



On combining self-optimizing control and extremum-seeking control – Applied to an ammonia reactor case study

Julian Straus, Dinesh Krishnamoorthy*, Sigurd Skogestad

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

ARTICLE INFO

Article history:

Received 20 February 2018

Received in revised form 9 October 2018

Accepted 23 January 2019

Available online 28 April 2019

Keywords:

Plant-wide control

Self-optimizing control

Extremum-seeking control

Real-time optimization

Measurement-based optimization

Chemical reactor

ABSTRACT

In this paper, we combine self-optimizing control and extremum-seeking control in the context of real-time optimization. Self-optimizing control is based on controlling a single measurement or measurement combination whose optimal setpoint is insensitive to the expected disturbances. This gives a fast reaction to disturbances, but the optimal setpoint may change over time because of larger deviations from the nominal optimal point or due to unmodelled disturbances. Extremum-seeking control, on the other hand, belongs to a class of model-free methods that optimizes the system based on directly measuring the cost. However, it converges slowly. In this paper, we propose to use an extremum-seeking controller to provide setpoint adjustments to the self-optimizing controller in order to improve the convergence rate. The key idea here is to keep the process near the optimal region on a fast timescale using self-optimizing control and fine tune the setpoint on a longer timescale using extremum-seeking control. We verify the proposed method using an ammonia reactor case study.

© 2019 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

We want to optimize the operation of a process by minimizing a cost function J subject to process constraints. Often a real-time optimizer (RTO) or a model-based controller such as economic model predictive control (E-MPC) is proposed. Both approaches use a process model together with measurements from the process to determine the optimal operating conditions by solving a numerical optimization problem online. However, this is often computationally expensive, even with today's computing power. In addition, model-based controllers may be sensitive to unmodelled disturbances and structural mismatch, but this is not the main issue in this paper.

Several different measurement-based alternatives to RTO have been developed in the past two decades that avoid the online optimization by simply transforming the optimization problem into a feedback control problem. These are sometimes classified under direct input adaptation-based methods. Such methods are computationally cheap compared to numerical optimization-based methods, since optimization is done via feedback. Extremum-seeking control, necessary conditions of optimality (NCO) tracking,

and self-optimizing control belong to such methods. A classification of the different methods available to handle uncertainty in real-time optimization can be found in [1] and [2].

Self-optimizing control deals with using the model offline to find an appropriate set of controlled variables. The objective is to translate economic objectives into control objectives [3]. This may eliminate the need for an expensive online optimizer to re-optimize when disturbances occur or at least the re-optimization may be performed less frequently. One of the main advantages of self-optimizing control is that it moves the slower economic optimization into the faster control layer. It can also handle unmeasured disturbances as long as they have been modelled. This is achieved by selecting an optimal combination of measurements, which when held at a constant setpoint, gives acceptable loss in the presence of varying disturbances. However, unmodelled disturbances will generally result in a loss. Note that the ideal self-optimizing control variable is the steady-state gradient from the inputs to the cost, \mathbf{J}_u , as noted in [3].

An alternative feedback strategy, is to use the measurements directly to estimate the steady-state gradient. The estimated gradient can then be driven to zero in order to optimize the operation. Extremum-seeking control, NCO-tracking methods and hill-climbing controllers (see e.g. [4]) belong to this class of model-free optimization methods. The concept of extremum-seeking control can be dated as far back as 1922 and since about 2000

* Corresponding author.

E-mail addresses: dinesh.krishnamoorthy@ntnu.no (D. Krishnamoorthy), skoge@ntnu.no (S. Skogestad).

extremum-seeking control has received much attention and several developments have been made [5].

Model-free extremum-seeking control methods solely rely on cost measurements based on exciting the process. Consequently, a clear advantage is that they are not affected by plant-model mismatch and unknown disturbances. However, since the steady-state gradient from the inputs to the cost is estimated from the cost measurements, it requires timescale separation between the process dynamics, excitation signal and convergence rate [6]. Thus, unless the process dynamics are very fast, the convergence to the optimal point is usually prohibitively slow [7,8]. The use of transient measurements may result in erroneous steady-state gradient estimation. In addition, external disturbances may affect the gradient estimation, and lead to deviations from the optimal operation [9]. Disturbance handling in extremum-seeking control may be improved using measured disturbances as proposed in [9,10], and [11].

The self-optimizing control and extremum-seeking control methods have been developed relatively independently since 2000. The authors in [12] successfully combined self-optimizing control with NCO tracking in a hierarchical structure and demonstrated that the measurement-based optimization techniques and the model-based self-optimizing concepts are complementary. However, the gradient was estimated using finite differences which gave relatively poor NCO tracking. The authors suggested that more advanced gradient estimation and input adaptation methods may give a better overall performance as a future research direction. The use of extremum-seeking control on top of self-optimizing control was briefly discussed in [13] using the classical extremum-seeking method. However, the authors only considered measured disturbances for a single-input single output system. Additionally, based on the simulation results presented, the authors in [13] did not consider a clear time scale separation between the extremum-seeking and self-optimizing controllers.

In this paper we extend the work from [12] and [13] and provide a detailed description on the combination of the model-free extremum-seeking controller with model-based self-optimizing control. We propose to use an improved gradient estimation method using least squares estimation and provide a framework for a multivariable system. We then apply the proposed control structure to a multivariable ammonia reactor case study with 3 control inputs and consider both unmeasured and unmodelled disturbances. We compare the performance of the proposed method with just the self-optimizing control and just the extremum-seeking control implemented independently and demonstrate clear performance improvements due to the time-scale separation between the extremum-seeking and self-optimizing controllers.

The paper is structured as follows. Section 2 briefly describes the ideas of self-optimizing control as described in [3] and [14] and extremum-seeking control as described in [15] and [16]. Section 3 describes the framework in which we combine the two methods in a hierarchical structure. We then exemplify the proposed method using an ammonia synthesis reactor in Section 4. The results are discussed in Section 5 before concluding the paper in Section 6.

2. Background

Consider a process where the steady-state optimal operation of the process can be formulated as,

$$\begin{aligned} \min_{\mathbf{u}} \quad & J(\mathbf{u}, \mathbf{d}) \\ \text{s.t.} \quad & \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d}) = \mathbf{0} \end{aligned} \quad (1)$$

where $J \in \mathbb{R}$ is a scalar cost, $\mathbf{x} \in \mathbb{R}^{n_x}$ denote the vector of state variables, $\mathbf{u} \in \mathbb{R}^{n_u}$ denotes the vector of manipulated variables and

$\mathbf{d} \in \mathbb{R}^{n_d}$ denotes the vector of disturbances and \mathbf{f} represents the n_x independent steady-state model equations. Note that we have no inequality constraints. More precisely, we assume that any active constraints are satisfied and the n_u manipulated variables \mathbf{u} are the remaining unconstrained degrees of freedom which are available for optimization [3].

Assumption 1. There exists a smooth function $l : \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \rightarrow \mathbb{R}^{n_x}$ such that, $\mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d}) = \mathbf{0}$ if and only if $\mathbf{x} = \mathbf{l}(\mathbf{u}, \mathbf{d})$.

We use the steady-state model equations $\mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d}) = \mathbf{0}$, enforced as equality constraints to formally eliminate the states. The steady-state cost can then be expressed just in terms of the inputs \mathbf{u} and disturbances \mathbf{d} ,

$$J_{ss}(\mathbf{u}, \mathbf{d}) = J(\mathbf{l}(\mathbf{u}, \mathbf{d}), \mathbf{u}, \mathbf{d}) \quad (2)$$

The steady-state version of the optimization problem (1) is then equivalent to

$$\min_{\mathbf{u}} J_{ss}(\mathbf{u}, \mathbf{d}) \quad (3)$$

This says that the input \mathbf{u} should be manipulated to optimize the steady state performance for any given disturbance \mathbf{d} . We make the following additional assumption:

Assumption 2. There exists $\mathbf{u} = \mathbf{u}^{opt}(\mathbf{d})$

such that,

$$\frac{\partial J_{ss}}{\partial \mathbf{u}}(\mathbf{u}^{opt}, \mathbf{d}) = \mathbf{J}_{\mathbf{u}}^{opt}(\mathbf{d}) = \mathbf{0} \quad (4)$$

$$\frac{\partial^2 J_{ss}}{\partial \mathbf{u}^2}(\mathbf{u}^{opt}, \mathbf{d}) = \mathbf{J}_{\mathbf{uu}}^{opt}(\mathbf{d}) > \mathbf{0} \quad (5)$$

Assumption 2 ensures that the problem is convex with a unique minimum at $\mathbf{u} = \mathbf{u}^{opt}(\mathbf{d})$ for a given disturbance \mathbf{d} .

2.1. Self-optimizing control

Self-optimizing control is a strategy of selecting an optimal measurement combination \mathbf{c} as controlled variables, such that the impact of known but unmeasured disturbances \mathbf{d} on the optimal operation is minimized. This is achieved by using the system model offline to compute an optimal measurement combination. The ideal self-optimizing variable would be the gradient $\mathbf{J}_{\mathbf{u}}$ which should be controlled to a constant setpoint of $\mathbf{0}$. However, in most applications, the gradient cannot be measured. An alternative is to identify a controlled variable $\mathbf{c} \in \mathbb{R}^{n_c}$ (with $n_c = n_u$) as a function of the available measurements $\mathbf{y} \in \mathbb{R}^{n_y}$. The simplest approach to select a linear combination of measurements is given by,

$$\mathbf{c} = \mathbf{H}\mathbf{y}_m \quad (6)$$

where, $\mathbf{y}_m = \mathbf{y} + \mathbf{n}^y$ is the vector of available measurements, which generally is corrupted by measurement noise \mathbf{n}^y , and $\mathbf{H} \in \mathbb{R}^{n_c \times n_y}$ is the measurement combination or selection matrix. In addition to finding \mathbf{H} , we must also decide on the setpoint \mathbf{c}_s which is typically chosen as the nominal optimal value, $\mathbf{c}_s = \mathbf{H}\mathbf{y}_0^{opt}$, where the subscript 0 denotes the nominal operation point with $\mathbf{d} = \mathbf{d}_0$.

Several approaches can be used to calculate the optimal measurement combination $\mathbf{c} = \mathbf{H}\mathbf{y}$. The reader is referred to [17] for a comprehensive review. Most approaches are based on local linearization around the nominal optimal point. In this paper, we consider the exact local method as introduced in [14] and further developed in [18] and [19]. In this method, the optimization problem (3) is approximated by a quadratic approximation and a linearized measurement model. Let the linearized measurement model be represented by,

$$\mathbf{y} = \mathbf{G}^y\mathbf{u} + \mathbf{G}_d^y\mathbf{d} \quad (7)$$

where $G^y \in \mathbb{R}^{n_y \times n_u}$ and $G_d^y \in \mathbb{R}^{n_y \times n_d}$ are the gain matrices from \mathbf{u} to \mathbf{y} and \mathbf{d} to \mathbf{y} respectively. The optimal selection matrix \mathbf{H} , in terms of minimizing the loss for J_{ss} with respect to the expected disturbances and measurement noise, is then given by [18]

$$\mathbf{H}^T = (\mathbf{Y}\mathbf{Y}^T)^{-1}G^y \quad (8)$$

where,

$$\mathbf{Y} = [\mathbf{F}\mathbf{W}_d \quad \mathbf{W}_{n^y}] \quad (9)$$

and \mathbf{W}_d and \mathbf{W}_{n^y} are diagonal scaling matrices for the expected magnitudes of the disturbance and the measurement noise, respectively. $\mathbf{F} = \frac{\partial \mathbf{y}^{opt}}{\partial \mathbf{d}}$ is the optimal sensitivity matrix which describes how the measurement vector that correspond to the optimal operation \mathbf{y}^{opt} , change with unit change in the disturbance. The optimal sensitivity matrix may be determined analytically using

$$\mathbf{F} = -(G^y J_{uu}^{-1} J_{ud} - G_d^y) \quad (10)$$

or it may also be determined numerically by perturbing the disturbances and re-solving the optimization problem as described in [20]. Note that the optimal \mathbf{H} in (8) is not unique, but the non-uniqueness may be absorbed into the controller.

As seen from the equations above, the optimal selection matrix \mathbf{H} in (8) is based on the plant model G^y and the optimal sensitivity matrix \mathbf{F} for the expected disturbances. Due to the linearization around the nominal optimal point, the controlled variables combination is only locally valid around this nominal optimal point. If a disturbance moves the process far from the nominal optimal point, the local model approximation may be poor, resulting in higher steady-state loss. Over time, as the plant-model mismatch increases, the increase in the loss may no longer be acceptable. This requires re-optimization and computation of new optimal set-points \mathbf{c}_s . Additionally, any unmodelled disturbances that are not accounted for in the optimal sensitivity matrix cannot be handled efficiently.

2.2. Extremum-seeking control

Extremum-seeking control is a model-free optimization method, where the steady-state performance of the system is optimized purely based on measuring the cost. The objective is to drive the estimated steady-state gradient of the cost J_u to zero. The main advantage of extremum-seeking control compared to many other real-time optimizers is that no plant model is required. This enables extremum-seeking control to optimize the performance of complex systems where the process model is not known accurately. The main disadvantages are that it requires that the cost function is measured and that the convergence can be very slow.

Unlike self-optimizing control, which is based on local linearization of the *model* around the nominal operating point, extremum-seeking control is based on local linearization of the *measured cost* around the current operating point. The input and the cost measurements are used to continuously estimate the steady-state gradient J_u around the current operating point. The estimated gradient is then controlled to a setpoint of zero.

There are different ways of estimating the gradient based on the input and cost measurements. The classical approach is based on exciting the system with a sinusoidal signal and using a correlation based on high-pass and low-pass filters to retrieve the steady-state gradient information [6]. An alternative extremum-seeking scheme method was proposed in [16], where a linear least square estimation method was used to estimate the steady-state gradient which allows for a more general class of excitation signals. The least square method is simple to implement and has fewer

tuning parameters than the classical method. The least squares approach also provides a natural platform extending to multi-variable systems. Improved performance using a recursive least squares approach was also reported in [21]. Therefore, we use the least-square based extremum-seeking control in the rest of the paper.

In this work, we extend the least-square based gradient estimation presented in [16] to a multivariable system. The goal is to estimate the gradient from the inputs $\tilde{\mathbf{u}}$ to the measured cost J . In the least-squares based extremum-seeking control, the last N samples of data is used to fit a local linear cost model of the form,

$$J = J_{\tilde{\mathbf{u}}}^T \tilde{\mathbf{u}} + m \quad (11)$$

where $J_{\tilde{\mathbf{u}}} \in \mathbb{R}^{n_u}$ is the vector of gradients from $\tilde{\mathbf{u}}$ to J and $m \in \mathbb{R}$ is the bias.

At the current sample time k , let $\tilde{\mathbf{J}} = [J_k \dots J_{k-N+1}]^T \in \mathbb{R}^N$ be the vector of the last N samples of the measured cost and $\tilde{\mathbf{U}} = [\tilde{\mathbf{u}}_k \dots \tilde{\mathbf{u}}_{k-N+1}]^T \in \mathbb{R}^{N \times n_u}$ be the vector of the last N samples of the input. A moving window of fixed length N is then used to estimate the gradient using the linear least squares method [22]

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \|\tilde{\mathbf{J}} - \Phi^T \boldsymbol{\theta}\|_2^2 \quad (12)$$

where $\boldsymbol{\theta} \in \mathbb{R}^{(n_u+1)}$ is the vector of parameters to be estimated and is given by

$$\boldsymbol{\theta} = [J_{\tilde{\mathbf{u}}}^T \quad m]^T \quad (13)$$

and $\Phi \in \mathbb{R}^{N(n_u+1)}$ is the regressor vector given by

$$\Phi = \begin{bmatrix} \tilde{\mathbf{u}}_k^T & 1 \\ \tilde{\mathbf{u}}_{k-1}^T & 1 \\ \vdots & \vdots \\ \tilde{\mathbf{u}}_{k-N+1}^T & 1 \end{bmatrix} = [\tilde{\mathbf{u}} \quad \mathbf{1}]^T \quad (14)$$

The analytical solution to the least squares problem is given by [22]

$$\hat{\boldsymbol{\theta}} = [\Phi^T \Phi]^{-1} \Phi^T \tilde{\mathbf{J}} \quad (15)$$

The application of ordinary least squares requires that $N > n_u$. Note that in theory, it is not necessary to use a dither signal when this approach is used, but for practical purposes it is recommended, and in our case study we use a sinusoidal dither signal with an appropriately chosen amplitude.

Once the gradient vector $J_{\tilde{\mathbf{u}}}$ is estimated, n_u integral controllers can be used to drive the gradients to zero using as degrees of freedom $\tilde{\mathbf{u}}$ (setpoints to the lower level controllers). The integral controller in general can be written as,

$$\tilde{\mathbf{u}}_{k+1} = \tilde{\mathbf{u}}_k + T_s \mathbf{K}_I \hat{J}_{\tilde{\mathbf{u}}} \quad (16)$$

where $\mathbf{K}_I \in \mathbb{R}^{n_u \times n_u}$ is the gain matrix and T_s is the sample time. In this paper we use decentralized control where \mathbf{K}_I is diagonal. A schematic representation of the least squares-based extremum-seeking control is shown in Fig. 1

An additional dither signal is added to the control input to provide sufficient excitation in the input and cost measurements for accurate gradient estimation. The frequency of the dither signal must be significantly slower than the plant dynamics such that the plant can be approximated as a static map as described in [5,21] and the references therein. For a multi-input system, each input should have a unique perturbation frequency in order to estimate the gradient of the cost measurement with respect to each input.

In order to estimate the static gradient $\hat{J}_{\tilde{\mathbf{u}}}$ using dynamic data, the integral gain \mathbf{K}_I must be chosen small enough such that the time-scale of the gradient estimation is even slower than that of the

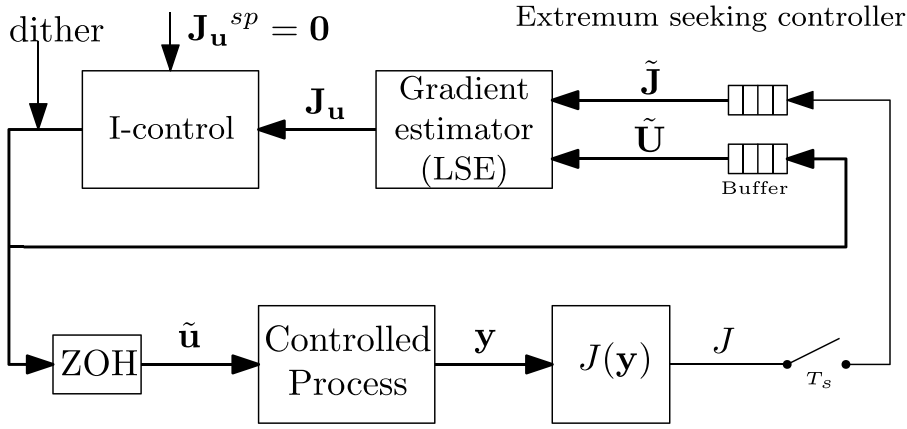


Fig. 1. Block diagram of the least-square based extremum-seeking controller.

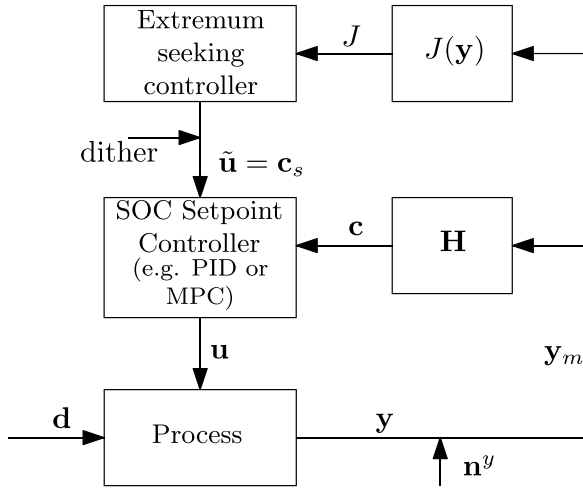


Fig. 2. Hierarchical implementation of combined self-optimizing control and extremum-seeking control. The extremum-seeking controller used in this paper is shown in Fig. 1.

dither signal. Since the linear model assumption is valid only locally around the current operating point, the gradient is estimated using only recent samples of data (i.e. a moving window of fixed length N_4). It was shown in [16] that the least-squares based extremum-seeking control is stable and that the error is small for a sufficiently small product of the adaptation gain and sample size $K_l N$.

Since the gradient estimation relies entirely on the cost measurements, it requires accurate cost measurements. The convergence to the optimum will also be slow for a dynamic process.

3. Proposed method

In this paper, we propose an hierarchical implementation with separate optimization and control layers as proposed in [14] and shown in Fig. 2.

Due to the timescale separation required between the optimization and control layers [3], the extremum-seeking controller is in the slow optimization layer and thus replaces the conventional RTO. Self-optimizing control is in the faster setpoint control layer below and tracks the updated setpoint given by the extremum-seeking controller. In other words, the extremum-seeking controller uses the measured cost J to compute the setpoint \mathbf{c}_s which is provided to the self-optimizing control. The con-

troller output from the extremum-seeking controller is $\tilde{\mathbf{u}} = \mathbf{c}_s$ in (11)–(15).¹

It may be argued that the self-optimizing control layer is redundant since an extremum-seeking scheme can directly manipulate the process to optimize the objective function. However, by using a purely data-driven approach, we ignore any *a-priori* knowledge about the system and the effect of disturbances. In addition, the extremum-seeking controller does not make use of measurements besides the cost measurements. Finally, the convergence to the optimum is slow. The proposed hierarchical combination of extremum-seeking control and self-optimizing control avoids the shortcomings of the extremum-seeking scheme and improves the convergence to the optimum. This is primarily due to a faster initial reaction of the self-optimizing layer to known (modelled) disturbances. Following a disturbance, the self-optimizing control quickly brings the operation point close to the optimal region, and on a slower timescale, the extremum-seeking control fine-tunes the setpoint and removes any loss associated with the self-optimizing control.

The extremum-seeking layer handles the plant-model mismatch and unmodelled disturbances and removes any steady-state loss by adjusting the setpoint \mathbf{c}_s . This also avoids re-optimization e.g. using real-time optimization.

In summary, we use the knowledge about the system to stay in the near-optimal region using self-optimizing control in the presence of disturbances. The extremum-seeking control helps to remove, or at least reduce the losses due to plant-model mismatch, it handles any unexpected disturbances, and it fine-tunes the optimal operating point. The key properties of the two methods are summarized and compared in Table 1, which shows that the self-optimizing control and extremum-seeking control are complementary rather than competing.

Note that, the proposed method is not just restricted to the least squares-based extremum-seeking control, and other extremum-seeking control methods, such as classical extremum-seeking control, or recursive least squares based extremum-seeking control etc. [6,21,5] may be used instead on top of the self-optimizing control layer in Fig. 2.

3.1. Stability issues

In this section, we provide some discussions on the stability of the combined self-optimizing control and extremum-seeking

¹ In the case where extremum-seeking controller controls the plant directly, $\tilde{\mathbf{u}} = \mathbf{u}$.

Table 1
Properties of self-optimizing control and extremum-seeking control.

Self-optimizing control	Extremum-seeking control
Offline model required	Model free
Fast rejection of disturbances	Slow rejection of disturbances
Local linearization around nominal optimal point	Local linearization around current operating point
Handles unmeasured but expected disturbances	Handles unmeasured and unexpected disturbances
Needs no cost measurement	Requires measurement of cost

control layout presented in Fig. 2. As mentioned earlier, extremum-seeking controller has three timescales, namely,

- fast – controlled plant dynamics
- medium – dither frequency
- slow – convergence to the optimum

A good extremum-seeking controller is tuned such that there is a clear timescale separation between these three timescales. The self-optimizing controller is included in the controlled plant and belongs to the fast timescale [6]. By introducing the self-optimizing control below the extremum-seeking control, the perturbation frequency and the adaptation (integral) gain \mathbf{K}_I must be chosen such that the timescale separation enables the static map assumption. Therefore, when seen from the slow timescale of the extremum-seeking controller, the closed-loop system comprising of the self-optimizing controller and plant is a static map $J = h(\mathbf{c}_s)$.

The stability results for both the classical extremum-seeking controller provided in [6] and the least squares based extremum-seeking controller provided in [16], both assume a smooth stabilizing control law parameterized by a “performance parameter”. This performance parameter is used as the handle by the extremum-seeking controller. In our paper, the control law is given by the self-optimizing control layer and the “performance parameter” is equivalent to the setpoint for the self-optimizing variable \mathbf{c}_s .

To summarize, existing stability results from [6] for the classical extremum-seeking control and [16] for the least squares based extremum-seeking control also hold for the combined hierarchical structure in Fig. 2, if the following two conditions are met:

- 1 The self-optimizing setpoint control layer is closed-loop stable.
- 2 The perturbation frequency of the ESC is sufficiently small compared to the timescale of the controlled plant which includes the stabilizing self-optimizing controller.

Choice of tuning parameters. According to the stability analysis results in [16], the product $\mathbf{K}_I N$ is the only important quantity that must be chosen small enough. This is a reasonable measure, since a small product of the adaptation gain \mathbf{K}_I and the time window N intuitively means that the measurements used for gradient estimation are in the small neighbourhood of the current operating point \mathbf{u} . As for the choice of the window length for the past measurements, N must be sufficiently small such that the error in the gradient estimate is bounded as described in detail in [16]. Alternatively, instead of using a fixed moving window of the past N samples, a forgetting factor may be used in a recursive least squares estimation framework, as adopted in [21]. As for any extremum-seeking control framework, the adaptation gain \mathbf{K}_I must be chosen sufficiently small, such that there is a clear time scale separation between the dither and the convergence to the optimum. This is required in order to validate the assumption of the plant being a static map, which is required for the extremum-seeking control. For more detailed guidelines on tuning the adaptation gain, the reader is referred to [5].

3.2. Effect of disturbances

Very little of the literature on extremum-seeking control consider explicitly the effect of disturbances. Disturbances typically trigger fast dynamics, which may invalidate the assumption of the plant operating close to a static map. Consequently, the introduction of the fast dynamics leads to erroneous gradient estimation, especially with the least squares gradient estimation method used in this paper. In other words, if the cost measurement in the gradient estimation time window is in transients due to the disturbances, then the least squares method fits a wrong gradient $\hat{\mathbf{J}}_{\mathbf{u}}$. The effect of abrupt disturbances on the extremum-seeking scheme has been well motivated in [9,11], and [10], along with some modifications to improve disturbance rejection. However, all these modifications require the disturbances to be measured. Measured disturbances may also be handled by the least squares-based extremum-seeking scheme described in Section 2.2, by explicitly including the measured disturbances as a part of the regressor Φ in (15) and replacing (11) with

$$J = \mathbf{J}_{\mathbf{u}}^T \hat{\mathbf{u}} + \mathbf{J}_{\mathbf{d}'}^T \mathbf{d}' + m \quad (17)$$

where $\mathbf{J}_{\mathbf{d}'} \in \mathbb{R}^{m_{d'}}$ is the vector of gradients from the measured disturbances \mathbf{d}' to the cost J .

Unfortunately, unmeasured disturbances may still result in erroneous gradient estimation. Given that the extremum-seeking control problem at hand is essentially a static optimization problem, the only way to avoid this problem is to use a steady-state detection, as used in the traditional steady-state RTO. The extremum-seeking scheme can be triggered only if the cost measurement in the gradient estimation window is close to steady-state operation. Dynamic changes in the cost measurement resulting from a disturbance will be flagged by the steady-state detection routine and the extremum-seeking scheme is temporarily halted until the cost measurement in the gradient estimation time window comes close to steady-state operation. By doing so, the static map assumption used by the extremum-seeking scheme is always valid. This halt is typically known as the *steady-state wait time* and is commonly used in the traditional steady-state RTO paradigm. In fact, the method proposed in [11] is precisely a steady-state wait time routine implemented using a supervisory state machine.

In many processes with long settling times, the steady-state wait time can lead to a very slow convergence to the optimum. Alternatively, in some process systems, one can make use of some heuristics to avoid the steady-state wait time, such as to bound the magnitude of the individual gradients ($\hat{\mathbf{J}}_{\mathbf{u}_i}$) in (16) to a value $\mathbf{J}_{\mathbf{u}_i, \max}$

$$|\hat{\mathbf{J}}_{\mathbf{u}_i}| \leq \mathbf{J}_{\mathbf{u}_i, \max} \quad (18)$$

The bounding limits aggressive input usage. Based on (16), we propose to introduce a maximum change in the input between samples, $\Delta \hat{\mathbf{u}}_{i, \max}$, and choose,

$$\mathbf{J}_{\mathbf{u}_i, \max} = \frac{\tilde{\Delta \hat{\mathbf{u}}_{i, \max}}}{T_s \mathbf{K}_I} \quad (19)$$

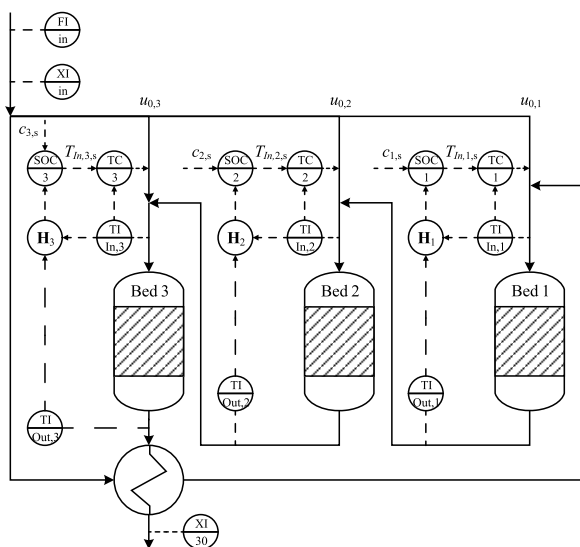


Fig. 3. Flowsheet of the reactor, modified from [23] to include the proposed control structure.

Note that although this approach is used in our case study as illustrated in Fig. 7, this additional heuristic of bounding the gradient is not part of the core methodology presented in this paper, but should be viewed as an alternative approach to the steady-state wait time.

4. Case study – ammonia synthesis reactor

In this paper, we apply the proposed method to a three-bed ammonia reactor with heat integration. A flowsheet, including the control structure for the proposed method, can be found in Fig. 3. The model was first described in [23] for stability analysis of the ammonia reactor. It was recently utilized in [24] for the application of economic non-linear model predictive control (E-NMPC). The E-NMPC approach was able to reject disturbances and to avoid limit-cycle behaviour while achieving optimal operation. However, plant-model mismatch and the non-linear problem may limit its practical application. Furthermore, in order to avoid repeated numerical optimization and to handle disturbances and plant-model mismatch more efficiently, we consider in this paper self-optimizing control, extremum-seeking control and a combination of the both approaches. The incorporation of the reactor into the ammonia synthesis loop requires adjustments to the setpoints of the self-optimizing controllers as well [25].

The objective is to maximize the extent of reaction ξ for a given feed, that is

$$J = \xi \quad (20)$$

$$= \dot{m}_{in} (w_{\text{NH}_3,30} - w_{\text{NH}_3,in}) \quad (21)$$

As a maximized conversion *per pass* (or alternatively a maximized extent of reaction for a given feed) results in less reactants recycled, we have a smaller feed to the reactor resulting in an even higher conversion *per pass*. Consequently, the positive feedback of the extent of reaction results in minimizing the recycle flow, and therefore, the cooling cost and the recycle compressor duty.

4.1. Summary of the model

The model consists of three sequential reactor beds and one heat exchanger. The inlet stream to the reactor system (denoted by subscript *in*) is split into four streams; one quench flow to each bed and a preheated flow to the first reactor bed. The quench

Table 2
PI tuning parameters of the temperature and SOC controllers in Fig. 3.

		Input (MV)	Output (CV)	K_p [-]	K_I [s^{-1}]
Slave	TC 1	$u_{0,1}$	$T_{in,1}$	[-]	-2.1×10^{-4}
	TC 2	$u_{0,2}$	$T_{in,2}$	[-]	-2.7×10^{-4}
	TC 3	$u_{0,3}$	$T_{in,3}$	[-]	-4.2×10^{-4}
Master	SOC 1	$T_{in,1,s}$	c_1	0.169	0.76×10^{-3}
	SOC 2	$T_{in,2,s}$	c_2	0.209	1.25×10^{-3}
	SOC 3	$T_{in,3,s}$	c_3	1.000	5.0×10^{-2}

split ratios correspond to the three manipulated variables $\mathbf{u}_0 = [u_{0,1} \ u_{0,2} \ u_{0,3}]^T$. The three reactor beds are discretized into a cascade of continuously stirred tank reactors (CSTR). We use the Temkin-Pyzhev kinetic expression for the reaction rate. The heat-exchanger is modelled using the number of transfer units (NTU) method. The resulting model without controllers corresponds to a differential-algebraic system with $\mathbf{x} \in \mathbb{R}^{30}$ as dynamic state variables corresponding to the temperatures in the beds, $\mathbf{z} \in \mathbb{R}^{30}$ as algebraic state variables corresponding to the ammonia mass fractions in the beds, and $\mathbf{u}_0 \in \mathbb{R}^3$ as manipulated variables. A detailed model description can be found in [24]. The system is modelled using CasADi [26]. The nominal optimal point and the optimal sensitivity matrix \mathbf{F} for self-optimizing control were computed using the IPOPT non-linear problem solver [27]. The plant model was simulated using IDAS [28].

4.2. Controller design

The potential instability in case of disturbances as described in [23] requires a stabilizing “slave” control layer below the self-optimizing control layer. It was shown in [24], that if the reactor is operated close to the nominal optimum and without control, reactor extinction may result even from small disturbances compared to the large disturbances investigated by [23]. Hence, also for the case when extremum-seeking control is utilized without self-optimizing control, a stabilizing “slave” control layer is required resulting in a cascade control structure in all investigated control structures. The chosen “slave” controllers are temperature controllers and will be explained in the following subsection. The slave temperature loops as well as the master self-optimizing control loops were tuned using the SIMC rules [29]. These controllers were directly included into the differential-algebraic model increasing the number of differential variables by three for the extremum-seeking control or six for the combined self-optimizing and extremum-seeking control. The extremum-seeking controllers were implemented in discrete time resulting in a discrete-continuous representation.

Slave temperature controller pairing and tuning

The slave controllers use the splits (bypass) $u_{0,i}$ to control the corresponding bed inlet temperature. This is a pure mixing process with instantaneous dynamics, and an integrating controller is recommended [29]. The desired closed loop time constant for the three controllers was chosen to be $\tau_c = 10$ s. The resulting integral gain K_I for the three temperature loop controllers can be found in Table 2.

SOC controller pairing and tuning

In our case, the SOC controllers give the setpoints to the respective slave temperature controllers. The measurements \mathbf{y} for

self-optimizing control are selected to be the inlet and outlet temperature of each reactor bed; *i.e.*

$$\mathbf{y}_i = \begin{bmatrix} T_{in,i} \\ T_{i0} \end{bmatrix} \quad \forall i = 1, 2, 3 \quad (22)$$

Hence, only two measurements were used for the calculation of \mathbf{H}_i in (8). This local treatment of each bed does not necessarily result in overall optimal selection matrices \mathbf{H}_i . It would be possible to increase the number of measurements, *e.g.* using all 6 measurements for the calculation of the selection matrices. This will reduce somewhat the steady-state loss in self-optimizing control [19]. However, it may also lead to undesired dynamic behaviour through coupling and delays in the self-optimizing variables \mathbf{c} .

The disturbances are the inlet conditions;

$$\mathbf{d} = [\dot{m}_{in} \quad p_{in} \quad T_{in} \quad w_{\text{NH}_3, in}]^T \quad (23)$$

The chosen scaling matrices in (9) are

$$\mathbf{W}_d = \begin{bmatrix} 7 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 0.008 \end{bmatrix} \quad (24)$$

$$\mathbf{W}_{n^v} = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} \quad (25)$$

These scaling matrices are based on the expected magnitude of the disturbances and measurement noise. The expected disturbance magnitudes (\mathbf{W}_d) are 10% of the nominal feed values whereas \mathbf{W}_{n^v} corresponds to a measurement noise of 4 K for each measurement. The resulting (scaled) selection matrices \mathbf{H}_i from (8) are then given by

$$\mathbf{H}_1 = \begin{bmatrix} -0.03 \\ 1.00 \end{bmatrix}, \quad \mathbf{H}_2 = \begin{bmatrix} 0.23 \\ 1.00 \end{bmatrix}, \quad \mathbf{H}_3 = \begin{bmatrix} 1.00 \\ 0.74 \end{bmatrix} \quad (26)$$

To get a fast response in the self-optimizing control layer, (as it will be combined with an upper extremum-seeking layer), the closed-loop time constant τ_c for each of the three controllers is equal to its respective time delay, which is in the range of 300 s. The resulting PI parameters (K_p and K_I) can be found in Table 2.

The introduction of “slave” temperature controllers reduces the coupling between the self-optimizing controllers as is shown by the relative gain array [30]. The relative gain array of the self-optimizing controllers including the “slave” temperature loops are given by a value close to 1 in the diagonal elements. Without the “slave” temperature loops, we receive diagonal elements close to 2 which indicates a stronger coupling in-between the controlled variables \mathbf{c} .

Extremum-seeking controllers tuning

The upper layer in the control structure in Fig. 2 consists of the extremum-seeking controllers. These slow integral controllers give the setpoints to either the base layer temperature (\mathbf{T}_s , denoted T+ESC) or the self-optimizing controllers (\mathbf{c}_s , denoted T+SOC+ESC). The estimation of the gradient $\hat{\boldsymbol{\theta}}$ according to (15) is performed using $\hat{\mathbf{u}}$ as the setpoint of the respective slave controller (T or SOC). It is assumed that the disturbances are unmeasured. Hence, $\boldsymbol{\theta}$ is given by

$$\boldsymbol{\theta} = [\mathbf{J}_{\hat{\mathbf{u}}}^T \quad \mathbf{m}]^T \quad (27)$$

where m is the present value of the cost and $\mathbf{J}_{\hat{\mathbf{u}}}$ the gradient. As the disturbances are not corrected for, this will result in wrong gradient estimation when disturbances occur. One way to rectify this problem is to temporarily turn off the extremum-seeking controllers

Table 3

Controller tuning parameters for the extremum-seeking controllers in the case of only temperature controllers (T) and also self-optimizing control (SOC) layer providing setpoints to the temperature controller.

	Controller	Amplitude [K]	Period [h]	$J_{u_i, \max}$ [kg s ⁻¹ K ⁻¹]	K_I [s ⁻¹]
T	ESC 1	1.0	2.0	0.3	0.05
	ESC 2	1.0	2.5	0.225	0.05
	ESC 3	1.0	3.0	0.225	0.05
SOC	ESC 1	1.0	2.0	0.3	0.08
	ESC 2	1.0	2.5	0.45	0.04
	ESC 3	1.0	3.0	0.45	0.04

when this happens. It may, however, not be obvious that a disturbance is occurring. Hence, instead the gradients $\hat{\mathbf{J}}_{\mathbf{u}_i}$ are bounded as shown in (18).

The tuning of extremum-seeking controllers depends on many factors. We need to choose the number of past measurements N , the periods and amplitudes of the sinusoidal dithers, as well as the integral gains. All these parameters have an influence on each other resulting in a difficult selection. The parameters were chosen based on trial and error to achieve satisfactory performance and are given in Table 3. The time horizon for the past measurements was chosen to be 1 h in all cases. This corresponds to $N=240$ samples with a chosen integrator step time of $t_{int}=15$ s. Equal effort for both T+ESC and T+SOC+ESC tuning was attempted to achieve comparable performance.

4.3. Results

In order to compare the proposed methods, two disturbances were investigated; a disturbance in the inlet mass flow rate \dot{m}_{in} , corresponding to a modelled disturbance in self-optimizing control, and an unmodelled disturbance in the reaction rate r . These disturbances were chosen as they correspond to the largest losses for the self-optimizing control structure (not shown). Hence, the improvement using extremum-seeking control is most pronounced. In addition, both disturbances would result in reactor extinction, if the stabilizing temperature controllers would not be present. The integrated loss (cost difference),

$$J_{int}(t) = \int_0^t [\xi_{opt,ss}(t') - \xi(t')] dt' \quad (28)$$

is used to compare the proposed methods.

The first considered disturbance is a +20% step change in the inlet mass flow rate as this disturbance results in the highest steady-state loss for self-optimizing variables [25]. This disturbance is considered in the calculation of the SOC variables. The cost $J=\xi$ and the integrated loss (28) are shown in Fig. 4. The cases with extremum-seeking control (solid lines) settle to the new optimum in contrast to pure self-optimizing control (dashed red line). The combination of self-optimizing control and extremum-seeking control gives a large reduced loss in produced tons of ammonia. As seen in Fig. 4, this reduction corresponds to 4.95 t ammonia in the investigated time-frame of 18 h. One could argue that this is caused by suboptimal tuning parameters in the pure extremum-seeking control. By taking a look at the time the disturbance is occurring, we claim that this is not the case. Fig. 5 shows a close-up of the response in the cost function for the first 1.2 h after the disturbance occurs. From this figure, it can be clearly seen that both ESCs (solid lines) initially follow their respective slave controllers, before deviating when the ESCs start changing the setpoints to the slave controllers. Both ESC control structures are in fact moving initially in the wrong direction, that is, to a reduced extent of reaction. This can be explained by the past measurements, before the disturbance,

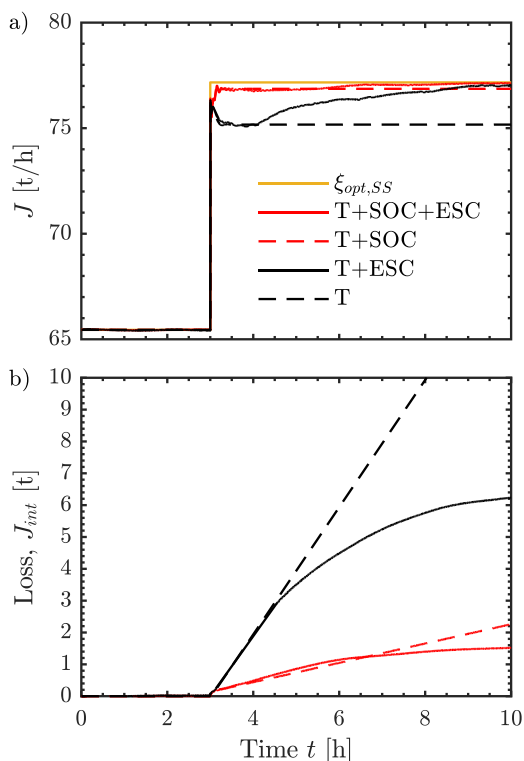


Fig. 4. Response of (a) extent of reaction ξ and (b) integrated loss to a +20% disturbance in inlet mass flow rate, $\Delta \dot{m}_{in} = +54 \text{ t/h}$, at $t = 3 \text{ h}$.

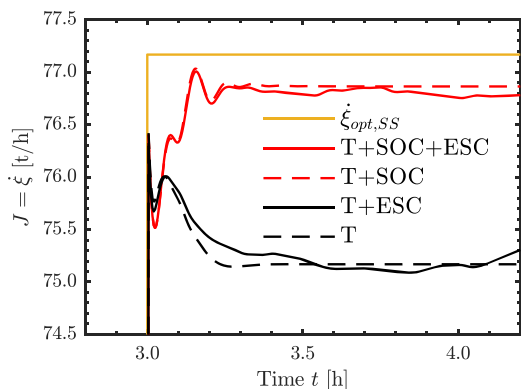


Fig. 5. Closeup of Fig. 4(a) at the time when the disturbance occurs.

which are still used at this point. One approach to circumvent this behaviour is to use a smaller time horizon (smaller N). This results on the other hand in a drift away from the optimal setpoint on a long time scale. Hence, it is preferable to have a slightly suboptimal initial performance.

A disturbance in the reaction rate r is an unmodelled disturbance which is not considered in the calculation of the optimal selection matrices according to (8). It can be considered a plant-model mismatch. The simulation results for a -20% step change in the reaction rate r are shown in Fig. 6. Similarly to a disturbance in the inlet mass flow \dot{m}_{in} , the control structure based on the proposed method with both self-optimizing and extremum-seeking control settles to the new optimum after 7 h whereas extremum-seeking control alone requires around 13 h. During the time the controllers require to settle to the new optimum, the loss is reduced in the proposed control structure with SOC. Over 18 h, the proposed control structure has a reduced loss of 6.71 tons of produced ammonia. Here it has to be noted, that despite this disturbance was not included in the

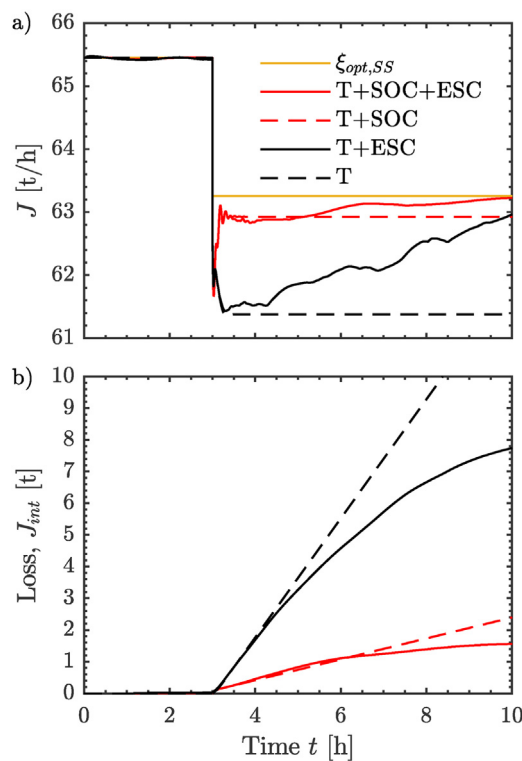


Fig. 6. Response of (a) extent of reaction ξ and (b) integrated loss to a -20% disturbance in pre-exponential factors of the Arrhenius equations at $t = 3 \text{ h}$.

design phase, the self-optimizing control structure has a reduced loss. This can be explained by general favourable properties of self-optimizing feedback with regard to disturbances and plant-model mismatch.

In Section 3.2 we proposed a bounding heuristic in (18) for $\tilde{\mathbf{J}}_{\mathbf{u}_i}$ to reduce the effect of erroneous gradient estimates caused by disturbances, which was used in the above simulations. Fig. 7 shows the bounds and the gradient estimate for gradient 1 as well as the corresponding manipulated variable, $c_{s,1}$, for a +20% step change in the inlet mass flow rate. As we can see, the gradient estimate at the time of the disturbance ($t = 3 \text{ h}$) is indeed outside the bound in (18).

At $t = 3.025 \text{ h}$ it reaches a minimum value of $\tilde{\mathbf{J}}_{\mathbf{u}_i} = -93.42 \text{ kg s}^{-1} \text{ K}^{-1}$, a value 300 times as large as the bound. The estimate is back within the bound one hour after the occurrence of the disturbance ($t = 4 \text{ h}$). This corresponds to the time horizon of the past measurements according to (15). If no bounds are introduced, the wrong estimate of the gradient would require a very small integrator gain K_I for the extremum-seeking controller resulting in a very slow convergence.

In Fig. 8 we compare the proposed correction scheme with steady-state detection and with no correction. The controller tunings were adjusted in both additional cases. As we can see in Fig. 8, no correction on the gradient results in worse performance. The extremum-seeking controllers have to be tuned slower to avoid moving too far into the “wrong” direction. (See Fig. 8a). As a result, the convergence to the new steady-state optimum is much slower. Steady-state detection as an alternative was implemented by deactivating the extremum-seeking controllers if the variance of the N past measurements of controlled variables of the self-optimizing controllers is above a certain threshold (as commonly used in traditional RTO schemes). We see that the convergence rate to the new optimum is similar to bounding the gradient. However, the loss is increased in the first hour while the extremum-seeking controllers are deactivated.

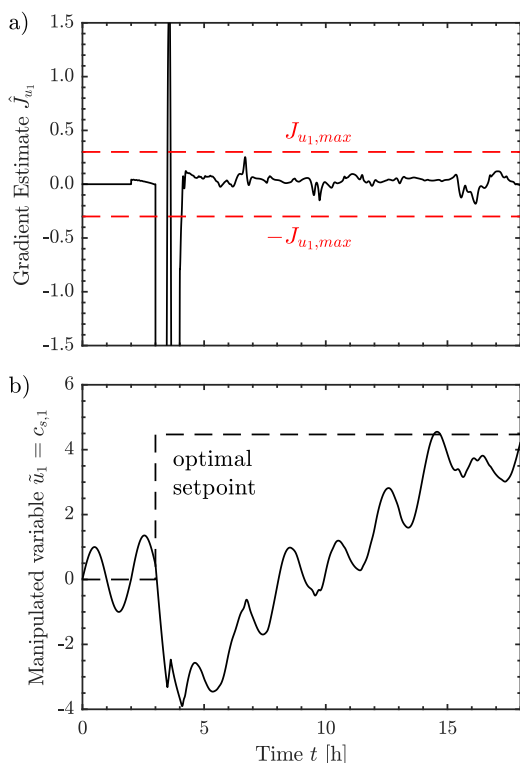


Fig. 7. (a) Gradient 1 estimate and (b) resulting controller output $\tilde{u}_1 = c_{s,1}$ with the proposed correction in (18).

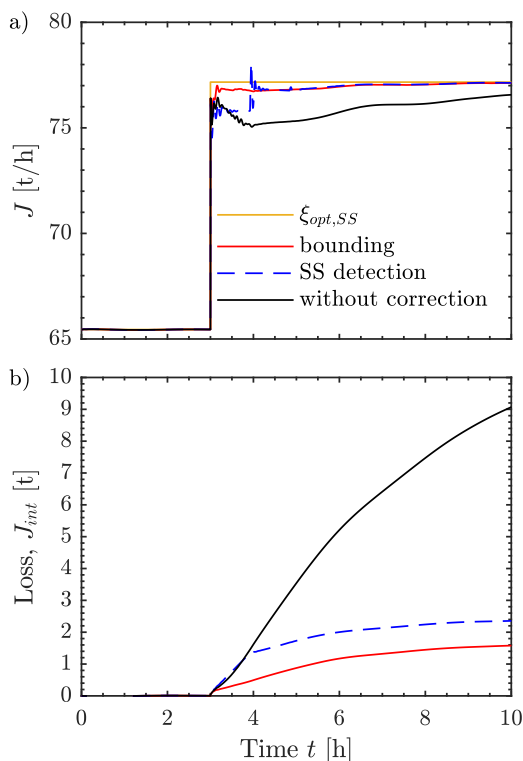


Fig. 8. Comparison of the proposed bounding scheme for correcting gradient estimate (red solid lines) with steady state detection (black dashed lines) and no correction (black solid lines). Response of (a) extent of reaction ξ and (b) integrated loss to a +20% disturbance in inlet mass flow rate, $\Delta \dot{m}_{in} = +54$ t/h, at $t = 3$ h. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5. Discussion

5.1. General discussion

As shown, the hierarchical combination of self-optimizing control with extremum-seeking control improves the rejection of disturbances in the ammonia-reactor case study. This is caused by the (fast) rejection of the disturbance through self-optimizing control combined with the final adjustment of the setpoints c_s by the extremum-seeking controllers. Is it still possible to speak of self-optimizing control in the context, when the setpoint is adjusted? Yes, the idea of self-optimizing control is to allow for less frequent changes in the setpoint. Skogestad [3] mentions in the original paper on self-optimizing control explicitly the possibility to adjust the SOC setpoints using an optimizing layer.

This is especially important considering the incorporation of the reactor into the synthesis loop in which the recycle is neglected [25]. The proposed method adjusts then the setpoints, if it is not possible to solve an optimization problem for the overall process.

Self-optimizing control is model-based. However, the proposed method is less reliant on the accuracy of the model than many other model-based approaches. This is because it only uses the model offline for the calculation of the optimal selection matrices H_i and the extremum-seeking controller accounts for any plant-model mismatch that is not accounted for in the self-optimizing layer below, hence making it more resilient to modelling errors. The setpoints of the self-optimizing controllers are handled by the extremum-seeking controllers in a model-free approach.

There remains however one limitation to the proposed methodology; it is necessary to measure (or estimate) the cost function for the extremum-seeking controller. This is an inherent limitation of extremum-seeking controllers.

5.2. Impact of the scaling matrices

The performance of the self-optimizing controllers depends on the scaling matrices given in Eqs. (24) and (25). As outlined in [14], these scaling matrices will scale the combined disturbance and measurement noise vector to unit length. If the magnitude of a disturbance or measurement noise changes, then the exact-local method will wrongly weigh the disturbance and the resulting measurement selection matrix is not optimal. Correspondingly, the steady state loss of the overestimated disturbance will be lower whereas the steady state loss of the other disturbances can be both higher or lower [25].

The combination of self-optimizing control with extremum-seeking control alleviates however the problem associated with the higher steady-state loss as the extremum-seeking controllers are adjusting the setpoints to the self-optimizing controllers. Consequently, the integrated loss may be increased with wrongly chosen scaling matrices but there will be still an improvement over pure extremum-seeking control.

6. Conclusion

We have shown that extremum-seeking control and self-optimizing control are complementary rather than competing. This is caused by the different timescale at which the control strategies are operating. By combining self-optimizing control and extremum-seeking control, we are able to utilize the advantages of each method and improve the convergence to the optimum. Using a three-bed ammonia reactor case study, we demonstrate that the combined system can handle unmeasured disturbances and at the same time correct for plant-model mismatch.

Acknowledgement

J.S. gratefully acknowledges the financial support from YARA International ASA and D.K. gratefully acknowledges the financial support from SUBPRO, which is financed by the Research Council of Norway, major industry partners and NTNU.

References

- [1] B. Srinivasan, D. Bonvin, E. Visser, S. Palanki, Dynamic optimization of batch processes: II. Role of measurements in handling uncertainty, *Comput. Chem. Eng.* 27 (1) (2003) 27–44.
- [2] B. Chachuat, B. Srinivasan, D. Bonvin, Adaptation strategies for real-time optimization, *Comput. Chem. Eng.* 33 (10) (2009) 1557–1567.
- [3] S. Skogestad, Plantwide control: the search for the self-optimizing control structure, *J. Process Control* 10 (5) (2000) 487–507.
- [4] V. Kumar, N. Kaistha, Hill-climbing for plantwide control to economic optimum, *Ind. Eng. Chem. Res.* 53 (42) (2014) 16465–16475.
- [5] Y. Tan, W. Moase, C. Manzie, D. Nešić, I. Mareels, Extremum seeking from 1922 to 2010, in: *Proceedings of the 29th Chinese Control Conference, IEEE, 2010*, pp. 14–26.
- [6] M. Krstić, H.-H. Wang, Stability of extremum seeking feedback for general nonlinear dynamic systems, *Automatica* 36 (4) (2000) 595–601.
- [7] M. Guay, D.J. Burns, A comparison of extremum seeking algorithms applied to vapor compression system optimization, in: *American Control Conference (ACC), 2014, IEEE, 2014*, pp. 1076–1081.
- [8] O. Trollberg, E.W. Jacobsen, Greedy extremum seeking control with applications to biochemical processes, *IFAC-PapersOnLine* 49 (7) (2016) 109–114.
- [9] D. Krishnamoorthy, A. Pavlov, Q. Li, Robust extremum seeking control with application to gas lifted oil wells, *IFAC-PapersOnLine* 49 (13) (2016) 205–210.
- [10] S. Marinkov, B. de Jager, M. Steinbuch, Extremum seeking control with adaptive disturbance feedforward, *IFAC Proc. Vol.* 47 (3) (2014) 383–388.
- [11] S. Marinkov, B. de Jager, M. Steinbuch, Extremum seeking control with data-based disturbance feedforward, *2014 American Control Conference (2014) 3627–3632*, <http://dx.doi.org/10.1109/ACC.2014.6858832>.
- [12] J. Jäschke, S. Skogestad, NCO tracking and self-optimizing control in the context of real-time optimization, *J. Process Control* 21 (10) (2011) 1407–1416.
- [13] B.D. Keating, A. Alleyne, Combining self-optimizing control and extremum seeking for online optimization with application to vapor compression cycles, in: *American Control Conference (ACC), 2016, IEEE, 2016*, pp. 6085–6090.
- [14] I.J. Halvorsen, S. Skogestad, J.C. Morud, V. Alstad, Optimal selection of controlled variables, *Ind. Eng. Chem. Res.* 42 (14) (2003) 3273–3284.
- [15] K.B. Ariyur, M. Krstić, *Real-time Optimization by Extremum-seeking Control*, John Wiley & Sons, 2003.
- [16] 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 Conference on Decision and Control, IEEE, 2014*, pp. 2679–2684.
- [17] J. Jäschke, Y. Cao, V. Kariwala, Self-optimizing control – a survey, *Annu. Rev. Control* 43 (2017) 199–223.
- [18] V. Alstad, S. Skogestad, E.S. Hori, Optimal measurement combinations as controlled variables, *J. Process Control* 19 (1) (2009) 138–148.
- [19] R. Yelchuru, S. Skogestad, Convex formulations for optimal selection of controlled variables and measurements using mixed integer quadratic programming, *J. Process Control* 22 (6) (2012) 995–1007.
- [20] V. Alstad, S. Skogestad, Null space method for selecting optimal measurement combinations as controlled variables, *Ind. Eng. Chem. Res.* 46 (3) (2007) 846–853.
- [21] 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.
- [22] L. Ljung, *System Identification: Theory for the User*, 2nd ed., PTR Prentice Hall, Upper Saddle River, NJ, 1999.
- [23] J.C. Morud, S. Skogestad, Analysis of instability in an industrial ammonia reactor, *AIChE J.* 44 (4) (1998) 888–895.
- [24] J. Straus, S. Skogestad, Economic NMPC for heat-integrated chemical reactors, *2017 21st International Conference on Process Control (PC) (2017) 309–314*, <http://dx.doi.org/10.1109/PC.2017.7976232>.
- [25] J. Straus, S. Skogestad, Self-optimizing control in chemical recycle systems, *IFAC-PapersOnLine* 51 (18) (2018) 530–535.
- [26] J. Andersson, *A General-Purpose Software Framework for Dynamic Optimization*, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium, 2013, October, PhD thesis.
- [27] A. Wächter, L.T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Math. Program.* 106 (1) (2006) 25–57.
- [28] A.C. Hindmarsh, P.N. Brown, K.E. Grant, S.L. Lee, R. Serban, D.E. Shumaker, C.S. Woodward, SUNDIALS: suite of nonlinear and differential/algebraic equation solvers, *ACM Trans. Math. Software (TOMS)* 31 (3) (2005) 363–396.
- [29] S. Skogestad, C. Grimholt, *The SIMC Method for Smooth PID Controller Tuning*, Springer London, London, 2012, pp. 147–175.
- [30] E. Bristol, On a new measure of interaction for multivariable process control, *IEEE Trans. Autom. Control* 11 (1) (1966) 133–134, <http://dx.doi.org/10.1109/TAC.1966.2661098>.