NCO tracking and Self-optimizing Control in the Context of Real-Time Optimization

Johannes Jäschke, Sigurd Skogestad*

Department of Chemical Engineering, Norwegian University of Science and Technology, NTNU, N-7491 Trondheim, Norway

Abstract

This paper reviews the role of self-optimizing control (SOC) and necessary conditions of optimality tracking (NCO tracking) as presented by [1]. We show that self-optimizing control is not an alternative to NCO tracking for steady state optimization, but is to be seen as complementary. In self-optimizing control, offline calculations are used to determine controlled variables (CVs), which by use of a lower layer feedback controller indirectly keep the process close to the optimum when a disturbance enters the process. Preferably, the setpoints are kept constant, but they may be adjusted by some optimization layer. Good CVs reduce the need for frequent setpoint changes. When selecting self-optimizing CVs, a set of disturbances must be assumed, as unexpected disturbances are not rejected in SOC. On the other hand, the presented NCO tracking procedure adapts the inputs at given sample times without a model or any assumptions on the set of disturbances. However, disturbances with high frequencies or those which do not lead to a steady state are not rejected. By using NCO tracking in the optimization layer and SOC in the lower control layer, we demonstrate that the methods complement each other, with SOC giving fast optimal correction for expected disturbances, while other disturbances are compensated by the model free NCO tracking procedure on a slower time scale.

Keywords: Self-optimizing control, NCO tracking, Real-time optimization, Optimal operation

*Correspondening author, tel: +47-735-94154
Email address: `skoge@chemeng.ntnu.no` (Sigurd Skogestad)

1. Introduction

Most processes in industrial practice are operated in such a way that the operators set the setpoints for PID controllers that keep the controlled variables (CVs) at the desired setpoint. Which measurements are chosen as CVs is mostly based on process knowledge and best practices. However, due to stronger competition and environmental regulations, it has become increasingly important to operate the processes close to optimality. In many cases, steady state operation accounts for the largest part of the operating cost, and significant economical improvements can be achieved by operating the plant optimally at steady state.

Depending on how this is realized, the methods for achieving optimal process operation generally may be categorized into one of the following three categories:

- Model used online (e.g. Real-time optimization (RTO) [2])
- Model used offline (e.g. self-optimizing control (SOC) [3])
- Explicit Model not used (e.g. NCO tracking [1])

In all cases, measurements are collected online, with the aim of driving the process towards optimality. In the first approach, online optimization, measurements from the process are used together with a mathematical model in a two step procedure, where first the model and the disturbances are updated, and then the new optimal setpoints are determined by solving an optimization problem online [2].

In the offline approach, expensive online computations are avoided, and optimal operation is achieved by designing a "smart" control structure. This controlled variable (CV) selection procedure's objective is to transform the economic objectives into control objectives [4]. A process model is used to support decision making in control structure design, but it will not be used online. Self-optimizing control [5] belongs into this category.

A third strategy avoids using an explicit process model, but uses measurements to obtain gradient information about the process. This information is used to update the inputs to obtain optimal operation. Necessary conditions of optimality tracking (NCO tracking) as presented in [1] and extremum seeking control [6] represent this category. This idea is relatively old [7], but has recently gained increased attention.

These approaches to achieve steady state optimal operation have been developed by research groups with different backgrounds for different kinds of problems. The authors feel that there has been some confusion about the use, interplay, applicability and practicability of some of the concepts.

Our paper is structured as follows: The next three sections briefly describe the setting and the ideas from self-optimizing control and NCO tracking. In particular, this work focuses on the null space method as described in [8], which uses a model offline, and the model free NCO tracking procedure for steady state optimization as described by [1].

In Section 5 we describe the framework in which we place the two methods and consider the properties of the two approaches. Based on this discussion, we consider the methods as *complementary* and propose to use them together. The ideas are illustrated by simulation results for a dynamic CSTR in Section 6, followed by a discussion in Section 7, and conclusions in Section 8.

2. Optimal operation

The task is to optimize the steady state operation of continuous systems. In virtually all practical cases plant operation is subject to operational and safety constraints, and the problem of achieving steady state optimal operation can be formulated as minimizing a scalar cost function J',

$$\min_{\mathbf{u}'_{all}} J'(\mathbf{u}'_{all}, \mathbf{x}', \mathbf{d}') \quad \text{s.t.} \quad \begin{cases} \text{plant: } \mathbf{g}(\mathbf{u}'_{all}, \mathbf{x}', \mathbf{d}') = 0 \\ \text{constraints: } \mathbf{h}(\mathbf{u}'_{all}, \mathbf{x}', \mathbf{d}') \leq 0, \end{cases} \tag{1}$$

where $\mathbf{u}'_{all} \in \mathbb{R}^{n_{\mathbf{u}'_{all}}}$ denotes the steady state degrees of freedom (e.g. a valve opening or a pump speed), $\mathbf{x}' \in \mathbb{R}^{n'_{\mathbf{x}}}$ denotes the states, and $\mathbf{d}' \in \mathbb{R}^{n_{\mathbf{d}'}}$ denotes the vector of unknown disturbances and parameters. Although, the signal $\mathbf{d}'$ may vary with time, we make a pseudo steady state assumption [4].

The $n_{\mathbf{g}}$-dimensional and $n_{\mathbf{h}}$-dimensional vector valued functions $\mathbf{g}$ and $\mathbf{h}$ denote the model equations and the operational constraints respectively. A constraint is said to be "active" if it is optimal to have $h_i = 0$ (equality). The set of active constraints may change depending on the disturbances. However, in this paper, we assume that the set of active constraints does not change with the disturbances.

In terms of plant safety and economy it is often significantly more important to satisfy the active constraints than to handle the unconstrained degrees of freedom optimally. Therefore, the first step when designing the control structure is to determine the active constraints, and to design a control system that keeps them close to the optimal value.

With all active constraints implemented, problem (1) can be re-written as an unconstrained optimization problem, and if we also formally eliminate the internal (state) variables $\mathbf{x}'$, the problem becomes

$$\min_{\mathbf{u}'} J''(\mathbf{u}', \mathbf{d}'), \tag{2}$$

where $\mathbf{u}' \in \mathbf{R}^{n_{n_{\mathbf{u}'}}}$ now denotes the remaining unconstrained degrees of freedom.

For notational convenience, we introduce a variable transformation, where the nominal optimal point is used as a reference:

$$\begin{aligned} \mathbf{u} &= \mathbf{u}' - \mathbf{u}^{opt'}(\bar{\mathbf{d}}') && (3) \\ \mathbf{d} &= \mathbf{d}' - \bar{\mathbf{d}}'. && (4) \end{aligned}$$

The transformed variables are $\mathbf{u} \in \mathbf{R}^{n_{\mathbf{u}}}$, and $\mathbf{d} \in \mathbf{R}^{n_{\mathbf{d}}}$, respectively, and $\bar{\mathbf{d}}' \in \mathbb{R}^{n_{\mathbf{d}'}}$ denotes the nominal disturbance and $\mathbf{u}^{opt'}(\bar{\mathbf{d}}) \in \mathbb{R}^{n_{\mathbf{u}'}}$ denotes the nominal optimal value of the unconstrained degrees of freedom, respectively. Then problem (2) is rewritten as

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

3. Self-optimizing control

3.1. The loss concept

We here discuss the implementation of the solution for the unconstrained optimization problem (5). The term self-optimizing control refers to a procedure for selecting the controlled variables $\mathbf{c} \in \mathbb{R}^{n_{\mathbf{u}}}$, which are controlled by a feedback controller (Figure 1). The focus is set on selecting a good set of controlled variables

$$\mathbf{c} = \mathbf{H}\mathbf{y}_m, \tag{6}$$

such that the operating cost $J(\mathbf{u}, \mathbf{d})$ is minimized. Here $\mathbf{H} \in \mathbb{R}^{n_{\mathbf{u}} \times n_{\mathbf{y}}}$, and the measured quantities are defined as

$$\mathbf{y}_m = \mathbf{y} + \mathbf{n}^{\mathbf{y}} \tag{7}$$

where $\mathbf{y} \in \mathbf{R}^{n_{\mathbf{y}}}$ denotes the measurement vector and $\mathbf{n}^{\mathbf{y}} \in \mathbb{R}^{n_{\mathbf{n}^{\mathbf{y}}}}$ denotes the noise vector. If the controller has integral action, then $\mathbf{c} = \mathbf{c}_s$ at steady

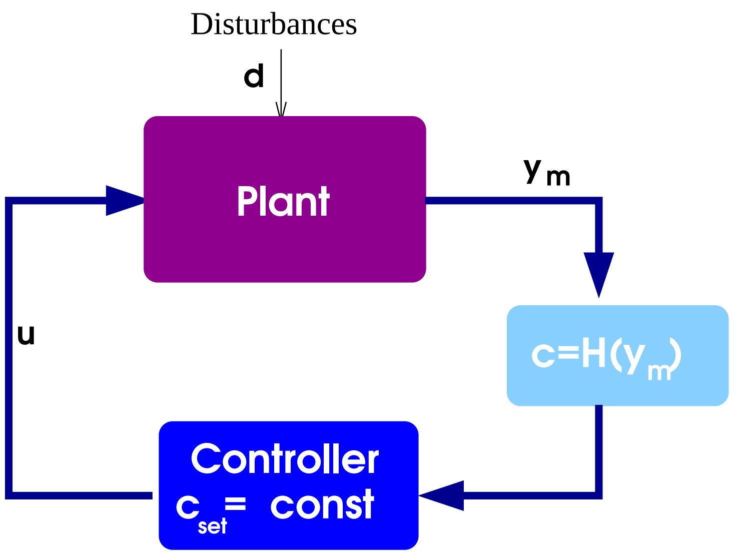


Figure 1: Block diagram self-optimizing control (SOC)

state. In the case of single measurements, each row of $\mathbf{H}$ contains only one entry, whereas if combinations of measurements are allowed, $\mathbf{H}$ will be a full matrix.

The criterion for evaluating different candidates for controlled variables is the loss from optimality, which is defined as the difference between the actual cost $J(\mathbf{u},\mathbf{d})$, which is caused by using the input $\mathbf{u}$ imposed by the selected control structure, and the cost which would have been obtained using the optimal input $\mathbf{u}^{opt}(\mathbf{d})$:

$$L = J(\mathbf{u},\mathbf{d}) - J(\mathbf{u}^{opt}(\mathbf{d}),\mathbf{d}), \tag{8}$$

The goal is to select a control structure which is self-optimizing [5]:

> "Self-optimizing control is when we can achieve an acceptable loss with constant setpoint values for the controlled variables (without the need to re-optimize when disturbances occur)."

The ideal self-optimizing variable would be the gradient $\mathbf{c} = \mathbf{J_u}(\mathbf{u},\mathbf{d}) = \frac{\partial J(\mathbf{u},\mathbf{d})}{\partial \mathbf{u}}$, which should be zero for optimal operation under all disturbances. This was already formulated in [9], where it is written:

"...Thus the search is now reduced to find some measurement function $h(\mathbf{u},\mathbf{d})$ with these required properties. An example of this kind of ideal measurement function is in fact the gradient of the criterion function."

The idea has been also mentioned in [10], where the authors wrote of the gradient as an ideal controlled variable. The gradient $\mathbf{J_u}$ satisfies the conditions of not being at a constraint, and the optimal value (zero) does not vary with changing disturbances. Controlling such invariants, in particular the gradient of a process, has also been proposed by other authors, e.g. [11] and [12].

3.2. Relationship between the gradient and the loss

Several methods for finding self-optimizing variables have been reported in the literature [13, 8, 14] and [3]. All these methods are based on a local approximation of J by a second order Taylor series [13]. Around the nominal optimum ($\mathbf{J_u}(\mathbf{u}^{opt}(\bar{\mathbf{d}}),\bar{\mathbf{d}}) = 0$), the solution of problem (5) can then be approximated by solving ([13]):

$$\min_{\mathbf{u}} [\mathbf{u}^{\mathrm{T}} \quad \mathbf{d}^{\mathrm{T}}] \begin{bmatrix} \mathbf{J_{uu}} & \mathbf{J_{ud}} \\ \mathbf{J_{du}} & \mathbf{J_{dd}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix}. \tag{9}$$

Here, $\mathbf{J_{uu}} = \frac{\partial^2 J}{\partial \mathbf{u}^2}$, $\mathbf{J_{ud}} = \mathbf{J}_{\mathbf{du}}^{\mathrm{T}} = \frac{\partial^2 J}{\partial \mathbf{u} \partial \mathbf{d}}$, and $\mathbf{J_{dd}} = \frac{\partial^2 J}{\partial \mathbf{d}^2}$. It is further assumed that $\mathbf{J_{uu}}$ is positive definite.

Locally, the relationship between the gradient $\mathbf{J_u}$ and the loss L is given by following theorem:

Theorem 1. *For a quadratic approximation* (9) *of the optimization problem* (5) *around the optimum, the relationship between the loss in* (8) *and the gradient* $\mathbf{J_u} = \frac{\partial J(\mathbf{u},\mathbf{d})}{\partial \mathbf{u}}$*, is*

$$L = \frac{1}{2}\mathbf{J_u}^{\mathrm{T}}\mathbf{J}_{\mathbf{uu}}^{-1}\mathbf{J_u} = \frac{1}{2}\left|\left|\mathbf{J}_{\mathbf{uu}}^{-\frac{1}{2}}\mathbf{J_u}\right|\right|_2^2. \tag{10}$$

Proof. See Appendix A. □

This shows, as expected, that the loss L is zero when $\mathbf{J_u} = 0$. However, for a non-zero gradient, which will occur in practice, we note that the Hessian $\mathbf{J_{uu}}$ also affects the loss. While NCO tracking methods focus on making $\mathbf{J_u}$ as close to zero as possible (which corresponds to minimizing $||\mathbf{J_u}||_*$, where $||\cdot||_*$ denotes a suitable norm, e.g. the 2-norm), self-optimizing control focuses on minimizing the loss L, which is a *weighted* norm of $\mathbf{J_u}$. Thus, the approaches are different. From an operational point of view, it is more correct to minimize the loss L.

Another issue is that in most cases the gradient cannot be measured, for example, because it is a function of the unknown disturbances $\mathbf{d}$. The concept of self-optimizing control [5] includes the special case of gradient control $\mathbf{c} = \mathbf{J_u}$, while leaving room for "suboptimal cases", $\mathbf{c} = \mathbf{Hy}$, in which the gradient cannot be determined exactly from measurements. In some cases it might be desirable to control only single measurements, or to exclude a set of measurements. Then the gradient will not be zero and the loss L provides an objective selection criterion. In other words, a self-optimizing control structure may be considered to be an acceptable (in terms of the loss L) approximation to the unmeasured gradient $J_\mathbf{u}$ using the available measurements.

In addition, a linear measurement model

$$\mathbf{y} = \mathbf{G^y u} + \mathbf{G^y_d d} \tag{11}$$

is used, where $\mathbf{y}$ is the $n_\mathbf{y}$-dimensional measurement vector, and the matrices $\mathbf{G^y} \in \mathbb{R}^{n_\mathbf{y} \times n_\mathbf{u}}$ and $\mathbf{G^y_d} \in \mathbb{R}^{n_\mathbf{y} \times n_\mathbf{d}}$ are the gain matrices from the inputs and the disturbances, respectively, to the outputs.

3.3. Self-optimizing control using the null space method

Theorem 2 (Null space method, [8]). *Assume that optimal operation corresponds to minimizing the cost*

$$J = [\mathbf{u}^\mathrm{T}\, \mathbf{d}^\mathrm{T}] \begin{bmatrix} \mathbf{J_{uu}} & \mathbf{J_{ud}} \\ \mathbf{J_{du}} & \mathbf{J_{dd}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix}, \tag{12}$$

where $\mathbf{u} \in \mathbb{R}^{n_\mathbf{u}}$ *are the available degrees of freedom, and* $\mathbf{d} \in \mathbb{R}^{n_\mathbf{d}}$ *are parametric disturbances. In addition, we assume noise free measurements*[1] $\mathbf{y} = \mathbf{G^y u} + \mathbf{G^y_d d}$. *If there exist* $n_\mathbf{y} \geq n_\mathbf{u} + n_\mathbf{d}$ *independent measurements (independent here means that* $\tilde{\mathbf{G}}^\mathbf{y} = [\mathbf{G^y}\, \mathbf{G^y_d}]$ *has full row rank), then there*

[1]Note that we include the degrees of freedom $\mathbf{u}$ in the measurement vector $\mathbf{y}$. The reason for this is that for steady state optimization, the inputs are not clearly defined, and it makes more sense to speak about degrees of freedom. The choice of which variable to select as an actual input is done when implementing the control structure.

exist $n_{\mathbf{u}}$ linear variable combinations $\mathbf{c} = \mathbf{H}\mathbf{y}$, $\mathbf{c} \in \mathbb{R}^{n_{\mathbf{u}}}$, which are invariant to disturbances $\mathbf{d}$. $\mathbf{H}$ is selected such that

$$\mathbf{H}\mathbf{F} = 0, \tag{13}$$

where

$$\mathbf{F} = \frac{\partial \mathbf{y}^{\mathrm{opt}}}{\partial \mathbf{d}}. \tag{14}$$

Controlling $\mathbf{c} = \mathbf{H}\mathbf{y}$ to zero yields optimal operation with zero loss, that is the optimal value of $\mathbf{c}$ is independent of $\mathbf{d}$.

For a proof, we refer to [8]. In Appendix B we show that choosing $\mathbf{H}$ in the left null space of $\mathbf{F}$ is indeed identical to selecting $\mathbf{c} = J_{\mathbf{u}}$, where $J_{\mathbf{u}} = \partial J/\partial \mathbf{u}$ is the gradient of (12). However, when the measurements are corrupted by biased noise $\mathbf{n}^{\mathbf{y}}$, the null space method will not give the optimal self-optimizing variable. To find the best controlled variable with noise, we refer to [3].

To obtain the optimal sensitivity matrix $\mathbf{F}$, there are several possibilities. It can be obtained numerically by re-optimization of a process model, or calculated using

$$\mathbf{F} = -\mathbf{G}^{\mathbf{y}}\mathbf{J}_{\mathbf{uu}}^{-1}\mathbf{J}_{\mathbf{ud}} + \mathbf{G}_{\mathbf{d}}^{\mathbf{y}}, \tag{15}$$

where $\mathbf{J}_{\mathbf{uu}}$ and $\mathbf{J}_{\mathbf{ud}}$ are obtained from a model, and $\mathbf{G}^{\mathbf{y}}$ and $\mathbf{G}_{\mathbf{d}}^{\mathbf{y}}$ are defined as in (11). Alternatively, one can run experiments, or use optimal measurement data as shown in [15].

4. NCO tracking

4.1. The optimality conditions concept

Necessary conditions of optimality (NCO) tracking is a general framework that turns a (dynamic or static) optimization problem into a control problem. It uses the fact that at the optimal operating point, the first order necessary optimality conditions must hold, e.g. $\mathbf{J}_{\mathbf{u}} = 0$ for an unconstrained problem. Basically, the necessary conditions of optimality are the controlled variables, $\mathbf{c} = \mathbf{J}_{\mathbf{u}}$. This general concept has been applied both to dynamic optimization problems (e.g. [16, 17, 18] and [19]), and static optimization (e.g. [1, 20] and [21]).

For steady state optimization, the Karush, Kuhn Tucker conditions [22] represent the optimality conditions. If the sensitivities are available as online

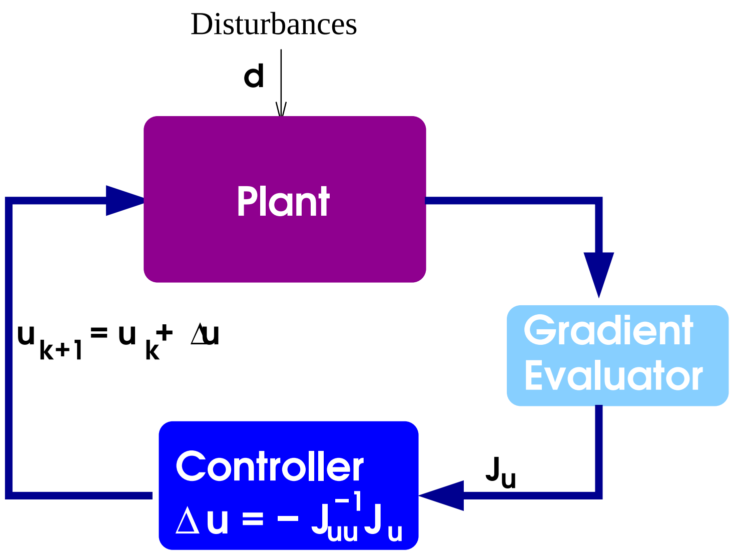


Figure 2: Block diagram NCO tracking

measurements (or estimates), they may be controlled by using a continuous feedback controller, such as a PI controller. Alternatively, the inputs may be updated iteratively until the NCO are satisfied. To the authors' knowledge, previous publications on steady state optimization using NCO tracking ([1, 20, 21]) have been applying discrete input updates for iteratively approaching the steady state optimal input value which satisfies the NCO (at least for the unconstrained part of the NCO).

4.2. NCO tracking procedure as described by [1]

In this paper we refer to "NCO tracking" as described in [1]. This is a truly measurement based optimization method, which does not rely on any process model. Instead of controlling "normal" measurements $\mathbf{y}$, the gradient is measured (or estimated), and used as a controlled variable. When a disturbance enters the process, the NCO tracking control scheme adapts the inputs iteratively such that the NCO are satisfied after some iterations. The block diagram is given in Figure 2.

We do not present the general NCO tracking procedure (with constraints) here, but we rather give a derivation of the special case without constraints, i.e. only the sensitivity seeking directions, as applied in e.g. [20]. Assume that optimal operation corresponds to

$$\min_{\mathbf{u}} J(\mathbf{u}, \mathbf{d}). \tag{16}$$

Omitting to write the explicit dependence on $\mathbf{d}$, the first order necessary condition for optimality is:

$$\mathbf{J_u}(\mathbf{u}) = \frac{\partial J(\mathbf{u})}{\partial \mathbf{u}} = 0. \tag{17}$$

To achieve optimal operation, we update the input $\mathbf{u}$ at each sample time k using the update equation

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \Delta\mathbf{u}, \tag{18}$$

until (17) is satisfied. To obtain the update term $\Delta\mathbf{u}$, we linearize (17) around the current operating point $\mathbf{u}_k$,

$$\mathbf{J_u}(\mathbf{u}_k + \Delta\mathbf{u}) = \mathbf{J_u}(\mathbf{u}_k) + \mathbf{J_{uu}}(\mathbf{u}_k)\Delta\mathbf{u}. \tag{19}$$

Since we want the update $\Delta\mathbf{u}$ to force the sensitivity to zero, we set the left hand side of (19) to zero and solve for $\Delta\mathbf{u}$ [1].

$$\Delta\mathbf{u} = -\mathbf{J}_{\mathbf{uu}}^{-1}(\mathbf{u}_k)\mathbf{J_u}(\mathbf{u}_k) \tag{20}$$

This Newton update step is exact for a quadratic approximation of the system (16), in the sense that the NCO (17) are satisfied after one iteration. In practice we do not apply the full update step $\Delta\mathbf{u}$, because this may lead to feasibility and convergence problems as the process can move outside the region where the quadratic approximation is valid. To avoid this, the update term $\Delta\mathbf{u}$ is multiplied by some tuning parameter $\beta \in [0\,1]$, such that $\mathbf{u}_{k+1} = \mathbf{u}_k + \beta\Delta\mathbf{u}$.

To evaluate (20) we need the derivative $\mathbf{J_u}(\mathbf{u}_k)$ for a given input $\mathbf{u}_k$. In this work it is chosen to make a small perturbation in the input and to run the process for a given time to estimate the gradient by finite differences. The magnitude of the perturbation is desired to be small in order not to upset the process excessively. At the same time it has to be larger than the process noise to yield sufficient information about the descent direction.

Since the Hessian $\mathbf{J_{uu}}(\mathbf{u}_k)$ is difficult to obtain, it is often determined once at the nominal operating point. Alternatively, as we choose to do in

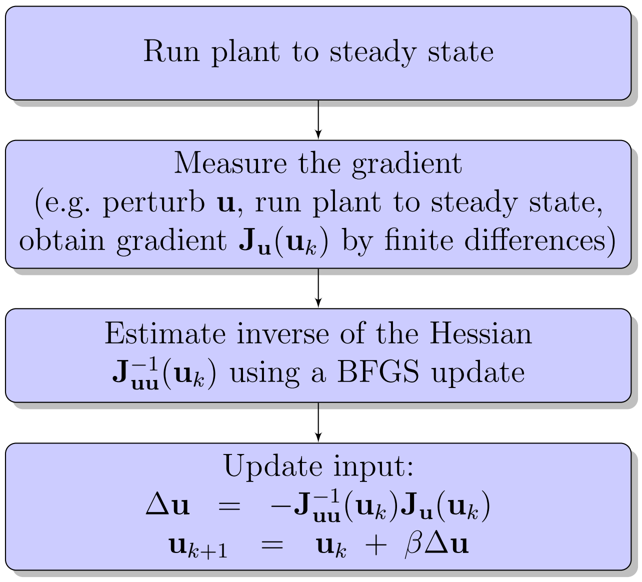

Run plant to steady state
Measure the gradient
(e.g. perturb $\mathbf{u}$, run plant to steady state, obtain gradient $\mathbf{J_u}(\mathbf{u}_k)$ by finite differences)
Estimate inverse of the Hessian
$\mathbf{J}_{\mathbf{uu}}^{-1}(\mathbf{u}_k)$ using a BFGS update
Update input:
$\Delta\mathbf{u} = -\mathbf{J}_{\mathbf{uu}}^{-1}(\mathbf{u}_k)\mathbf{J_u}(\mathbf{u}_k)$
$\mathbf{u}_{k+1} = \mathbf{u}_k + \beta\Delta\mathbf{u}$

Figure 3: Simple NCO tracking procedure ([1, 20])

this work, an approximation of the inverse of the Hessian can be obtained by a BFGS update scheme [23]. The NCO tracking algorithm is summarized in Figure 3. This procedure is analog to a Newton(like) method in optimization. In the analogy, the steady state operating periods correspond to function evaluations in the Newton procedure, and the solution is found when the NCO hold.

Just like any (quasi) Newton method, NCO tracking depends crucially on the availability of good gradient estimates. Beside estimating the gradients using input perturbations and finite differences, there exist other methods which do not require frequent perturbations. In [24] past inputs are used in Broyden's formula to obtain the gradients. Other methods which do not rely on input perturbations are described in [25] and [26]. However, in this work, the authors choose to use finite differences because of its simplicity. Avoiding input perturbations for gradient estimation will result in less nervous process operation. However, inputs will still be updated iteratively, but only at given sample times.

Remark 1. *The idea of NCO tracking [1] has been extended to the case where the gradient estimate is based on output feedback in combination with a process model. Gros and co-workers [21] use a linear measurement model* *to eliminate the disturbance from the gradient expression. Their results are based on the same idea as the null space method, which uses a measurement model to eliminate unknown disturbances and internal states from the gradient expression. However, [21] determines directly the actual change in the inputs, whereas in self-optimizing control, the generation of the inputs* $\mathbf{u}$ *is done by the feedback controller.*

The authors of [21] consider zero mean noise, and show that if the model is invertible and the number of unknowns $n_{\mathbf{d}}$ *is less than or equal to the number of measurements* $n_{\mathbf{y}'}$ *(where the degrees of freedom (inputs* $\mathbf{u}$*) are not included in the measurement vector* $\mathbf{y}'$*), the inputs can be updated to converge to the optimum. This is the same result as found with the null-space method [8], where it is required that* $n_{\mathbf{y}} \geq n_{\mathbf{u}} + n_{\mathbf{d}}$*, but where the degrees of freedom* $\mathbf{u}$ *are included in the measurement vector,* $\mathbf{y} = [\mathbf{y}', \mathbf{u}]^{\mathrm{T}}$*, such that* $n_{\mathbf{y}} = n_{\mathbf{y}'} + n_{\mathbf{u}}$*.*

In the case of biased noise, neither the null space method nor the NCO tracking modifications introduced by [21] will give the best achievable operation. In this case it is necessary to use other methods which find a trade-off between the loss caused by the disturbance and the loss caused by the measurement offset. A method applicable in this case is the "minimum loss method" [3].

5. Self-optimizing control and NCO tracking [1] combined

In order to suggest how to combine self-optimizing control as described in [8] and NCO tracking as in [1], we first consider how a chemical plant is usually operated today.

5.1. Time scale separation of the overall control system

The control structure of a complete chemical plant can be decomposed vertically into different layers, which operate on different time scales, [27, 28]. Each control layer implements the setpoints which are given from the layer above, Figure 4.

The top layer consists of planning and scheduling. This includes management decisions on e.g. the product specifications, and on profit and safety parameters, such as utility prices and constraints. Usually, this layer has a time scale of weeks or days, strongly depending on the type of process and the production scale.

The optimization layer is located below the planning and scheduling layer and implements the goals given from the planning and scheduling layer. In

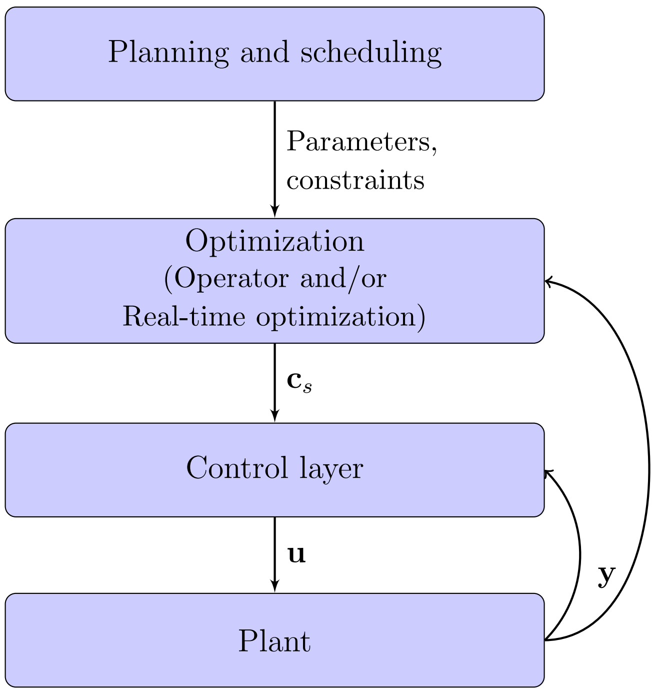


Figure 4: Vertical decomposition of the control structure

most plants this is done by operators, but in recent years online optimization (RTO) has been increasingly used to find setpoints for the controlled variables of the lower layers. However, this can become complicated as it involves several difficult steps such as steady state detection, data estimation and reconciliation, and solving a large nonlinear optimization problem. Once the optimization problem has been solved successfully or the operators have decided to change the setpoints, the new setpoints are passed on to the control layer and implemented. It is typical for this layer that the setpoints are updated at discrete time instances with update intervals in the time scale of several hours.

The control layer below the optimization layer generally consists of model predictive controllers (MPC) or PID controllers, which act on the plant inputs $\mathbf{u}$. This layer has a time scale ranging from fractions of seconds up to minutes and to a few hours. Finally, the plant layer contains the actual plant, but usually with some stabilizing (regulatory) control loops.

When a disturbance enters the system, the control layer will try to keep the setpoints of the controlled variables $\mathbf{c}$ at their original setpoints. After the plant has settled and a (suboptimal) steady state has been reached (and detected), the operator may adjust the setpoints based on experience, or the real-time optimizer may re-calculate the setpoints. Then the setpoints of the controlled variables are ramped to their new values, and the plant is allowed to settle again. The long time delay between start of the disturbance and reaching the final optimized operation point is one of the problems for the optimization layer. In particular if RTO is used, it is not possible to counteract disturbances which occur on a fast time scale [29]. This limits successful RTO applications to cases with sporadic disturbances, which, after a short transition period, lead to a new steady state, e.g. step changes in the plant throughput or the like. Disturbances occurring on a faster time scale cannot be detected and rejected in RTO implemented as described above.

Using a dynamic model in the real-time optimization layer with an economic objective function would allow setpoint changes without having to wait for steady state. However, practical obstacles have prevented dynamic RTO (DRTO) from becoming a standard tool in the process industries. The main problems arise from the reliability of the information used in the DRTO, because good models are difficult to obtain and maintain with justifiable efforts. In addition, the state estimation causes additional challenges. Even if a good model and states are available in the DRTO, the dynamic optimization problem itself is difficult to solve.

Table 1: Summary of properties

Null space method [8]	NCO tracking [1]
Offline analysis to find $\mathbf{c} = \mathbf{Hy}$	Controlled variable: $\mathbf{c} = \mathbf{J_u}$
$\mathbf{J_u}$ not measured	$\mathbf{J_u}$ measured (or estimated)
Set of important $\mathbf{d}$ assumed a priori	No assumptions on disturbances
$\mathbf{c} = \mathbf{Hy}$ obtained from model	No model needed
Input generated by feedback	Input computed explicitly
Near-optimal for expected disturbances	Optimal for unexpected disturbance
Linearized at nom. point	Linearization point moves
Fast (feedback)	Slow (acts only at sample times)

5.2. Properties of self-optimizing control and NCO tracking

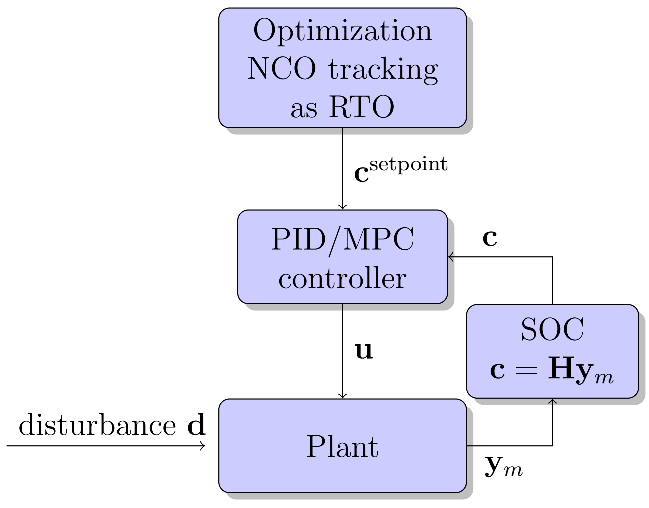
Both methods, NCO tracking and self-optimizing control, pursue the same goal, namely minimization of the operating cost. The main difference is that in unconstrained NCO tracking [1] the gradient is "measured"online, and the inputs are explicitly manipulated.

On the other hand, in self-optimizing control, a model is used offline to find controlled variables $\mathbf{c} = \mathbf{Hy}$ which do not need frequent updates. One example would be the gradient, $\mathbf{c} = \mathbf{J_u}$, but since the gradient is usually not available as a measurement, self-optimizing control does not in general aim for controlling the gradient to zero, but instead aims for finding controlled variables, which give a small loss when the inputs are adjusted to keep $\mathbf{c}$ constant.

In summary, we may say that the NCO tracking procedure [1] measures the gradient, and calculates explicitly the required (steady state) input change $\Delta\mathbf{u}$, whereas in self-optimizing control [8] the inputs, are generated implicitly by the controllers to keep $\mathbf{c} = \mathbf{Hy}$ constant. In Table 1 we have listed the main differences between the null space method and NCO tracking [1].

5.3. Using self-optimizing control and NCO tracking together

The previous observations lead us to consider self-optimizing control and NCO tracking as described by [1] as *complementary*, and use them together.



Optimization NCO tracking as RTO
$\mathbf{c}^{\text{setpoint}}$
PID/MPC controller
$\mathbf{c}$
SOC
$\mathbf{c} = \mathbf{H}\mathbf{y}_m$
$\mathbf{u}$
disturbance $\mathbf{d}$
Plant
$\mathbf{y}_m$

Figure 5: Combining NCO tracking and self-optimizing control

The NCO tracking [1] fits better into the optimization layer as an alternative to model based real-time optimization (RTO), while self-optimizing control should be used to identify the controlled variable $\mathbf{c}$ for the lower layer, as shown in Figure 5. This is a cascade structure, where the NCO tracking controller updates the setpoints for the self-optimizing variable $\mathbf{c}$.

It may be argued that if NCO tracking or an RTO system is installed, there is no need to select a self-optimizing control structure because the setpoints are updated by the optimization layer. However, a combination of an RTO layer (or NCO tracking) and self-optimizing control avoids the shortcomings of conventional RTO:

1. The use of self-optimizing controlled variables enables a faster optimal reaction to expected (main) disturbances; not only at sample times.
2. The RTO has to change the setpoints less frequently and the sytem will be more robust.

Infrequent RTO updates result in fewer complex operations such as steady state detection, data reconciliation, and solving the resulting nonlinear optimization problems. At the same time, the self-optimizing control structure can benefit significantly from an RTO system or NCO tracking controller on top of it. One reason is that self-optimizing control is based on a model and can only handle expected (modelled) disturbances, whereas NCO tracking

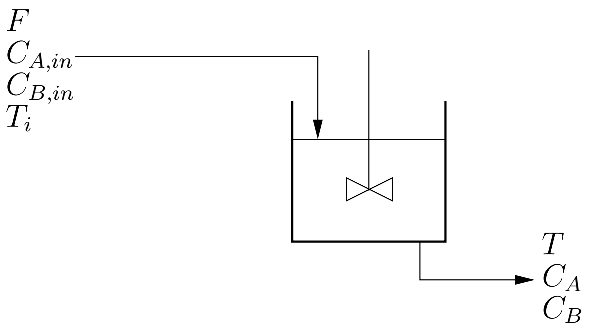


Figure 6: Schematic diagram of a CSTR

can also handle unknown disturbances. Another reason is that if a disturbance moves the process far from the linearization point, the local model approximation used for finding the self-optimizing variables may be poor. Therefore, the self-optimizing control structure cannot reject disturbances which move the process far away from the linearization point. They have to be counteracted by re-optimization of the system.

In summary, it is recommended to always use a self-optimizing control layer below the optimization layer instead of directly computing the plant input **u**. This rejects known disturbances on a fast timescale, while the unexpected disturbances can be rejected by the NCO tracking/RTO layer updates. Applying self-optimizing control is thus a robust way to implement the control layer below the RTO layer.

6. Case study

6.1. Model

To illustrate the ideas above, we present simulation results for a dynamic CSTR with a feed stream F containing mainly the component A, and a reversible chemical reaction $A \rightleftharpoons B$, see Figure 6. The process model is taken from [30], and the dynamics of the system are described by following set of equations,

$$\begin{aligned}
\frac{dC_A}{dt} &= \frac{1}{\tau}(C_{A,in} - C_A) - r, && (21)\\
\frac{dC_B}{dt} &= \frac{1}{\tau}(C_{B,in} - C_B) + r, && (22)\\
\frac{dT}{dt} &= \frac{1}{\tau}(T_i - T) + \frac{-\Delta H_{rx}}{\rho c_p} r, && (23)
\end{aligned}$$

where C_A, C_B, T, and T_i denote the concentrations of components A and B, the reactor temperature and the feed inlet temperature, respectively. Further, τ is the residence time, ρ is the density, c_p is the heat capacity, and $-\Delta H_{rx}$ is the reaction enthalpy. The reaction rate r is defined by

$$r = k_1 C_A - k_2 C_B \tag{24}$$

where

$$k_1 = A_1 e^{\frac{-E_1}{RT}} \quad \text{and} \quad k_2 = A_2 e^{\frac{-E_2}{RT}}, \tag{25}$$

and A_1 and A_2 are the Arrhenius factors for the reaction constants k_1 and k_2.

This process has one manipulated input (u), the inlet temperature T_i. The expected disturbances d_1 and d_2 are variations in the feed concentrations $C_{A,in}$ and $C_{B,in}$, and the measured variables are

$$\begin{aligned}
y_1 &= C_A,\\
y_2 &= C_B,\\
y_3 &= T.
\end{aligned} \tag{26}$$

The objective is to maximize the profit function, which is the difference between the income from selling the product B and the cost for heating the feed ([31]):

$$-J = P = [p_{C_B} C_B - (p_{T_i} T_i)^2], \tag{27}$$

Here p_{C_B} is the price of the desired product B, and p_{T_i} is the cost for heating. The parameter values are given in Table 2, and the nominal operation values for all variables are listed in Table 3.

Table 2: Objective function parameters (prices)

Parameter	Value
p_{C_B}	2.009
p_{T_i}	$1.657 \cdot 10^{-3}$

Table 3: Nominal values for the CSTR model

Variable	Value	Unit	Description
T_i	424.20	K	Feed temperature (input u)
C_A	0.4978	mol/l	Concentration A in product (y_1)
C_B	0.5022	mol/l	Concentration B in product (y_2)
T	426.71	K	Reactor temperature (y_3)
$C_{A,in}$	1.000	mol/l	Concentration A in feed, (d_1)
$C_{B,in}$	0.000	mol/l	Concentration B in feed, (d_2)
F	1.000	holdup min^{-1}	Flow rate
A_1	5000	s^{-1}	Arrhenius factor 1
A_2	$1 \cdot 10^6$	s^{-1}	Arrhenius factor 2
c_p	1000	cal kg^{-1}K^{-1}	Heat capacity
E_1	10000	cal mol^{-1}	Activation energy 1
E_2	15000	cal mol^{-1}	Activation energy 2 (d_3)
R	1.987	cal mol^{-1}K^{-1}	Ideal gas constant
$-\Delta H_{rx}$	5000	cal mol^{-1}	Heat of reaction
ρ	1.000	kg/l	Density
τ	1.000	min	Residence time

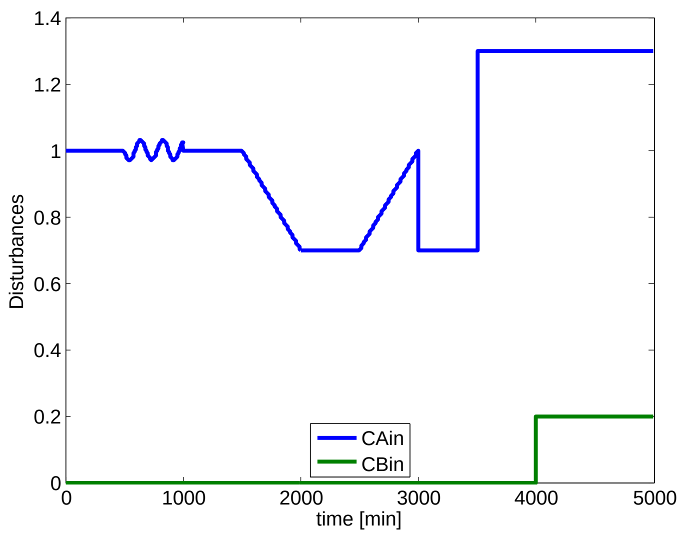


Figure 7: Disturbance trajectories $C_{A,in}, C_{b,in}$

6.2. *Simulations*

First, we control the process for the expected disturbances using direct NCO tracking [1]. Next, compare it with self-optimizing control with $\mathbf{c} = \mathbf{Hy}$ obtained using the null-space method. After comparing the individual methods, for an unexpected disturbance, we finally combine the methods as shown in Figure 5.

The expected disturbance scenario is given in Figure 7. After 500 minutes at the nominal value, the concentration $C_{A,in}$ (d_1) varies sinusoidally for 500 minutes before returning to its nominal value. Then ramp disturbances in $C_{A,in}$ are introduced, followed by large step disturbances. At 4000 minutes, the concentration $C_{B,in}$ (d_2) makes a step change of 0.2 mol/l. The non-steady state periods (sinusoid and ramp) are included to test how the controller behaves in these cases. Note that strictly speaking, the gradient is not defined when the process is not at steady state.

6.2.1. *Direct NCO tracking*

To obtain the gradient information, the input T_i is perturbed with a step of 1 K. Starting with a positive value, the sign is altered every fourth NCO iteration. Changing the sign of the perturbation was found to give better overall performance of the NCO procedure. No steady state detection is

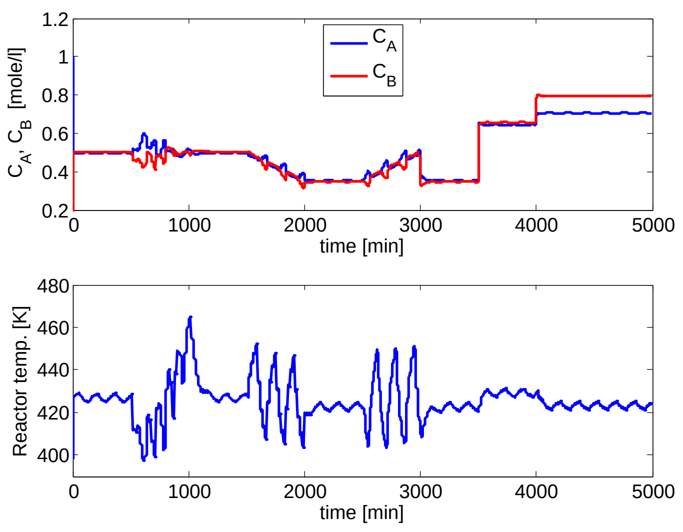


Figure 8: NCO tracking, concentrations and temperature

implemented in the NCO tracking procedure. Instead, a step test is used to determine the approximate time for the system to settle down to a new steady state. At the nominal point, the system has a time constant of less than two minutes for an input step of $\Delta T_i = 5$ K. To let the system settle down far from the nominal point, where the system dynamics are different, a sample time of 10 minutes is chosen for the direct NCO tracking procedure. The step size parameter β is set to 0.4.

Figure 8 shows the concentration and temperature trajectories for the NCO tracking procedure. The control strategy enables acceptable control for the steady state periods and the step disturbances. Since the method assumes steady state after 10 minutes, and uses the results at each sample time for calculating the input update, it has difficulties handling sinusoidal and ramp disturbances which do not give a steady state. However, the controller manages to keep the system stable during these periods. The performance of the NCO tracking algorithm is very sensitive to the tuning parameter β, the sample time, and the timing and kind of disturbance, and of course is sensitive to the size of perturbation for estimating the gradient.

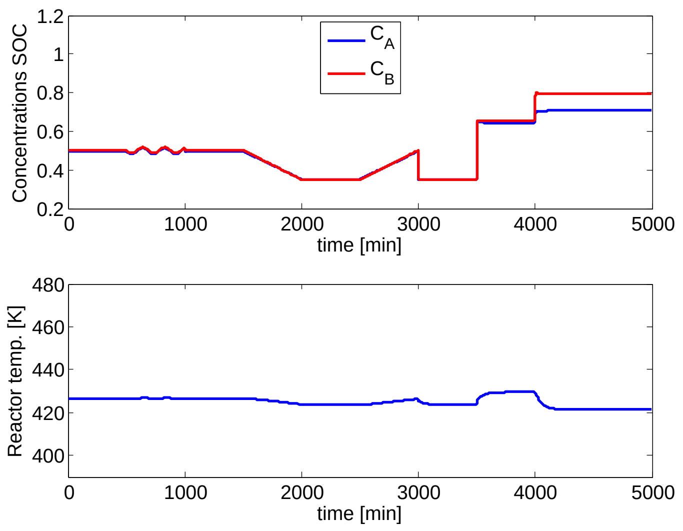


Figure 9: Self-optimizing control (SOC), concentrations and temperature

6.2.2. Self-optimizing control using the null space method

Next, we obtain the self-optimizing controlled variable $\mathbf{c} = \mathbf{H}\mathbf{y}$ using the null space method from Section 3. Since we have one input and 2 disturbances to compensate for, we need three measurements for the invariant variable combination, and we choose $\mathbf{y} = [\ C_A \quad C_B \quad T\ ]^{\mathrm{T}}$. We use the model offline to optimize the steady state system at the nominal operating point and then introduce small perturbations in the disturbance variables $\mathbf{d} = [C_{A,in}\, C_{B,in}]^{\mathrm{T}}$. After re-optimizing we calculate

$$\mathbf{F} = \frac{\partial \mathbf{y}^{\mathrm{opt}}}{\partial \mathbf{d}} = \begin{bmatrix} -0.4862 & -0.3223 \\ -0.5138 & -0.6777 \\ -9.9043 & 40.5807 \end{bmatrix}. \tag{28}$$

The resulting nontrivial $\mathbf{H}$ which gives $\mathbf{HF} = 0$ (null-space method) is $\mathbf{H} = [\ -0.7688 \quad 0.6394 \quad 0.0046\ ]$. Using a PI controller, the self-optimizing variable $\mathbf{c} = \mathbf{H}\mathbf{y}_m = -0.7688C_A + 0.6394C_B + 0.0046T$ is controlled at a constant setpoint (zero if we use the deviation from nominal steady state). The concentration and temperature trajectories with self-optimizing control are plotted in Figure 9. Compared with the concentrations and temperature using NCO tracking, the trajectories are much smoother.

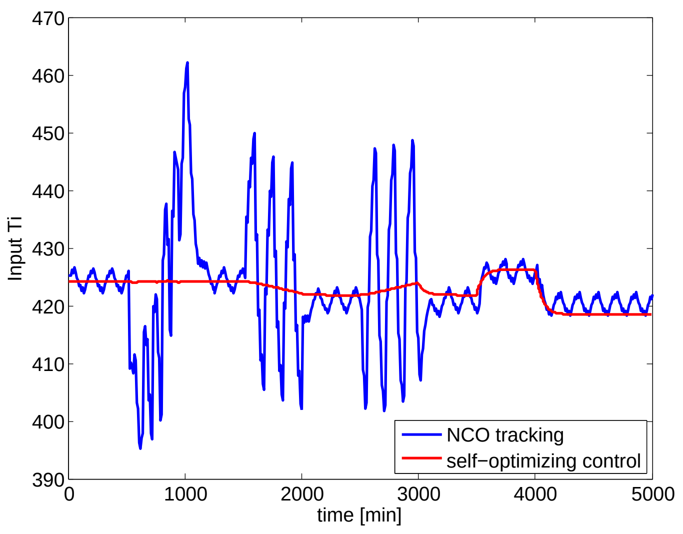


Figure 10: Input usage for SOC and NCO tracking

6.2.3. Comparing inputs and profit for NCO tracking and self-optimizing control

As may be seen from the trajectories for NCO tracking and self-optimizing control, Figure 10, the input usage for the two cases is quite different. While the NCO tracking procedure needs large input variations to estimate the gradient and to iteratively update the input, the input usage of the self-optimizing control structure is very moderate and smooth. Especially during the non-steady state disturbances NCO tracking leads to excessive input changes. This is because the effect of the input change and the effect of the disturbance cannot be distinguished when estimating the gradient.

Comparing the profits, Figure 11, shows that both methods are very similar in the steady state periods, but for disturbances where no steady state is reached within one sample time, NCO tracking is not performing as well as the self-optimizing control.

6.2.4. Using NCO tracking as RTO and self-optimizing control in the control layer

If it can be guaranteed that the disturbances in the feed concentration are the only ones entering the process, then self-optimizing control is sufficient, and an RTO layer is not necessary. However, the situation changes if there

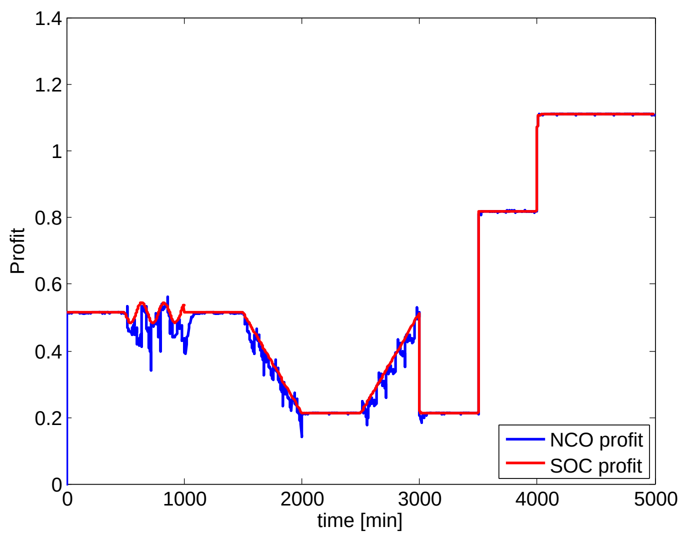


Figure 11: Profit for SOC and NCO tracking

are disturbances which were not included in the offline analysis (plant-model mismatch). Also, the nominal setpoint for the self-optimizing variables may not be correct.

As an unmodelled disturbance, consider a positive step change in the activation energy E_2 (d_3) of 3% at time 3100 min, which comes in addition to the disturbances in $C_{A,in}$ and $C_{B,in}$ in Figure 7. The unexpected disturbance reduces the reaction rate for the reverse reaction, especially at higher temperatures. Comparing the profits using the two control structures, Figure 12, we see that the self-optimizing control system cannot make use of the improved conditions caused by the unexpected disturbance.

A combined approach, where the self-optimizing control setpoints are adapted using RTO or NCO tracking can solve this problem, and at the same time reduce RTO or NCO tracking sample time. In Figure 13 the instantaneous profit for pure NCO tracking [1] (sample time: 10 min) is compared to the combined system with a sample time of 25 min. The combined system gives a similar performance in terms of the profit, and more importantly, we see from Figure 14 that the combined system gives a substantially smoother input action than direct NCO tracking. Using online RTO, the performance could be improved even further because the setpoints would move directly to the optimal values instead of iteratively approaching them. However, un-

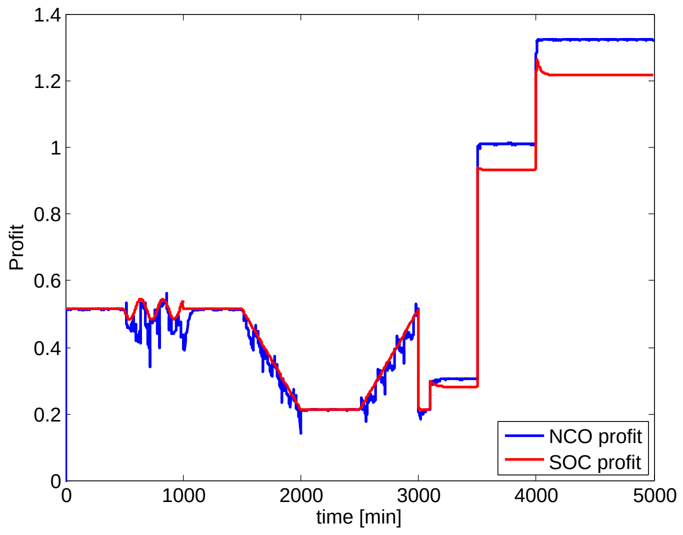


Figure 12: Profits for NCO tracking and SOC with unexpected disturbance (d_3) at 3100 min

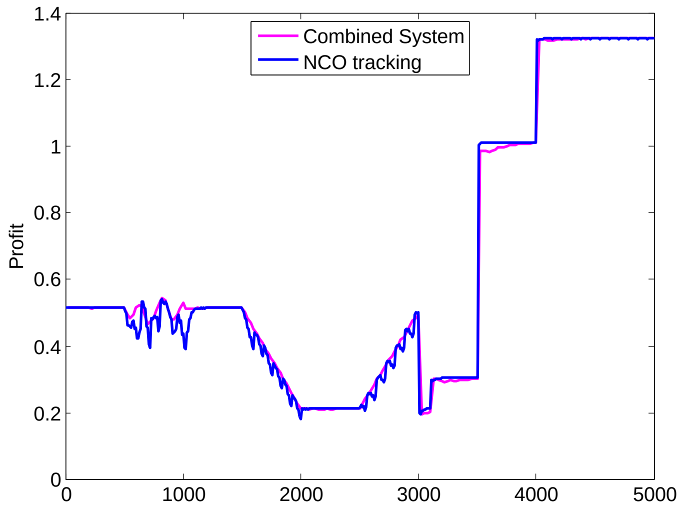


Figure 13: Profits for combined SOC/NCO tracking (25 min sample time) and pure NCO tracking (10 min sample time)

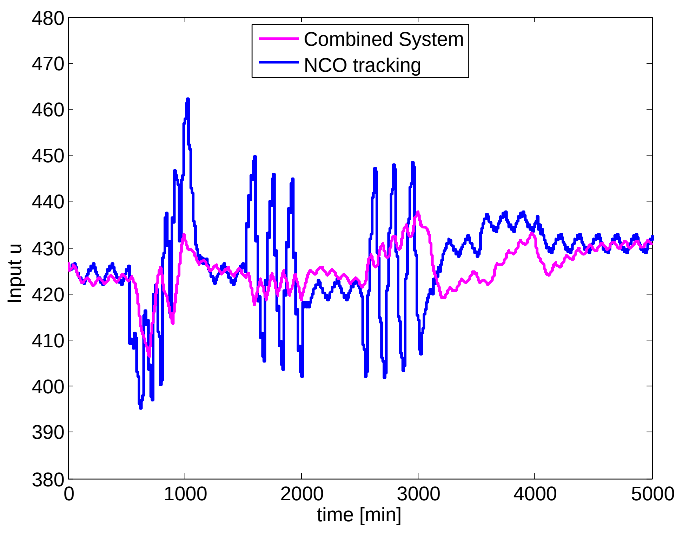


Figure 14: Input, combined NCO/SOC and direct NCO tracking

modelled (unexpected) disturbances are not rejected in online RTO either, so in this case (unmodelled disturbances) a method like NCO tracking would be required.

7. Discussion

There has been some confusion about the relationship between the "self-optimizing" control approach of Skogestad and co-workers and the NCO tracking approach of Bonvin and co-workers. As shown in this paper, the two methods may be successfully combined by controlling the self-optimizing variables $\mathbf{c}$ in the lower layer, and letting NCO tracking adjust the setpoints, $\mathbf{c} = \mathbf{c}_{opt}$ based on online estimates of the gradient.

In the case study, it was not easy to make the NCO tracking work in spite of the fact that we assumed no measurement noise. This could be partly attributed to the fact that we used a simple finite difference procedure to obtain the gradients, without, for example, steady state detection. The NCO tracking parameters (perturbation magnitude, step size β, sample time) which converge to the optimum, were found by trial and error. Parameters which perform well for one disturbance may give poor performance for a different disturbance. Non-steady state periods make it especially difficult to find parameters which optimize the cost with acceptable input usage.

The NCO tracking procedure hinges on good gradient estimates, and using finite differences for estimating the gradient gives poor NCO tracking updates, even with the assumptions of perfect measurements without noise, and perfect knowledge of the profit value. More advanced gradient estimation techniques and input adaptation methods may give better overall performance, especially in terms of input usage; because poor update steps caused by wrong gradient estimates would be avoided. In this work, we chose to apply the simple finite difference method, because our purpose is to demonstrate that the basic concepts of measurement based optimization techniques, such as NCO tracking, and the model based self-optimizing control concepts are complementary. Whatever technique for calculating the gradient and the NCO tracking updates is used, combining the two methods helps to overcome their limitations. An interesting task for future research might be to study the combination of self-optimizing control with a more advanced update/gradient estimation method and a more realistic case with non-zero mean random measurement noise.

8. Conclusion

The different characteristics of the two methods studied in this paper suggest considering them as complementary, not competing. NCO tracking is most suitable for use in the optimization layer, as an alternative to online RTO, while self-optimizing control is used for selecting CVs in the control layer.

Since almost every RTO system has a control system in the layer below, using a self-optimizing control structure in the lower layer improves performance and robustness, and can significantly reduce need for RTO updates. For NCO tracking as implemented in this paper, this means fewer perturbations for gradient estimation. For an online RTO, this means more time for complex, time intensive computations, with few compromises on performance.

The matlab simulation files are available on the home page of S. Skogestad, `http://www.nt.ntnu.no/users/skoge`, or as supplementary material from the journal.

Appendix A. Relationship between the gradient value and the loss

Proof for Theorem 1: Consider the quadratic approximation of the unconstrained optimization problem (5),

$$\min_{\mathbf{u}} [\mathbf{u}^{\mathrm{T}}\,\mathbf{d}^{\mathrm{T}}] \begin{bmatrix} \mathbf{J_{uu}} & \mathbf{J_{ud}} \\ \mathbf{J_{du}} & \mathbf{J_{dd}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix}. \tag{A.1}$$

The optimal input $\mathbf{u}^{opt}(\mathbf{d})$ is found by setting the gradient of the approximated cost function (A.1),

$$\mathbf{J_u} = \begin{bmatrix} \mathbf{J_{uu}} & \mathbf{J_{ud}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix} = \mathbf{J_{uu}u} + \mathbf{J_{ud}d}, \tag{A.2}$$

to zero and solving $\mathbf{J_u} = 0$ for the input $\mathbf{u}$. This gives

$$\mathbf{u}^{opt}(\mathbf{d}) = -\mathbf{J}_{\mathbf{uu}}^{-1}\mathbf{J_{ud}d}. \tag{A.3}$$

From [13] it is known that the loss can be written as

$$L = \frac{1}{2}(\mathbf{u} - \mathbf{u}^{opt}(\mathbf{d}))^{\mathrm{T}}\mathbf{J_{uu}}(\mathbf{u} - \mathbf{u}^{opt}(\mathbf{d})). \tag{A.4}$$

Inserting the expression for $\mathbf{u}^{opt}(\mathbf{d})$ from (A.3) into (A.4) yields (note that $\mathbf{J_{uu}}$ is symmetric):

$$\begin{aligned} L &= \frac{1}{2}(\mathbf{u} + \mathbf{J}_{\mathbf{uu}}^{-1}\mathbf{J_{ud}d})^{\mathrm{T}}\mathbf{J_{uu}}(\mathbf{u} + \mathbf{J}_{\mathbf{uu}}^{-1}\mathbf{J_{ud}d}) \\ &= \frac{1}{2}(\mathbf{u}^{\mathrm{T}} + \mathbf{d}^{\mathrm{T}}\mathbf{J}_{\mathbf{ud}}^{\mathrm{T}}\mathbf{J}_{\mathbf{uu}}^{-T})\mathbf{J_{uu}}(\mathbf{u} + \mathbf{J}_{\mathbf{uu}}^{-1}\mathbf{J_{ud}d}) \\ &= \frac{1}{2}(\mathbf{u}^{\mathrm{T}}\mathbf{J}_{\mathbf{uu}}^{\mathrm{T}} + \mathbf{d}^{\mathrm{T}}\mathbf{J}_{\mathbf{ud}}^{\mathrm{T}})\mathbf{J}_{\mathbf{uu}}^{-1}(\mathbf{J_{uu}u} + \mathbf{J_{ud}d}) \\ &= \frac{1}{2}\mathbf{J_u}^{\mathrm{T}}\mathbf{J}_{\mathbf{uu}}^{-1}\mathbf{J_u} \\ &= \frac{1}{2}\left|\left|\mathbf{J}_{\mathbf{uu}}^{-\frac{1}{2}}\mathbf{J_u}\right|\right|_2^2. \end{aligned} \tag{A.5}$$

Appendix B. Relationship between the gradient and the null space method

Consider the quadratic approximation of the unconstrained optimization problem (5),

$$\min_{\mathbf{u}} [\mathbf{u}^{\mathrm{T}}\,\mathbf{d}^{\mathrm{T}}] \begin{bmatrix} \mathbf{J_{uu}} & \mathbf{J_{ud}} \\ \mathbf{J_{du}} & \mathbf{J_{dd}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix}. \tag{B.1}$$

Differentiating the cost J with respect to $\mathbf{u}$ gives

$$\mathbf{J_u} = \left[\begin{array}{cc} \mathbf{J_{uu}} & \mathbf{J_{ud}} \end{array} \right] \left[\begin{array}{c} \mathbf{u} \\ \mathbf{d} \end{array} \right]. \tag{B.2}$$

The linear model (11) can be rewritten as

$$\mathbf{y} = \tilde{\mathbf{G}}^{\mathbf{y}} \left[\begin{array}{c} \mathbf{u} \\ \mathbf{d} \end{array} \right]. \tag{B.3}$$

If we assume that we have a sufficient number of measurements, $n_{\mathbf{y}} \geq n_{\mathbf{u}} + n_{\mathbf{d}}$, then the model may be inverted, and substitution into (B.2) gives

$$\mathbf{J_u} = \left[\begin{array}{cc} \mathbf{J_{uu}} & \mathbf{J_{ud}} \end{array} \right] [\tilde{\mathbf{G}}^{\mathbf{y}}]^{\dagger} \mathbf{y}, \tag{B.4}$$

where $(\cdot)^{\dagger}$ denotes the pseudo inverse of $(\cdot)$. At the optimum, we have $\mathbf{J_u} = 0$, or equivalently $\mathbf{c} = \mathbf{Hy} = 0$, where

$$\mathbf{H} = [\mathbf{J_{uu}} \ \ \mathbf{J_{ud}}][\tilde{\mathbf{G}}^{\mathbf{y}}]^{\dagger}. \tag{B.5}$$

This is the same expression for $\mathbf{H}$ as derived in [3]. And indeed, if we evaluate $\mathbf{HF}$ using $\mathbf{F}$ in (15), we get $\mathbf{HF} = 0$. This follows since $\mathbf{F}$ in (15) may be rewritten as

$$\mathbf{F} = \tilde{\mathbf{G}} \left[\begin{array}{c} -\mathbf{J}_{\mathbf{uu}}^{-1}\mathbf{J_{ud}} \\ \mathbf{I} \end{array} \right]. \tag{B.6}$$

Also note that the loss L and gradient are related by (Appendix A)

$$L = \frac{1}{2} \mathbf{J_u}^{\mathrm{T}} \mathbf{J}_{\mathbf{uu}}^{-1} \mathbf{J_u}, \tag{B.7}$$

so $\mathbf{J_u} = 0$ is equivalent to $L = 0$. In summary, we see that the null space method is identical to controlling the gradient, $\mathbf{J_u} = 0$.

References

[1] 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 (2005) 701 – 712.

[2] T. E. Marlin, A. Hrymak, Real-time operations optimization of continuous processes, in: Proceedings of CPC V, AIChE Symposium Series vol. 93., pp. 156–164.

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

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

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

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

[7] C. S. Draper, Y. T. Li, Principles of optimizing control systems and an application to the internal combustion engine, ASME publications 160 (1951) 1–16.

[8] V. Alstad, S. Skogestad, Null space method for selecting optimal measurement combinations as controlled variables, Industrial & Engineering Chemistry Research 46 (2007) 846–853.

[9] I. J. Halvorsen, S. Skogestad, Indirect on-line optimization through setpoint control, AIChE 1997 Annual Meeting, Los Angeles; paper 194h (1997).

[10] I. J. Halvorsen, S. Skogestad, Optimal operation of petlyuk distillation: steady-state behavior, Journal of Process Control 9 (1999) 407 – 424.

[11] D. Bonvin, B. Srinivasan, D. Ruppen, Dynamic Optimization in the Batch Chemical Industry, in: Chemical Process Control-VI (2001) 255–273.

[12] Y. Cao, Direct and indirect gradient control for static optimisation, International Journal of Automation and Computing 2 (2005) 60–66.

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

[14] V. Kariwala, Y. Cao, S. Janardhanan, Local self-optimizing control with average loss minimization, Industrial & Engineering Chemistry Research 47 (2008) 1150–1158.

[15] J. Jäschke, Invariants for Optimal Operation of Process Systems, Ph.D. thesis, Norwegian University of Science and Technology, 2011.

[16] B. Srinivasan, D. Bonvin, E. Visser, S. Palanki, Dynamic optimization of batch processes: Ii. role of measurements in handling uncertainty, Computers & Chemical Engineering 27 (2003) 27 – 44.

[17] B. Srinivasan, S. Palanki, D. Bonvin, Dynamic optimization of batch processes: I. characterization of the nominal solution, Computers & Chemical Engineering 27 (2003) 1 – 26.

[18] D. Bonvin, L. Bodizs, B. Srinivasan, Optimal grade transition for polyethylene reactors via NCO tracking, Chemical Engineering Research and Design 83 (2005) 692 – 697.

[19] J. V. Kadam, W. Marquardt, B. Srinivasan, D. Bonvin, Optimal grade transition in industrial polymerization processes via NCO tracking, AIChE Journal 53 (2007) 627–639.

[20] B. Srinivasan, L. T. Biegler, D. Bonvin, Tracking the necessary conditions of optimality with changing set of active constraints using a barrier-penalty function, Computers & Chemical Engineering 32 (2008) 572–279.

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

[22] M. S. Bazaraa, H. D. Sherali, C. M. Shetty, Nonlinear Programming: Theory and Algorithms, John Wiley & Sons, 2006.

[23] J. Nocedal, S. Wright, Numerical Optimization, Springer, 2006.

[24] P. Roberts, Broyden derivative approximation in isope optmizing and control algorithms, in: Proceedings of the 11th IFAC workshop on control applications of optimization, volume 1, pp. 283–288.

[25] M. Brdyś, P. Tajewski, An algorithm for steady-state optimising dual control of uncertain plants., in: Proceedings of the first IFAC workshop on new trends in design of control systems Smolenice, Slovakia, pp. 249–254.

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

[27] W. Findeisen, F. Nailey, M. Brdys, K. Malinowski, P. Tatjewski, A. Wozniak, Control and Coordination in Hierarchical Systems, John Wiley & Sons, 1980.

[28] S. Skogestad, Control structure design for complete chemical plants, Computers & Chemical Engineering 28 (2004) 219 – 234. Escape 12.

[29] S. Engell, Feedback control for optimal process operation, Journal of Process Control 17 (2007) 203 – 219.

[30] C. G. Economou, M. Morari, Internal model control. 5. extension to nonlinear systems, Industrial & Engineering Chemistry Process Design and Development 25 (1986) 403–411.

[31] V. Alstad, Studies on selection of controlled variables, Ph.D. thesis, Norwegian University of Science and Technology, Department of Chemical Engineering, 2005.