradient and the gradient of the function approximator.

$$\hat{\mathbf{J}}_{\mathbf{u}} = \hat{\mathbf{J}}_{\mathbf{u},model} + \frac{\partial \phi_\epsilon}{\partial \mathbf{u}} \tag{44}$$

Such an approach for estimating the plant gradient was recently used in the context of modifier adaptation in [84, 85, 86]. Although these methods were applied in modifier adaptation, the estimated gradients can also be directly driven to a constant setpoint of zero using feedback controllers.

6. Switching between active constraint regions

As mentioned earlier, the set of active constraints changes with changing operating conditions, which requires selecting different set of controlled variables and reconfiguration of the control loops. Feedback optimizing control structure design should also be able to automatically handle changes in the active constraint regions without the need to explicitly solve an optimization problem.

In order to design a feedback optimizing control structure for different operating conditions, we need to first identify the relevant active constraint regions/combinations that are expected. For an optimization problem given by (1) with n_c constraints, we can have a maximum of 2^{n_c} different constraint combinations. However in practice, the relevant active constraint combinations are lesser than this. Using the following guidelines one can eliminate certain constraint combinations.

- 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. most valuable product (avoid product giveaway [22]).

One can also solve the numerical optimization problem offline for expected disturbances to identify the relevant active constraint combinations.

Once the relevant active constraint regions are identified, we decide what to control in each active constraint region given by Theorem 1, and design control loops for the identified controlled variables. 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. Although, such classical advanced control elements have been in use in process control industries for several decades, this has surprisingly received very little attention in the research community.

6.1. MV-MV switching

Optimal operation is often limited by physical MV constraints, e.g. max valve opening. In this case, we may need more than one input to cover the whole steady-state range for one controlled variable, and switch between one MV to another. MV-MV switching can be achieved using three alternative classical control elements, namely,

1. Split-range control [18, 87]

2. Valve position control [87]

3. Individual controllers for each MV with different setpoints [88]

Split-range control. Split range control (SRC), dates back to 1940's [89] 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. 8a, 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 [18, 87] 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. 8b. 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 [51, 88].

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. 8c. 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" presented in [90], where each controller needs to identify when it has saturated. The reader is referred to [18, 87, 90] for more detailed information on using controllers with different setpoints to extend the steady-state operating range.

6.2. 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 [19, 87].

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 $i = 1, \ldots, N$

- 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 [21]. 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})

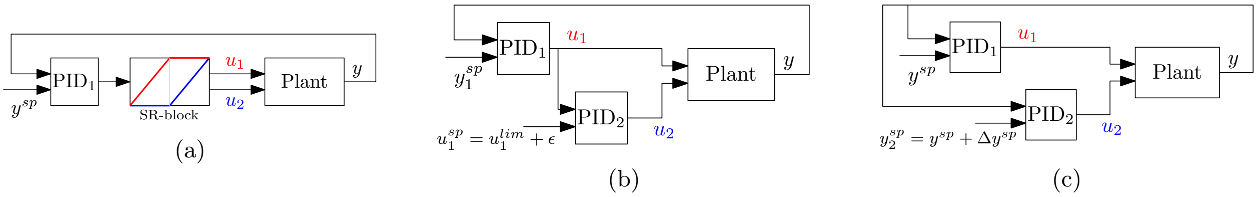


Figure 8: MV-MV switching using (a) split range control (b) Valve position control (c) controllers with different setpoints.

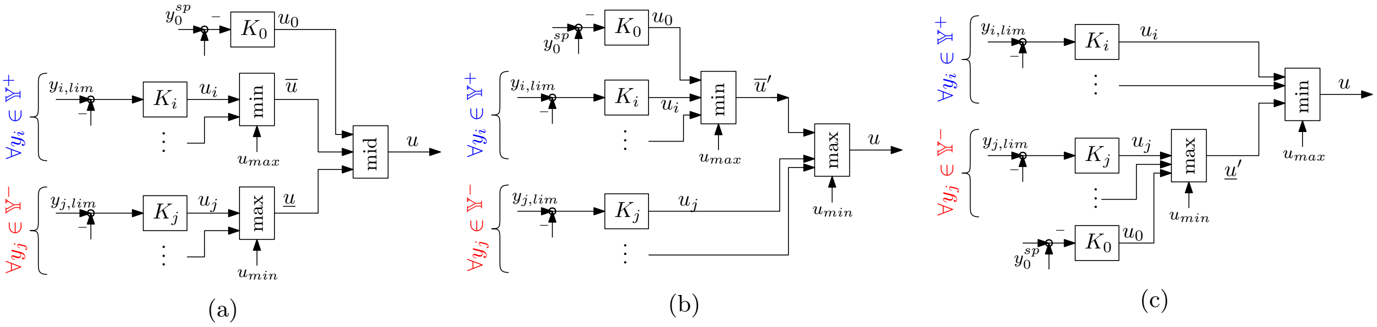


Figure 9: CV-CV switching using (a) min-max-mid selector blocks (b) min-max selector blocks (c) max-min selector blocks.

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, $\overline{u} = \min_{(} \mathbb{Y}^+)u_i$
- 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 $\overline{u} > \underline{u}$ [21], where $\overline{u}$ 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 $u = mid(\underline{u}, u_0, \overline{u})$, 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 $\overline{u}$ and $\underline{u}$, respectively as shown in Fig.9a. Alternatively, this can also be achieved using a min-max or max-min selector block in series as shown in Fig. 9b and Fig. 9c, respectively. The three structures shown in Fig. 9 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 $\overline{u} < \underline{u}$, then the three structures shown in Fig. 9 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. 9c). If not, we can use a min-max structure (Fig. 9b). 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.

To this end, the step-by-step procedure to design suitable switching strategies can be summarized as [87]:

- **Step 1** - Define the control objectives (CV with setpoint control), constraints (CV with limits) 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)
- **Step 3** - Identify relevant active constraint regions.
- **Step 4** - Design the control structure for nominal operating conditions.
- **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 in Sections 6.1-6.2.

Recent works such as [19, 20, 87, 88, 18, 21] 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.

7. Illustrative example - Williams-Otto reactor

In this section, we use a benchmark Williams-Otto reactor example [91] 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 a by-product G, through a series of reactions

$$\begin{aligned}
&A + B \rightarrow C && k_1 = 1.6599 \times 10^6 e^{-6666.7/T_r}\\
&B + C \rightarrow P + E && k_2 = 7.2177 \times 10^8 e^{-8333.3/T_r}\\
&C + P \rightarrow G && k_3 = 2.6745 \times 10^{12} e^{-11111/T_r}
\end{aligned}$$

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 MVs. The feed stream F_A with pure A component is given and 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,

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

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 x_G will always be 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.

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.

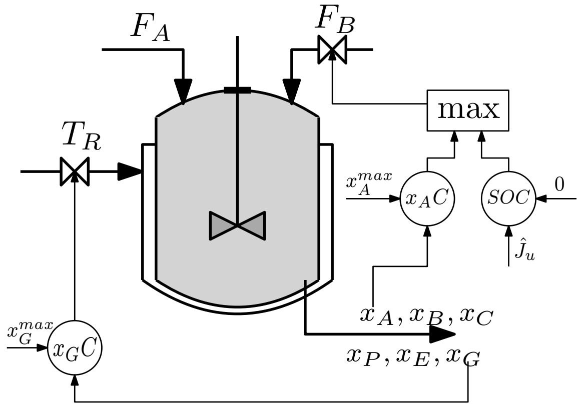


Figure 10: Proposed control structure design for achieving optimal operation of a Williams-Otto reactor using feedback controllers.

Automatic switching between the two regions can be achieved by using a max selector. Fig 10 shows the proposed control structure design for feedback optimizing control of the Williams-Otto reactor.

For the sake of brevity, we will only show simulation results corresponding to region 2 only, where we compare the different model-based and model-free optimization approaches for the unconstrained MV. Additional simulation results from region 1, as well as switching between active constraint regions can be found in [1, Appendix D]. The simulation results shown in this section were performed using `MATLAB v2019b`[3].

7.1. Model-based approaches

For the unconstrained degree of freedom, F_B, we compare the use of two offline methods namely, self-optimizing control (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).

Self-optimizing control and neighboring extremal control were designed around the nominal operating conditions of $F_A = 1.3$ kg/s. Fig. 11 shows the simulation results using the different model-based approaches. The true 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 converge to the optimum. At time $t = 120$ min, 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 however drives the process to its optimum, since these are based on local linear approximation around the

[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`

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. 12 shows the simulation results 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 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. 11.

8. Discussion

In this section, we review 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 modelling 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 [7]. Therefore, in order to use transient measurements, it is recommended to use dynamic models for online model-based methods [44].

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 used for monitoring the process operations.

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 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 function to be directly measurable. 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

$$J = p_Q Q + p_F F - p_1 P_1 - p_2 P_2 + \dots \qquad (45)$$

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 apply any of the model-free methods.

8.3. Accuracy

The oddline 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.

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 [52], Chachuat et al. [3] to name a few. Consequently, in online model-based methods, where the cost gradient is estimated using the model, plant-model mismatch implies that the process is driven to the model-optimum, which may not correspond to the 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. However, the caveat here is 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 [79]. 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

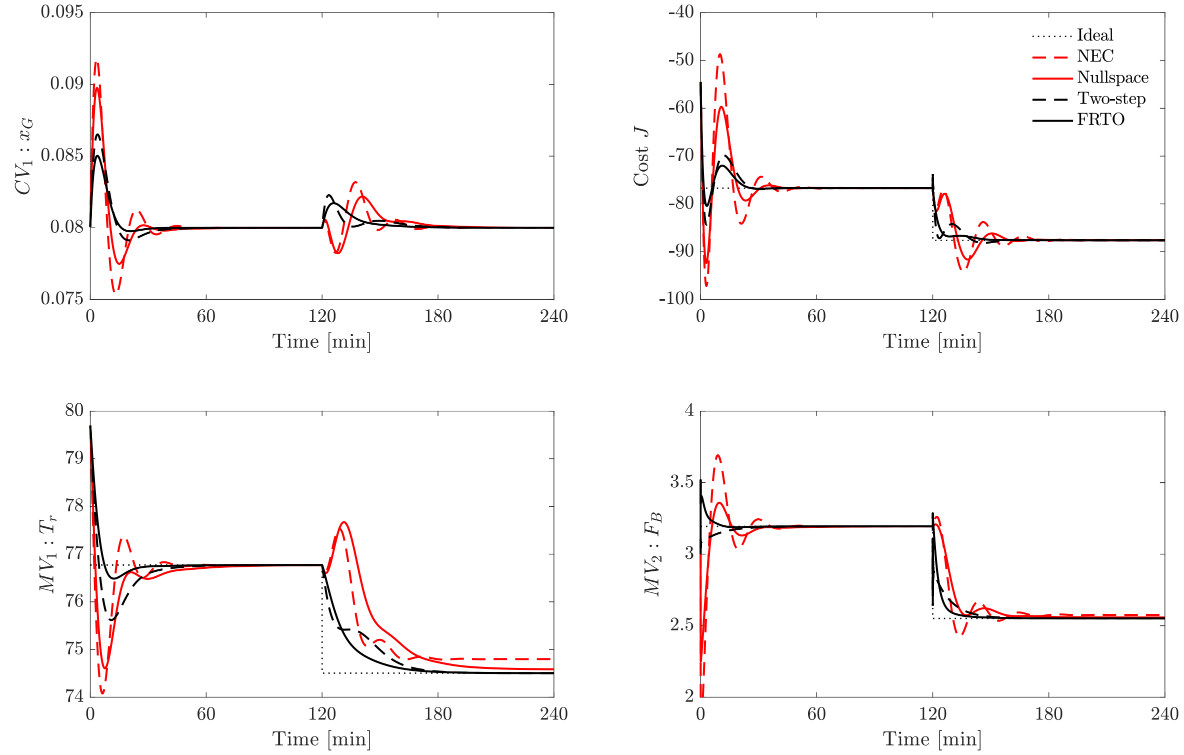


Figure 11: Comparison of four different model-based feedback optimizing control approaches on the benchmark Williams-Otto reactor example.

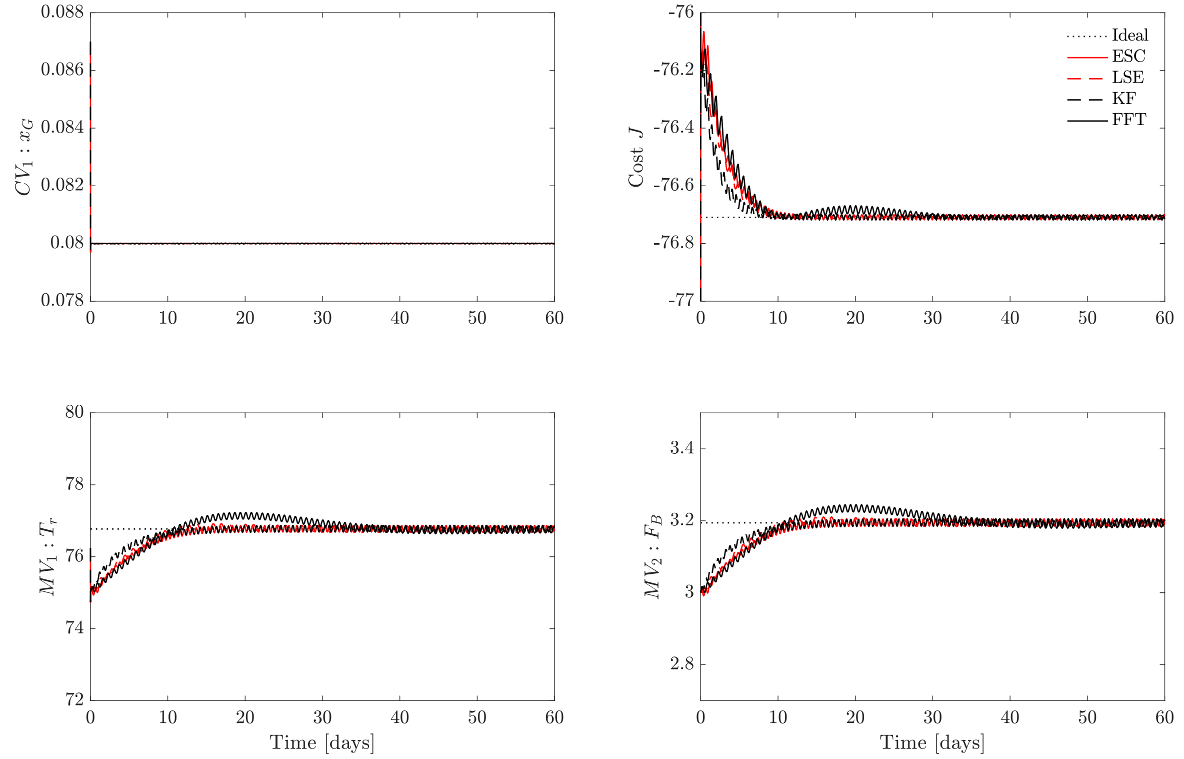


Figure 12: Comparison of four different model-free feedback optimizing control approaches on the benchmark Williams-Otto reactor example.

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 [92]. 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 changes observed in the cost measurement are induced by the perturbations in 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, [92]. 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 in [93]. 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 the 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 [61]or diminishing dither approaches [94] 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 chemical process industries.

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. Methods 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 [95] 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 (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 makes the implementation rather cumbersome.

8.8. Handling nonconvexity

So far, we assumed that the stationary point $\mathbf{u}^*$ of the unconstrained optimization problem (3) 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 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. An acquisition functions defined using the GP surrogate model is then used to update the next control input [96, 97, 98, 99]. The acquisition functions are chosen to trade-off exploitation vs. exploration in order to find the global minimum. Bayesian optimization can be seen as an alternative to the model-free approaches that drives the processes to its global minimum. 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. Consequently, this requires several CV-CV, MV-MV, and CV-MV switching elements, leading to a complicated and messy control structure design. This makes maintenance and monitoring of the control loops more challenging, and perhaps one would then be better off with the traditional RTO framework.

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 the 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.

Distributed Feedback-based RTO. The feedback controllers are also 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. [11].

To address some of the scalability issues, 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. A distributed feedback-based RTO (DFRTO) scheme based on the Lagrangian decomposition approach was recently proposed by Krishnamoorthy [100], where the self-optimizing controlled variable for local subsystems are given as a function of the Lagrange multiplier of the coupling constraints. A central coordinator updates the Lagrange multipliers thereby leading to overall system-wide optimal operation, as shown in Fig. 13. 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 [101, Section 7.3], where as in the latter case, the optimization problem is formulated as a consensus problem [101, Section 7.1]. The interested reader is referred to [100] for more detailed description and analysis of the distributed feedback-based RTO scheme.

9. Summary

In this paper we provided a comprehensive overview of the different approaches that aims to achieve optimal operation at steady-state without the need to solve numerical optimization problems.

We showed that the economic objectives can be translated into control objectives by controlling (in this order):

1. the active constraints
2. self-optimizing variables for the unconstrained degrees of freedom

The choice of the self-optimizing controlled variable is not obvious and there have been several approaches that have been developed in literature. For example, one could

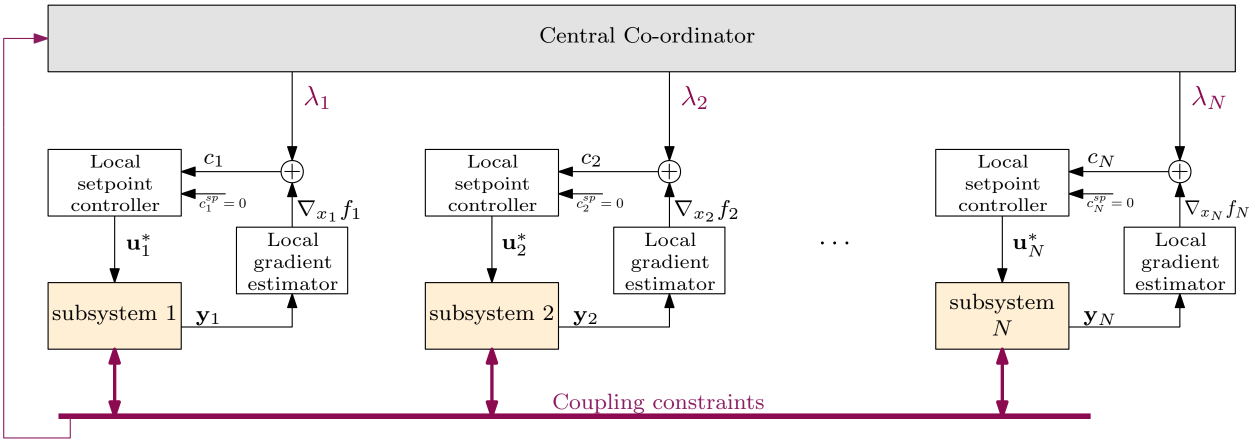


Figure 13: A distributed feedback-based RTO framework using Lagrangian decomposition. The large-scale process is decomposed into N smaller subsystems, each optimized locally using feedback controllers. The central coordinator ensures system-wide optimal operation.

use the process models offline to choose a single measurement or a linear combination of measurements as self-optimizing variables, as reviewed in Section 3.1. Alternatively, one can use the steady-state cost gradient as self-optimizing variable and drive them to zero in order to satisfy the necessary condition of optimality. Methods that use process models online to estimate the cost gradient were reviewed in Section 3.2, and methods that estimates the cost gradient directly from the cost measurement in a model-free fashion were reviewed in Section 4. We also showed that the the different approaches are not contradictory, but complementary. Works that exploit the synergistic combination of the different methods were reviewed in Section 5. The set of active constraints also changes with changing operating conditions. This requires a change in the controlled variables and reconfiguration of the control loops. The use of classical advanced control elements to handle changes in the active constraint set were reviewed in Section 6. A handful of the different methods reviewed were also implemented on a benchmark Williams-Otto reactor case example to provide a comparitive study of the different approaches in Section 7. Finally, a multifaceted discussion was provided in Section 8 to compare and contrast the key features of the different paradigms that constitutes the scope of *feedback optimizing control*.

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 EFCE Excellence in CAPE PhD Thesis Award.

The author gratefully acknowledges financial support from the Research Council of Norway through the IKT-PLUSS program (project number 299585) and SFI SUBPRO (project number 237893).

References

[1] D. Krishnamoorthy, Novel Approaches to Online Process Optimization under Uncertainty, Ph.D. thesis, Norwegian University of Science Techology, 2019.

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

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

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

[5] M. M. Câmara, A. D. Quelhas, J. C. Pinto, Performance Evaluation of Real Industrial RTO Systems, Processes 4 (4) (2016) 44.

[6] D. Krishnamoorthy, B. Foss, S. Skogestad, Steady-state real-time optimization using transient measurements, Computers & Chemical Engineering 115 (2018) 34–45.

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

[8] D. Shook, Best practices improve control system performance, Oil & gas journal 104 (38).

[9] M. G. Forbes, R. S. Patwardhan, H. Hamadah, R. B. Gopaluni, Model predictive control in industry: Challenges and opportunities, IFAC-PapersOnLine 48 (8) (2015) 531–538.

[10] K. Forsman, Implementation of advanced control in the process industry without the use of MPC, IFAC-PapersOnLine 49 (7) (2016) 514–519.

[11] M. Morari, Y. Arkun, G. Stephanopoulos, Studies in the synthesis of control structures for chemical processes: Part I: Formulation of the problem. Process decomposition and the classification of the control tasks. Analysis of the optimizing control structures, AIChE Journal 26 (2) (1980) 220–232.

[12] B. Srinivasan, D. Bonvin, 110th anniversary: a feature-based analysis of static real-time optimization schemes, Industrial & Engineering Chemistry Research 58 (31) (2019) 14227–14238.

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

[14] A. Maarleveld, J. Rijnsdorp, Constraint control on distillation columns, Automatica 6 (1) (1970) 51–58.

[15] M. G. Jacobsen, S. Skogestad, Active constraint regions for optimal operation of distillation columns, Industrial & Engineering Chemistry Research 51 (7) (2012) 2963–2973.

[16] Y. Arkun, G. Stephanopoulos, Studies in the synthesis of control structures for chemical processes: Part IV. Design of

steady-state optimizing control structures for chemical process units, AIChE Journal 26 (6) (1980) 975–991.

[17] W. R. Fisher, M. F. Doherty, J. M. Douglas, The interface between design and control. 3. Selecting a set of controlled variables, Industrial & engineering chemistry research 27 (4) (1988) 611–615.

[18] A. Reyes-Lúa, C. Zotică, T. Das, D. Krishnamoorthy, S. Skogestad, Changing between Active Constraint Regions for Optimal Operation: Classical Advanced Control versus Model Predictive Control, in: Computer Aided Chemical Engineering, vol. 43, Elsevier, 1015–1020, 2018.

[19] D. Krishnamoorthy, S. Skogestad, Online Process Optimization with Active Constraint Set Changes using Simple Control Structures, Industrial & Engineering Chemistry Research .

[20] D. Krishnamoorthy, K. Fjalestad, S. Skogestad, Optimal operation of oil and gas production using simple feedback control structures, Control Engineering Practice xx (2019) xxx–xxx.

[21] D. Krishnamoorthy, S. Skogestad, Systematic design of active constraint switching using selectors, Computers & Chemical Engineering 143 (2020) 107106.

[22] M. S. Govatsmark, S. Skogestad, Selection of controlled variables and robust setpoints, Industrial & engineering chemistry research 44 (7) (2005) 2207–2217.

[23] V. Alstad, S. Skogestad, Null space method for selecting optimal measurement combinations as controlled variables, Industrial & engineering chemistry research 46 (3) (2007) 846–853.

[24] I. J. Halvorsen, S. Skogestad, J. C. Morud, V. Alstad, Optimal selection of controlled variables, Industrial & Engineering Chemistry Research 42 (14) (2003) 3273–3284.

[25] G. François, B. Srinivasan, D. Bonvin, Comparison of six implicit real-time optimization schemes, Journal Européen des Systèmes Automatisés 46 (EPFL-ARTICLE-170545) (2012) 291–305.

[26] G. François, B. Srinivasan, D. Bonvin, Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty, Journal of Process Control 15 (6) (2005) 701–712.

[27] D. Krishnamoorthy, S. Skogestad, Linear Gradient Combination as self-optimizing variables, Computer Aided Chemical Engineering In-Press.

[28] K. B. Ariyur, M. Krstic, Real-time optimization by extremum-seeking control, John Wiley & Sons, 2003.

[29] Y. Tan, W. Moase, C. Manzie, D. Nešić, I. Mareels, Extremum seeking from 1922 to 2010, in: Control Conference (CCC), 2010 29th Chinese, IEEE, 14–26, 2010.

[30] A. Hauswirth, A. Zanardi, S. Bolognani, F. Dörfler, G. Hug, Online optimization in closed loop on the power flow manifold, in: 2017 IEEE Manchester PowerTech, IEEE, 1–6, 2017.

[31] V. Alstad, S. Skogestad, E. S. Hori, Optimal measurement combinations as controlled variables, Journal of Process Control 19 (1) (2009) 138–148.

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

[33] J. Jäschke, Y. Cao, V. Kariwala, Self-optimizing control–A survey, Annual Reviews in Control .

[34] R. Yelchuru, S. Skogestad, Convex formulations for optimal selection of controlled variables and measurements using mixed integer quadratic programming, Journal of Process control 22 (6) (2012) 995–1007.

[35] V. Alstad, Studies on selection of controlled variables, PhD thesis, 2005.

[36] L. Ye, Y. Cao, X. Yuan, Global approximation of self-optimizing controlled variables with average loss minimization, Industrial & Engineering Chemistry Research 54 (48) (2015) 12040–12053.

[37] L. Ye, Y. Cao, S. Skogestad, Global self-optimizing control for uncertain constrained process systems, IFAC-PapersOnLine 50 (1) (2017) 4672–4677.

[38] L. Ye, Y. Cao, Y. Li, Z. Song, Approximating necessary conditions of optimality as controlled variables, Industrial & Engineering Chemistry Research 52 (2) (2012) 798–808.

[39] G. François, B. Srinivasan, D. Bonvin, Equivalence between neighboring-extremal control and self-optimizing control for the steady-state optimization of dynamical systems, Industrial & Engineering Chemistry Research 53 (18) (2014) 7470–7478.

[40] J. Jäschke, S. Skogestad, NCO tracking and self-optimizing control in the context of real-time optimization, Journal of Process Control 21 (10) (2011) 1407–1416.

[41] S. Gros, B. Srinivasan, D. Bonvin, Optimizing control based on output feedback, Computers & Chemical Engineering 33 (1) (2009) 191–198.

[42] V. de Oliveira, J. Jaschke, S. Skogestad, Neighbouring-Extremal Control for Steady-State Optimization Using Noisy Measurements, IFAC-PapersOnLine 48 (8) (2015) 698–703.

[43] V. Adetola, M. Guay, Parameter convergence in adaptive extremum-seeking control, Automatica 43 (1) (2007) 105–110.

[44] D. Krishnamoorthy, E. Jahanshahi, S. Skogestad, A Feedback Real Time Optimization Strategy using a Novel Steady-state Gradient Estimate and Transient Measurements, Industrial and Engineering Chemistry Research 58 (2019) 207–216.

[45] W. Bamberger, R. Isermann, Adaptive on-line steady-state optimization of slow dynamic processes, Automatica 14 (3) (1978) 223–230.

[46] C. E. Garcia, M. Morari, Optimal operation of integrated processing systems. Part I: Open-loop on-line optimizing control, AIChE Journal 27 (6) (1981) 960–968.

[47] H. Bonnowitz, J. Straus, D. Krishnamoorthy, E. Jahanshahi, S. Skogestad, Control of the Steady-State Gradient of an Ammonia Reactor using Transient Measurements, Computer-Aided Chemical Engineering 43 (2018) 1111–1116.

[48] D. Krishnamoorthy, E. Jahanshahi, S. Skogestad, Gas-lift Optimization by Controlling Marginal Gas-Oil Ratio using Transient Measurements, IFAC-PapersOnLine 51 (8) (2018) 19–24.

[49] D. Krishnamoorthy, E. Jahanshahi, S. Skogestad, Control of the Steady-State Gradient of an evaporator process using Transient Measurements, PSE Asia (In-press) .

[50] V. Kumar, N. Kaistha, Hill-Climbing for Plantwide Control to Economic Optimum, Industrial & Engineering Chemistry Research 53 (42) (2014) 16465–16475.

[51] F. G. Shinskey, Process control systems: application, design, and tuning, vol. 4, McGraw-Hill New York, 1996.

[52] P. D. Roberts, T. Williams, On an algorithm for combined system optimisation and parameter estimation, Automatica 17 (1) (1981) 199–209.

[53] A. Marchetti, B. Chachuat, D. Bonvin, Modifier-adaptation methodology for real-time optimization, Industrial & engineering chemistry research 48 (13) (2009) 6022–6033.

[54] P. Roberts, Broyden derivative approximation in ISOPE optimising and optimal control algorithms, IFAC Proceedings Volumes 33 (16) (2000) 293–298.

[55] W. Gao, S. Engell, Iterative set-point optimization of batch chromatography, Computers & Chemical Engineering 29 (6) (2005) 1401–1409.

[56] E. A. Rodger, B. Chachuat, Design methodology of modifier adaptation for on-line optimization of uncertain processes, IFAC Proceedings Volumes 44 (1) (2011) 4113–4118.

[57] M. Brdyś, P. Tatjewski, An algorithm for steady-state optimizing dual control of uncertain plants, in: New Trends in Design of Control Systems 1994, Elsevier, 215–220, 1995.

[58] C. S. Draper, Y. T. Li, Principles of optimalizing control systems and an application to the internal combustion engine, American Society of Mechanical Engineers, 1951.

[59] M. Krstić, H.-H. Wang, Stability of extremum seeking feedback for general nonlinear dynamic systems, Automatica 36 (4) (2000) 595–601.

[60] N. J. Killingsworth, M. Krstic, PID tuning using extremum seeking: online, model-free performance optimization, IEEE control systems magazine 26 (1) (2006) 70–79.

[61] B. Hunnekens, M. Haring, N. van de Wouw, H. Nijmeijer, A dither-free extremum-seeking control approach using 1st-order least-squares fits for gradient estimation, in: 53rd IEEE Con-

ference on Decision and Control, IEEE, 2679–2684, 2014.

[62] L. Fu, Ü. Özgüner, Extremum seeking with sliding mode gradient estimation and asymptotic regulation for a class of nonlinear systems, Automatica 47 (12) (2011) 2595–2603.

[63] Y. Pan, Ü. Özgüner, T. Acarman, Stability and performance improvement of extremum seeking control with sliding mode, International Journal of Control 76 (9-10) (2003) 968–985.

[64] O. Trollberg, E. W. Jacobsen, Greedy Extremum Seeking Control with Applications to Biochemical Processes, IFAC-PapersOnLine (DYCOPS-CAB) 49 (7) (2016) 109–114.

[65] J. Choi, M. Krstic, K. Ariyur, J. Lee, Extremum seeking control for discrete-time systems, IEEE Transactions on automatic control 47 (2) (2002) 318–323.

[66] A. Ghaffari, M. Krstić, D. Nešić, Multivariable Newton-based extremum seeking, Automatica 48 (8) (2012) 1759–1767.

[67] H. Dürr, M. Stanković, C. Ebenbauer, K. Johansson, Lie bracket approximation of extremum seeking systems, Automatica 49 (6) (2013) 1538–1552.

[68] A. Ghaffari, S. Seshagiri, M. Krstić, Multivariable maximum power point tracking for photovoltaic micro-converters using extremum seeking, Control Engineering Practice 35 (2015) 83–91.

[69] M. Chioua, B. Srinivasan, M. Guay, M. Perrier, Performance Improvement of Extremum Seeking Control using Recursive Least Square Estimation with Forgetting Factor, IFAC-PapersOnLine 49 (7) (2016) 424–429.

[70] L. Dewasme, B. Srinivasan, M. Perrier, A. V. Wouwer, Extremum-seeking algorithm design for fed-batch cultures of microorganisms with overflow metabolism, Journal of Process Control 21 (7) (2011) 1092–1104.

[71] L. Henning, R. Becker, G. Feuerbach, R. Muminovic, R. King, A. Brunn, W. Nitsche, Extensions of adaptive slope-seeking for active flow control, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 222 (5) (2008) 309–322.

[72] G. Gelbert, J. P. Moeck, C. O. Paschereit, R. King, Advanced algorithms for gradient estimation in one-and two-parameter extremum seeking controllers, Journal of Process Control 22 (4) (2012) 700–709.

[73] E. Corti, C. Forte, G. Mancini, D. Moro, Automatic combustion phase calibration with extremum seeking approach, Journal of Engineering for Gas Turbines and Power 136 (9).

[74] J.-F. Beaudoin, O. Cadot, J.-L. Aider, J. E. Wesfreid, Bluff-body drag reduction by extremum-seeking control, Journal of fluids and structures 22 (6-7) (2006) 973–978.

[75] D. Krishnamoorthy, A Multivariable Extremum Seeking Control Approach using Fast Fourier Transform for Gradient Estimation, URL `http://folk.ntnu.no/dineshk/Papers/ESCFFT.pdf`, 2020.

[76] B. Srinivasan, Real-time optimization of dynamic systems using multiple units, International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal 17 (13) (2007) 1183–1193.

[77] A. Pavlov, M. Haring, K. Fjalestad, Practical extremum-seeking control for gas-lifted oil production, in: Decision and Control (CDC), 2017 IEEE 56th Annual Conference on, IEEE, 2102–2107, 2017.

[78] T. L. Silva, A. Pavlov, Dither signal optimization for multi-agent extremum seeking control, in: 2020 European Control Conference (ECC), IEEE, 1230–1237, 2020.

[79] M. Krstić, Performance improvement and limitations in extremum seeking control, Systems & Control Letters 39 (5) (2000) 313–326.

[80] R. McFarlane, D. Bacon, Empirical strategies for open-loop on-line optimization, The Canadian Journal of Chemical Engineering 67 (4) (1989) 665–677.

[81] M. P. Golden, B. E. Ydstie, Adaptive extremum control using approximate process models, AIChE journal 35 (7) (1989) 1157–1169.

[82] C. G. Economou, M. Morari, B. O. Palsson, Internal model control: Extension to nonlinear system, Industrial & Engineering Chemistry Process Design and Development 25 (2) (1986) 403–411.

[83] J. Straus, D. Krishnamoorthy, S. Skogestad, Combining self-optimizing control and extremum seeking control - Applied to ammonia reacotr case study, Journal of Process control 78 (2019) 78–87.

[84] W. Gao, S. Wenzel, S. Engell, A reliable modifier-adaptation strategy for real-time optimization, Computers & chemical engineering 91 (2016) 318–328.

[85] J. O. Matias, J. Jäschke, Using a neural network for estimating plant gradients in real-time optimization with modifier adaptation .

[86] E. A. del Rio Chanona, J. E. Alves Graciano, E. Bradford, B. Chachuat, Modifier-Adaptation Schemes Employing Gaussian Processes and Trust Regions for Real-Time Optimization .

[87] A. Reyes-Lúa, S. Skogestad, Systematic design of active constraint switching using classical advanced control structures, Industrial & Engineering Chemistry Research .

[88] A. Reyes-Lúa, S. Skogestad, Multiple-Input Single-Output Control for Extending the Steady-State Operating Range—Use of Controllers with Different Setpoints, Processes 7 (12) (2019) 941.

[89] D. P. Eckman, Principles of industrial process control .

[90] A. Reyes-Lúa, S. Skogestad, Multi-input single-output control for extending the operating range: Generalized split range control using the baton strategy, Journal of Process Control 91 (2020) 1–11.

[91] T. J. Williams, R. E. Otto, A generalized chemical processing model for the investigation of computer control, Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics 79 (5) (1960) 458–473.

[92] D. Krishnamoorthy, A. Pavlov, Q. Li, Robust Extremum Seeking Control with application to Gas Lifted Oil Wells, IFAC-PapersOnLine 49 (13) (2016) 205–210.

[93] S. Marinkov, B. de Jager, M. Steinbuch, Extremum seeking control with data-based disturbance feedforward, in: 2014 American Control Conference, ISSN 0743-1619, 3627–3632, doi:10.1109/ACC.2014.6858832, 2014.

[94] L. Wang, S. Chen, K. Ma, On stability and application of extremum seeking control without steady-state oscillation, Automatica 68 (2016) 18–26.

[95] S. Skogestad, Simple analytic rules for model reduction and PID controller tuning, Journal of process control 13 (4) (2003) 291–309.

[96] J. Mockus, Bayesian approach to global optimization: theory and applications, vol. 37, Springer Science & Business Media, 2012.

[97] D. R. Jones, M. Schonlau, W. J. Welch, Efficient global optimization of expensive black-box functions, Journal of Global optimization 13 (4) (1998) 455–492.

[98] P. I. Frazier, A tutorial on bayesian optimization, arXiv preprint arXiv:1807.02811 .

[99] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, N. De Freitas, Taking the human out of the loop: A review of Bayesian optimization, Proceedings of the IEEE 104 (1) (2015) 148–175.

[100] D. Krishnamoorthy, A distributed feedback-based online process optimization framework for optimal resource sharing, Journal of Process Control 97 (2021) 72–83.

[101] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends® in Machine Learning 3 (1) (2011) 1–122.