# 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). We argue that self-optimizing control is not an alternative to NCO tracking, 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 has to be assumed, as unexpected disturbances are not rejected in SOC. On the other hand, NCO tracking adapts the inputs at given sample times without assumptions on the set of disturbances. Disturbances with high frequencies or which 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 NCO tracking 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 a such a way that the operators set the setpoints for PID controllers to keep the controlled variables (CVs) at the desired setpoint. Which measurements are chosen as CVs is mostly decided based on process knowledge and best practices. However, due to stronger competition and environmental regulations, in recent years it has become increasingly important to operate the processes closer 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))
- Model used offline (e.g. self-optimizing control (SOC))
- Explicit Model not used (e.g. NCO tracking)

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 to determine the optimal set-points by solving an optimization problem online [1].

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 has the objective to transform the economic objectives into control objectives [2]. A process model is used to support decision making in control structure design, but it will not be used online. Self-optimizing control [3] belongs into this category.

A third strategy avoids using an explicit process model, but uses measurements in order to obtain gradient information about the process. This information is used to update the inputs directly in order to obtain optimal operation. Necessary conditions of optimality tracking (NCO tracking) [4] and extremum seeking control [5] represent this category. This idea is relatively old [6], 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 kind 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 two sections briefly describe the ideas from self-optimizing control and NCO tracking. In particular, we focus on the null-space method [7], which uses a model offline, and the NCO tracking procedure for steady state optimization [4], which uses no model at all. In Section 4 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 5, followed by a discussion in Section 6, and conclusions, Section 7.

## 2. Self-optimizing control

In virtually all practical cases, plant operation is subject to operational and safety constraints, so the problem of achieving optimal operation can be formulated as

$$\min_{\mathbf{u}} J(\mathbf{u},\mathbf{d}) \quad \text{s.t.} \quad \begin{cases} \text{plant,} \\ \text{constraints: } C_{all}(\mathbf{u},\mathbf{d}) \leq 0 \end{cases} \tag{1}$$

where $\mathbf{u}$ is the vector of adjustable input variables (e.g. a valve opening or a pump speed), $\mathbf{d}$ is a vector of unknown disturbances and parameters, and $C_{all}(\mathbf{u},\mathbf{d})$ is the vector of all equality and inequality constraints.

In practice, not all constraints are active during optimal operation of the plant and some constraints will remain inactive. 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 control them using some kind of (feedback) controller. After all active constraints have been implemented, problem (1) can be re-written as an unconstrained optimization problem,

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

where, by abuse of notation, $\mathbf{u}$ now denotes the remaining unconstrained degrees of freedom.

The term self-optimizing control refers to the procedure of selecting the control layer (see Fig. 1). The focus is set on selecting the best controlled variables $\mathbf{c} = \mathbf{H}\mathbf{y}$ such that the operating cost $J(\mathbf{u}, \mathbf{d})$ is minimized. Here $\mathbf{y}$ denotes all the candidate measurements and $\mathbf{H}$ is a selection or combination matrix. The criterion for evaluating different candidates for controlled

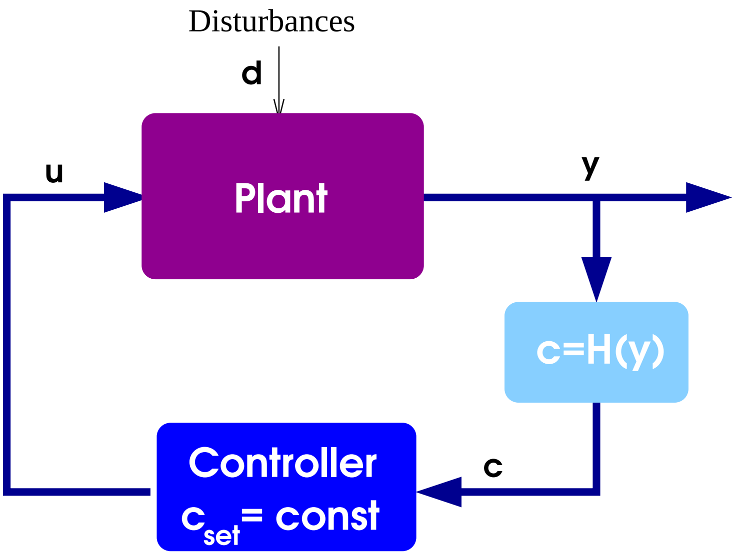


Figure 1: Block diagram SOC

variables is the loss from optimality

$$L = J(\mathbf{u}, \mathbf{d}) - J(\mathbf{u}^{opt}, \mathbf{d}), \tag{3}$$

which is imposed by the disturbance $\mathbf{d}$ and selected control structure which determines how $\mathbf{u}$ is adjusted. Using the loss $L$, Skogestad [3] defined:

> 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 candidate for this kind of controlled variable would be the gradient $\mathbf{c} = J_{\mathbf{u}}(\mathbf{u}, \mathbf{d}) = \frac{\partial J}{\partial \mathbf{u}}$, which should be zero for optimal operation under all disturbances.

This was already formulated in [8], 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.

This idea has been also mentioned in [9], where the authors write of the gradient as a controlled variable. It satisfies the conditions of not being at a constraint, while the optimal values does not vary with changing disturbances. Controlling invariants, in particular the gradient of a process has been proposed by other authors, too, see e.g. [10, 11].

However, in most cases, the gradient cannot be measured, for example, because it is a function of the unknown disturbances $\mathbf{d}$. The definition of self-optimizing control [3] includes the special case of gradient control, while leaving room for "suboptimal cases" 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 the best possible (in terms of the loss $L$) approximation to the unmeasured gradient $J_{\mathbf{u}}$ using the available measurements.

Several methods for finding self-optimizing variables have been reported in the literature [12, 7, 13, 14]. All these methods are based on a approximating the optimization problem (2) by a quadratic optimization problem

$$\min_{\mathbf{u}} [\mathbf{u}^{\mathrm{T}} \quad \mathbf{d}^{\mathrm{T}}] \begin{bmatrix} \mathbf{J_{uu}} & \mathbf{J_{ud}} \\ \mathbf{J_{du}} & \mathbf{J_{dd}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix}, \tag{4}$$

and by using a linear measurement model

$$\mathbf{y} = \mathbf{G^y u} + \mathbf{G^y_d d}. \tag{5}$$

The goal is to find a matrix $\mathbf{H}$ such that the controlled variable $\mathbf{c}$ is

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

where $\mathbf{y}_m = \mathbf{y} + \mathbf{n^y}$ and $\mathbf{n^y}$ denotes the measurement noise/bias. It is assumed that the inputs $\mathbf{u}$ are adjusted by a feedback controller to keep $\mathbf{c}$ at its setpoint $\mathbf{c}_s$. If the controller has integral action, then $\mathbf{c} = \mathbf{c}_s$ at steady 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.

*2.1. Self-optimizing control using the null-space method*

In the following we describe the null-space method [7] for determining a controlled variable $\mathbf{c} = \mathbf{H}\mathbf{y}$. We present a reformulation of the null-space theorem [14].

**Theorem 1.** *Given a sufficient number of measurements ($n_y \geq n_u + n_d$) and no measurement noise $\mathbf{n^y} = 0$, select $\mathbf{H}$ in the null space of the optimal sensitivity matrix $\mathbf{F}$,*

$$\mathbf{HF} = 0, \tag{7}$$

*where*

$$\mathbf{F} = \frac{\partial \mathbf{y}^{\text{opt}}}{\partial \mathbf{d}}. \tag{8}$$

*Controlling $\mathbf{c} = \mathbf{Hy}$ to zero yields locally zero loss from optimal operation.*

The optimal sensitivity $\mathbf{F}$ can be obtained numerically or calculated using

$$\mathbf{F} = -\mathbf{G}^y \mathbf{J}_{\mathbf{uu}}^{-1} \mathbf{J}_{\mathbf{ud}} + \mathbf{G}_d^y, \tag{9}$$

where $\mathbf{J}_{\mathbf{ud}} = \partial^2 J/(\partial \mathbf{u} \partial \mathbf{d})$ and $\mathbf{J}_{\mathbf{uu}} = \partial^2 J/\partial \mathbf{u}^2$, and we use the linearized process model (5).

We sketch a proof: In the neighborhood of the nominal point $\mathbf{d}^{\text{nom}}$ the optimal change in the measurements can be expressed using (8) as

$$\mathbf{y}^{\text{opt}}(\mathbf{d}) - \mathbf{y}^{\text{opt}}(\mathbf{d}^{\text{nom}}) = \mathbf{F}(\mathbf{d} - \mathbf{d}^{\text{nom}}). \tag{10}$$

The optimal variation in the controlled variables $\mathbf{c}$ then becomes

$$\mathbf{c}^{\text{opt}}(\mathbf{d}) - \mathbf{c}^{\text{opt}}(\mathbf{d}^{\text{nom}}) = \mathbf{HF}(\mathbf{d} - \mathbf{d}^{\text{nom}}), \tag{11}$$

and since $\mathbf{H}$ is chosen in left null space of $\mathbf{F}$, we have $\mathbf{c}^{\text{opt}}(\mathbf{d}) = \mathbf{c}^{\text{opt}}(\mathbf{d}^{\text{nom}})$ for any disturbance $\mathbf{d}$, and thus we do not need to change the setpoint for $\mathbf{c} = \mathbf{Hy}$. □

In Appendix A we show that choosing $\mathbf{H}$ in the null space of $\mathbf{F}$ is in indeed identical to selecting $\mathbf{c} = J_u$, where $J_{\mathbf{u}} = \partial J/\partial \mathbf{u}$ is the gradient of (4).

However, when the measurements are corrupted by biased noise $\mathbf{n^y}$, the null space method will not give the best possible solution. To find the best controlled variable with biased process noise on the measurements, we refer to [14].

## 3. NCO tracking

NCO tracking is the concept of adapting the inputs in such a way that the necessary conditions of optimality (NCO) are satisfied. For steady state optimization NCO tracking [4] updates the inputs iteratively. By using iterative input updates, the requirements on the model accuracy can be relaxed, because the iterative nature of the input updates introduces a natural correcting feedback. Cheng and Zafiriou [15], for example, have have developed an algorithm that can cope with severe plant-model mismatch. The NCO tracking [4] approach goes one step further and dispenses with a model completely. It has been described in literature as the idea of iteratively updating the inputs $\mathbf{u}$ of a plant to satisfy the first order necessary conditions for optimality upon convergence. Instead of controlling "normal" measurements $\mathbf{y}$, the gradient is 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.

As in self-optimizing control, it is assumed that the set of active constraints does not change when a disturbance enters the process. The optimization problem then can be written as

$$\min J(\mathbf{u}, \mathbf{d}) \quad \text{s.t.} \quad C(\mathbf{u}, \mathbf{d}) = 0. \tag{12}$$

The number of inputs, disturbances and constraints is denoted $n_u, n_d$ and $n_c$, respectively. In contrast to the previous section, $\mathbf{u}$ now denotes the all inputs, not only the unconstrained degrees of freedom. It is assumed that the linear independence constraint qualification (LICQ) hold, i.e. that the gradients of the $n_c$ active constraints are linearly independent at the optimal point.

Both, the values of the active constraints $C(\mathbf{u}, \mathbf{d})$ and the objective function are assumed to be measurable. In addition, the gradients of these functions are assumed to be "measurable", e.g. by estimating them experimentally (at each iteration) using $n_u$ perturbations and applying finite differences to estimate $J_{\mathbf{u}}(\mathbf{u}, \mathbf{d}) = \partial J/\partial \mathbf{u}$ and $C_{\mathbf{u}}(\mathbf{u}, \mathbf{d}) = \partial C/\partial \mathbf{u}$.

While in self-optimizing control, the active constraints are assumed to be directly controlled by feedback, and the task is to find the best controlled variables for the remaining degrees of freedom, NCO tracking updates all inputs iteratively to satisfy the constraints and to push the sensitivities of the unconstrained degrees of freedom to zero. This allows one to handle constraints that cannot be controlled by feedback, for example, end point constraints in dynamic optimization.

To determine appropriate update laws, which converge to the active constraints and force the unconstrained sensitivities to zero, the input directions are partitioned into two orthogonal components. The two-dimensional case is illustrated in Fig. 2, where the first component of the input $\bar{\mathbf{u}}$ is orthog-

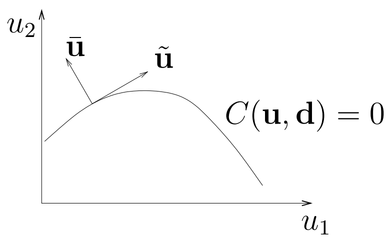


Figure 2: Input partitioning: constraint seeking input $\bar{\mathbf{u}}$, sensitivity seeking input $\tilde{\mathbf{u}}$

onal to the constraints, and is called "constraint seeking" input direction, and the second input direction $\tilde{\mathbf{u}}$ is parallel to the constraints and is called "sensitivity seeking" input direction.

The directions can be found by measuring the constraint sensitivity matrix $C_{\mathbf{u}}$, and by performing a singular value decomposition, $C_{\mathbf{u}} = USV^T$. We then write $S_G = [\bar{S}_G \quad 0]$ and $V_G = [\bar{V}_G \quad \tilde{V}_G]$ to obtain:

$$\bar{\mathbf{u}} = \bar{V}_G^T \mathbf{u} \quad \text{and} \quad \tilde{\mathbf{u}} = \tilde{V}_G^T \mathbf{u}. \tag{13}$$

Partitioning the input in this way has the advantage that changing the sensitivity seeking inputs $\tilde{\mathbf{u}}$ does not have any influence the constraints $C(\mathbf{u}, \mathbf{d})$, that is:

$$C_{\tilde{\mathbf{u}}} = \frac{\partial C(\bar{\mathbf{u}}, \tilde{\mathbf{u}}, \mathbf{d})}{\partial \tilde{\mathbf{u}}} = 0 \tag{14}$$

Assuming that the linear independence constraint qualifications hold, the NCO can be expressed using the partitioned inputs $\bar{\mathbf{u}}$ and $\tilde{\mathbf{u}}$ as:

$$\begin{aligned} C(\bar{\mathbf{u}}, \tilde{\mathbf{u}}, \mathbf{d}) &= 0 \quad \in \mathbb{R}^{n_c} \\ J_{\tilde{\mathbf{u}}} = \left( \frac{\partial J(\bar{\mathbf{u}}, \tilde{\mathbf{u}}, \mathbf{d})}{\partial \tilde{\mathbf{u}}} \right)^T &= 0 \quad \in \mathbb{R}^{n_u - n_c} \end{aligned} \tag{15}$$

To determine the input update equations, a Taylor expansion of the NCO (15) is performed around the current input point $(\bar{\mathbf{u}}, \tilde{\mathbf{u}})$.

$$\begin{aligned} C(\bar{\mathbf{u}} + \Delta\bar{\mathbf{u}}) &= C(\bar{\mathbf{u}}) + C_{\bar{u}}(\bar{\mathbf{u}})\Delta\bar{\mathbf{u}} \\ J_{\tilde{\mathbf{u}}}(\tilde{\mathbf{u}} + \Delta\tilde{\mathbf{u}}) &= J_{\tilde{\mathbf{u}}}(\tilde{\mathbf{u}}) + J_{\tilde{\mathbf{u}}\tilde{\mathbf{u}}}(\tilde{\mathbf{u}})\Delta\tilde{\mathbf{u}} \end{aligned} \tag{16}$$

At optimal operation the left hand sides of (16) are zero. So setting them equal to zero, and solving for $\Delta\bar{\mathbf{u}}$ and $\Delta\tilde{\mathbf{u}}$ gives the optimal updates of the inputs:

$$\Delta\bar{\mathbf{u}} = -\left[C_{\bar{\mathbf{u}}}(\bar{\mathbf{u}})\right]^{-1} C(\bar{\mathbf{u}}) \tag{17}$$

$$\Delta\tilde{\mathbf{u}} = -\left[J_{\tilde{\mathbf{u}}\tilde{\mathbf{u}}}(\tilde{\mathbf{u}})\right]^{-1} J_{\tilde{\mathbf{u}}}(\tilde{\mathbf{u}}) \tag{18}$$

These input updates are exact for a quadratic approximation of the system (12), in the sense that the NCO (15) are satisfied after one iteration, as in dead beat control. Considering the process at time instant $k$, the input at the next sample time $k+1$ is determine by

$$\bar{\mathbf{u}}_{k+1} = \bar{\mathbf{u}}_k + \bar{\beta}\Delta\bar{\mathbf{u}} \quad \text{and} \quad \tilde{\mathbf{u}}_{k+1} = \tilde{\mathbf{u}}_k + \tilde{\beta}\Delta\tilde{\mathbf{u}}. \tag{19}$$

Here $\bar{\beta}$ and $\tilde{\beta}$ are tuning parameters which adjust the step size, since applying the full updates may lead to infeasibility and convergence problems.

This procedure is analog to a Newton(like) optimization method. In this analogy, steady state operating periods correspond to function evaluations in the newton procedure, and the system is solved 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 [16], past inputs are used in Broyden's formula to obtain the gradients. Other methods which do not rely on input perturbations are described in [17, 18]. 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, but the inputs will still be updated iteratively only at given sample times.

Since the Hessian $J_{\tilde{\mathbf{u}}\tilde{\mathbf{u}}}$ in (18) is expensive to obtain numerically, one often uses the Hessian at the nominal operating point, which is determined experimentally only once. Alternatively, an approximation of the inverse of the Hessian can be obtained by a BFGS update.

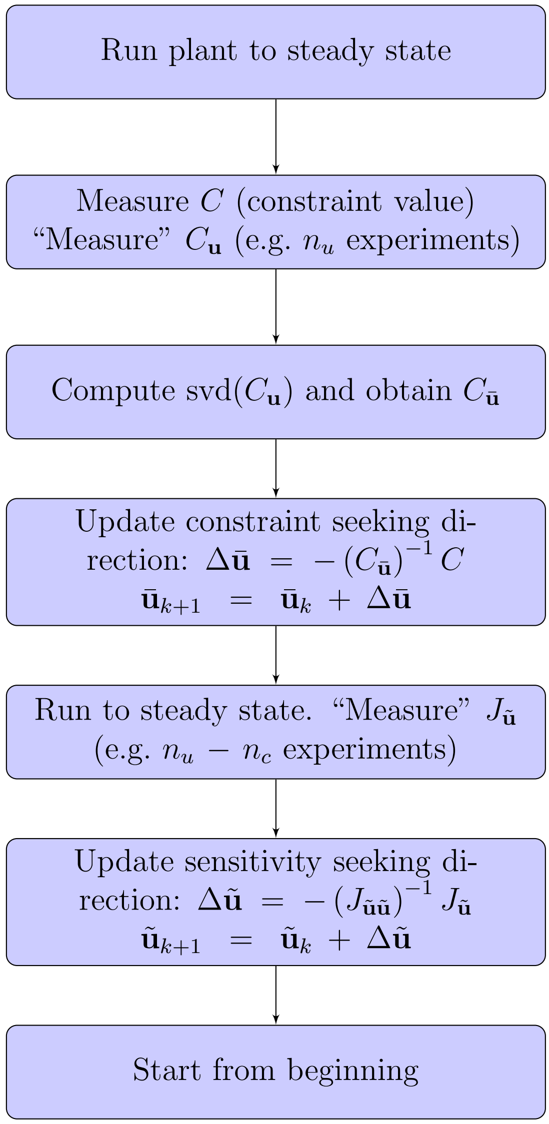


Figure 3: NCO tracking procedure

The complete NCO tracking procedure is visualized in Fig. 3. If more information on the directionality of the disturbance is available, this information can be used in the updating scheme to make the updating more efficient. For more details on the NCO tracking procedure for steady state optimization of processes, the reader is referred to [4].

*3.1. NCO tracking, unconstrained case*

This section briefly describes a simplified NCO tracking procedure which will be used later in this work. If there are no constraints, all inputs $\mathbf{u}$ are sensitivity seeking inputs, and the constraint seeking updates can be omitted. The block diagram is shown in Fig. 4. In practice, the gradient cannot just be

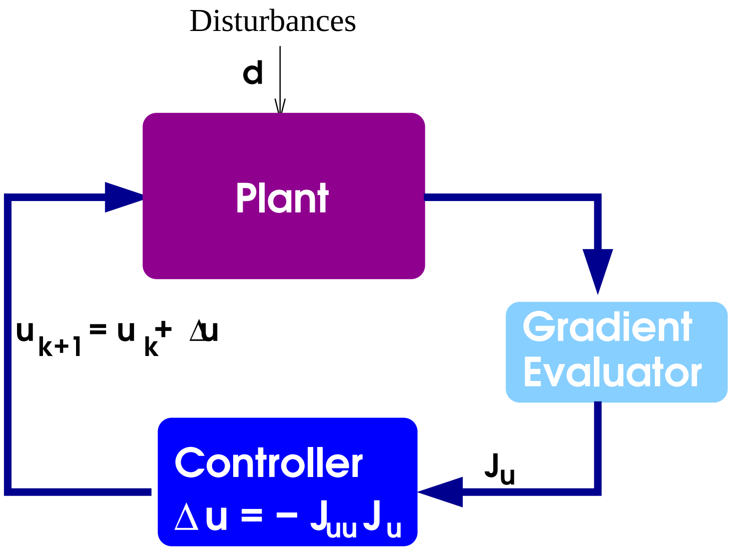


Figure 4: Block diagram NCO tracking

evaluated as indicated in Fig. 4. In this work we make a small perturbation in the input and run the process for a given time to estimate the gradient by finite differences. The simplified action sequence is shown in Fig. 5.

**Remark 1** (Recent advances in NCO tracking)**.** *Recently, the idea of NCO tracking has been extended to the case where the gradient estimate is based on output feedback. Gros and coworkers, [19] 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, [19] goes one step further and determines the actual change in the inputs. This step is omitted in the self-optimizing control structure design context, because the focus is set on finding good controlled variables, and the generation of the corresponding inputs,* $\mathbf{u}$*, is left to the feedback controller.*

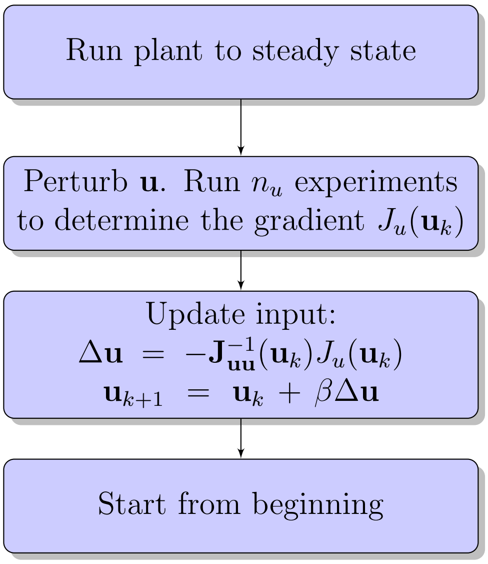


Figure 5: Simple NCO tracking procedure

*The authors of [19] consider zero mean noise, and show that if the model is invertible and the number of unknowns ($n_d$) is lesser or equal than the number of measurements ($n_{\mathbf{y}}$), the inputs can be updated to converge to the optimum.*

*In the case of biased noise, neither the null-space method nor the NCO tracking modifications introduced by [19] 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" [14].*

## 4. Self-optimizing control and NCO tracking in relation to each other

In order to suggest how to combine self-optimizing control and NCO tracking, we first consider how a chemical plant is usually operated today.

### *4.1. 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, [20, 21].

Each control layer implements the setpoints which are given from the layer above, Fig. 6.

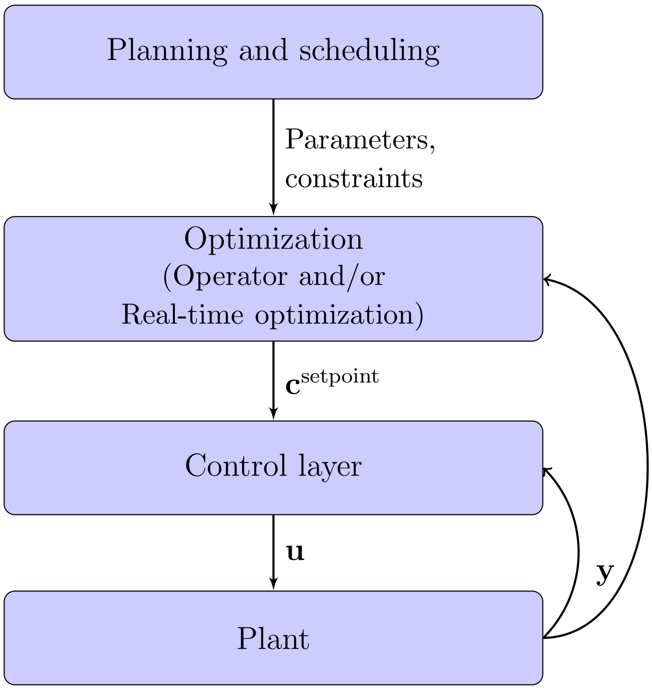


Figure 6: Vertical decomposition into control layers

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 most plants this is done by operators, but in recent years, online optimization (RTO) has been increasingly used to find good 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 PID controllers or model predictive controllers (MPC), which act directly 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 to their original setpoints. After the plant has settled down and (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 has to settle down again. The long time delay between start of the disturbance and reaching the final optimized operation point is one of the challenges for the optimization layer. In particular if RTO is used, it is not possible to counteract disturbances which occur on a fast time scale [22]. This limits successful RTO applications for 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 with an economic objective function would allow setpoint changes without having to wait for steady state. However, practical obstacles have prevented the dynamic RTO (DRTO) from becoming a standard tool in 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. And even if a good model and states are available in the DRTO, the dynamic optimization problem itself is difficult to solve.

### *4.2. Properties of self-optimizing control and NCO tracking*

Both methods, NCO tracking and self-optimizing control, pursue the same goal, minimization of the operating cost. The main difference, as we see it, is that in NCO tracking, we focus on manipulating the input values directly at given sample times to force the sensitivities to zero. As the name implies, the necessary conditions of optimality are the controlled variables, and the method is basically a kind of control law, which updates the inputs accordingly.

In self-optimizing control, we focus on finding controlled variables which do not need frequent updates. 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 to find controlled variables, which give acceptable operation. The active constraints and the self-optimizing variables are kept at their setpoints by feedback controllers, so there is no need for solving for the optimal inputs explicitly.

In summary, we may say that the NCO tracking procedure works as a controller which calculates directly the required input change $\Delta\mathbf{u}$. In self-optimizing control, we are not interested in the inputs, as they are taken care of by the controllers. We are rather interested in finding the right controlled variable $\mathbf{c} = \mathbf{H}\mathbf{y}$, which when kept constant, leads to the correct input action $\mathbf{u}$.

In table 1 we have listed the main differences between the null space method and NCO tracking.

Table 1: Summary of properties

| Null space method | NCO tracking |
|---|---|
| Procedure for finding $\mathbf{c} = \mathbf{H}\mathbf{y}$ | Controlled variable: $\mathbf{c} = J_{\mathbf{u}}$ |
| $J_{\mathbf{u}}$ and $C_{\mathbf{u}}$ not measured | $J_u$ and $C_{\mathbf{u}}$ measured |
| Set of important $\mathbf{d}$ assumed a priori | No assumptions on disturbances |
| $\mathbf{F} = \frac{\partial \mathbf{y}^{opt}}{\partial \mathbf{d}}$ obtained from model | No model needed |
| Active constr. satisfied by feedback | Active constr. satisfied iteratively |
| Optimal for expected disturbances | Optimal for unexpected disturbance |
| Local, linearized at nom. point | Local, linearization point moves |
| Direct, continuous input change (PID) | Discrete updates at sample times |
| Fast (feedback) | Slow (acts only at sample times) |

*4.3. Using self-optimizing control and NCO tracking together*

The previous observations lead us to consider self-optimizing control and NCO tracking (or RTO) as *complementary*, and use them together. NCO tracking fits better into the optimization layer and is thus an alternative to model based real-time optimization (RTO), while self-optimizing control

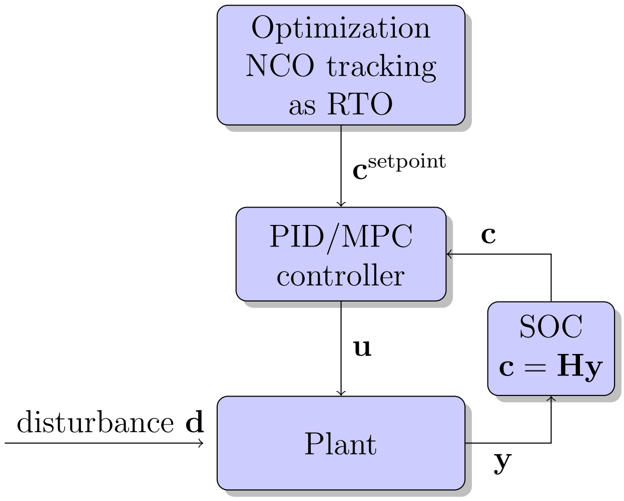


Figure 7: Relation between NCO tracking and self-optimizing control

should be used in the lower layer and follow the setpoints coming from the NCO tracking layer, as shown in Fig. 7.

It may be argued that if NCO tracking or a 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, this 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.

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 the measurement selection process is based on expected disturbances, whereas NCO tracking can also handle unknown disturbances. Another reason is that if a disturbance moves the process far from the linearization point, the local model approximation may be poor. So the self-optimizing control structure cannot reject unexpected disturbances or 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 underneath the optimization layer instead of directly computing the plant input $\mathbf{u}$. This rejects expected disturbances on a fast timescale, while the unexpected disturbances are rejected by the NCO tracking/RTO layer updates. Applying self-optimizing control is thus basically an intelligent way to implement the control layer below the RTO layer.

## 5. Case study

### 5.1. 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 Fig. 8. The process model is taken

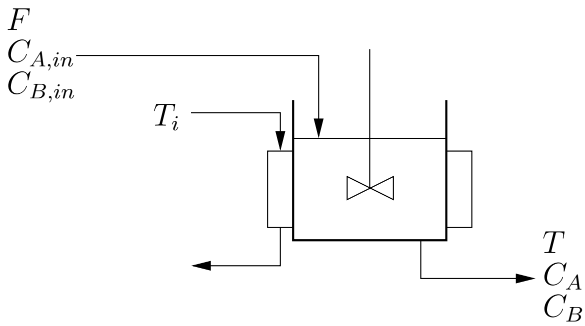


Figure 8: Schematic diagram of a CSTR

from [23], and the dynamics of the system are described by following set of equations,

$$\frac{dC_A}{dt} = \frac{1}{\tau}(C_{A,in} - C_A) - r \tag{20}$$

$$\frac{dC_B}{dt} = \frac{1}{\tau}(C_{B,in} - C_B) + r \tag{21}$$

$$\frac{dT}{dt} = \frac{1}{\tau}(T_i - T) + \frac{-\Delta H_{rx}}{\rho c_p} r \tag{22}$$

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

$$r = k_1 C_A - k_2 C_B \tag{23}$$

where

$$k_1 = A_1 e^{\frac{-E_1}{RT}} \quad \text{and} \quad k_2 = A_2 e^{\frac{-E_2}{RT}}, \tag{24}$$

and $C_1$ and $C_2$ are the Arrhenius factors for the reaction constants $k_1$ and $k_2$.

This process has one manipulated input ($u$), the jacket temperature $T_i$. The expected disturbances $d_1$ and $d_2$ enter the process as variations in the feed concentrations $C_{A,in}, C_{B,in}$, and the measured variables are $y_1 = C_A$, $y_2 = C_B$, $y_3 = T$. The objective is to maximize the profit function which is a trade-off between cooling cost and income from selling product $B$ ([24]):

$$P = [p_{C_B} C_B - (p_{T_i} T_i)^2], \tag{25}$$

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

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

Table 2: Objective function parameters

*5.2. Simulations*

First, we control the process for the expected disturbances using direct NCO tracking. Next, we use a self-optimizing CV, $\mathbf{c} = \mathbf{Hy}$, obtained using the null space method and compare the results with direct NCO tracking. After comparing both control structures for an unexpected disturbance, we finally combine the methods as shown in Fig.7.

The expected disturbance scenario is given in Fig. 9. After 500 minutes at the nominal value, the concentration $C_{A,in}$ ($d_1$) varies sinusoidal 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

| Variable | Value | Unit | Description |
|---|---|---|---|
| $T_i$ | 424.20 | K | Reactor 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 $\text{min}^{-1}$ | 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 $kg^{-1}\text{K}^{-1}$ | Heat capacity |
| $E_1$ | 10000 | cal $\text{mol}^{-1}$ | Activation energy 1 |
| $E_2$ | 15000 | cal $\text{mol}^{-1}$ | Activation energy 2 ($d_3$) |
| $R$ | 1.987 | cal $\text{mol}^{-1}\text{K}^{-1}$ | Ideal gas constant |
| $-\Delta H_{rx}$ | 5000 | cal $\text{mol}^{-1}$ | Heat of reaction |
| $\rho$ | 1.000 | kg/l | Density |
| $\tau$ | 1.000 | min | Residence time |

Table 3: Nominal values for the CSTR model

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.

### *5.2.1. 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 forth NCO iteration. Changing the sign of the perturbation was found to give better overall performance of the NCO procedure. No steady state detection is 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 $\beta$ is set to 0.4.

Fig. 10 shows the concentration and temperature trajectories for the NCO tracking procedure. The control strategy enables acceptable control. It is fur-

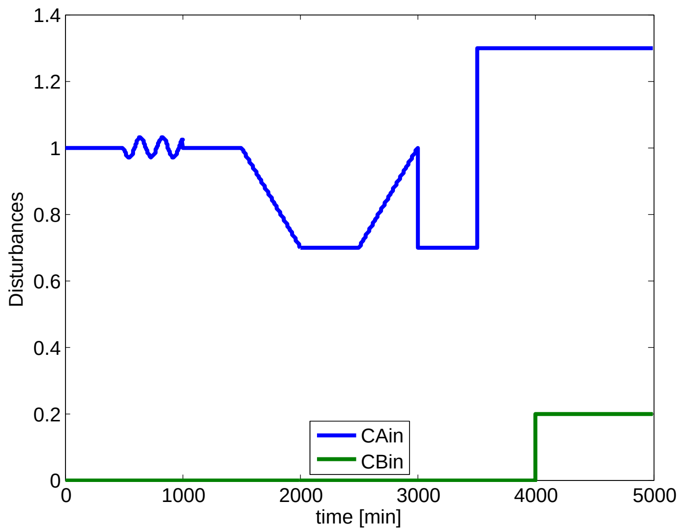


Figure 9: Disturbance trajectories $C_{A,in}, C_{b,in}$

thermore found that the step disturbances are very well handled. Since the

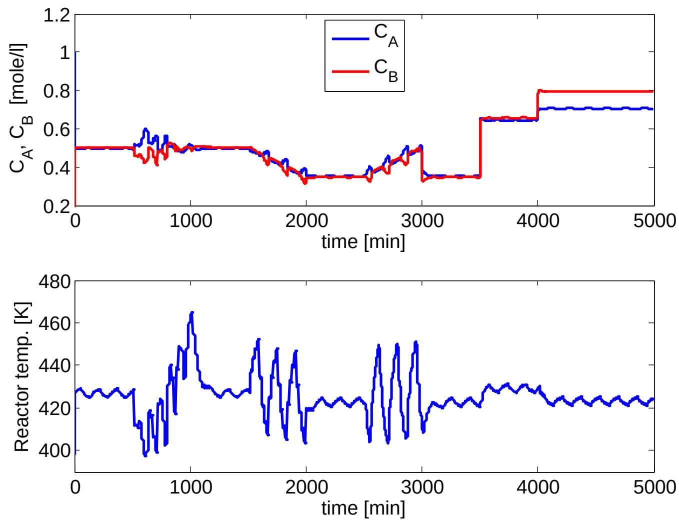


Figure 10: NCO tracking, concentrations and temperature

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 lead to 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 $\beta$, the sample time, and the timing and kind of the disturbance, and of course the perturbation for estimating the gradient.

*5.2.2. Self-optimizing control using the null-space method*

Next, the process is controlled using the null space method from Section 2. Since we have one input and 2 disturbances to compensate for, we need three measurements for the invariant variable combination, so $\mathbf{y} = [\ C_A \quad C_B \quad T\ ]^{\mathrm{T}}$. We 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

$$\mathbf{F} = \frac{\partial \mathbf{y}^{\mathrm{opt}}}{\partial \mathbf{d}} = \begin{bmatrix} -0.4862 & -0.3223 \\ -0.5138 & -0.6777 \\ -9.9043 & 40.5807 \end{bmatrix}. \tag{26}$$

Then with $\mathbf{H} = [\ -0.7688 \quad 0.6394 \quad 0.0046\ ]$ we have that $\mathbf{HF} = 0$. Using a PI controller, the self-optimizing variable $\mathbf{c} = \mathbf{Hy} = -0.7688C_A + 0.6394C_B + 0.0046T$ 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 Fig. 11. Compared with the concentrations and temperature using NCO tracking, the trajectories are much smoother.

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

As may be guessed from the trajectories for NCO tracking and self-optimizing control, the input usage for the two cases is quite different, Fig. 12. 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, the NCO tracking changes the input excessively.

Comparing the profits, Fig. 13, shows that both systems are very similar in the steady state periods, but for disturbances, where no steady state is

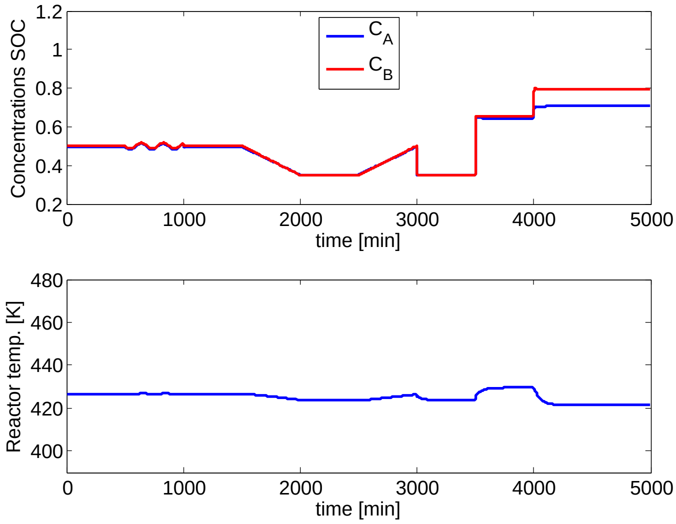


Figure 11: SOC, concentrations and temperature

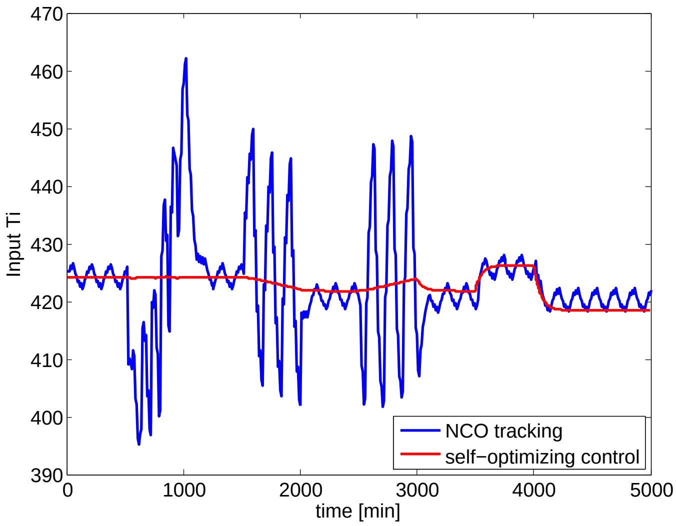


Figure 12: Input usage for SOC and NCO tracking

reached within one sample time, NCO tracking is not performing as well as the self-optimizing control policy using the null-space method.

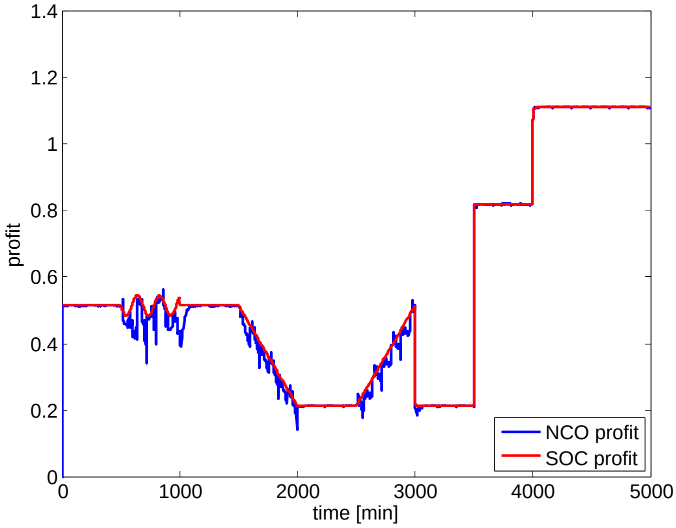


Figure 13: Profit for SOC and NCO tracking

#### *5.2.4. Using NCO tracking as RTO and self-optimizing control in the lower layer*

If it can be guaranteed that the disturbances in the feed concentration are the only ones entering the process, then using only self-optimizing control is sufficient, and a RTO layer is not necessary. However, the situation changes for disturbances not anticipated in the control structure design. Consider a positive step change in the activation energy $E_2$ ($d_3$) of 3% at time 3100 min. This disturbance reduces the reaction rate for the reverse reaction, especially at higher temperatures. Comparing the profits using the two control structures, Fig. 14, shows that the self-optimizing control system cannot exploit the improved conditions caused by the unexpected disturbance.

Adapting the self-optimizing control setpoints using RTO or NCO tracking can solve this problem, and at the same time reduce RTO or NCO tracking sample time. In Fig. 15 the instantaneous profit for direct NCO tracking (sample time: 10 min) and the combined system with a sample time of 25 min is shown. The combined system operates smoother than the pure NCO tracking system while giving similar performance in terms of the profit. However, considering the input usage, Fig. 16, we find that the combination of self-optimizing control and NCO tracking gives a substantially smoother input action than direct NCO tracking. Using online RTO, the performance

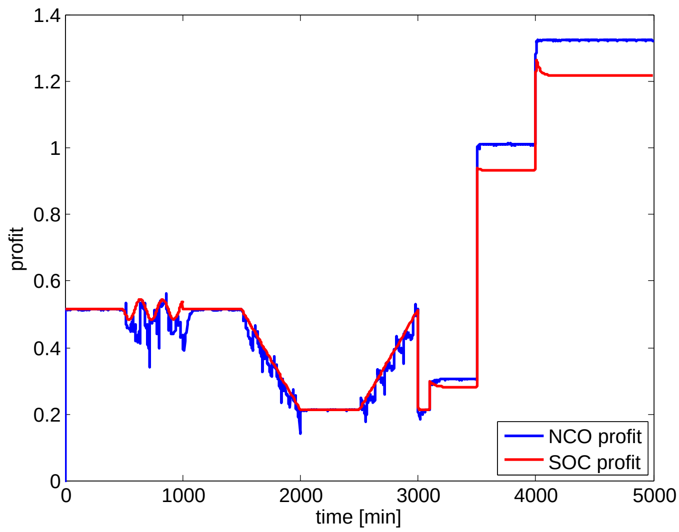


Figure 14: Profit, for NCO tracking and for SOC with unexpected disturbance ($d_3$) at 3100 min

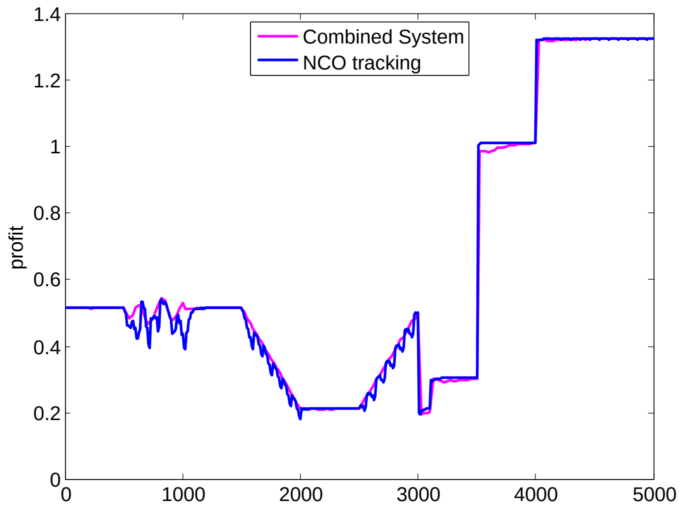


Figure 15: Profit for combined SOC/NCO tracking (25 min sample time) and direct NCO tracking (10 min sample time)

could be improved even further because the setpoints would move directly to the optimal values instead of iteratively approaching them. However, un-

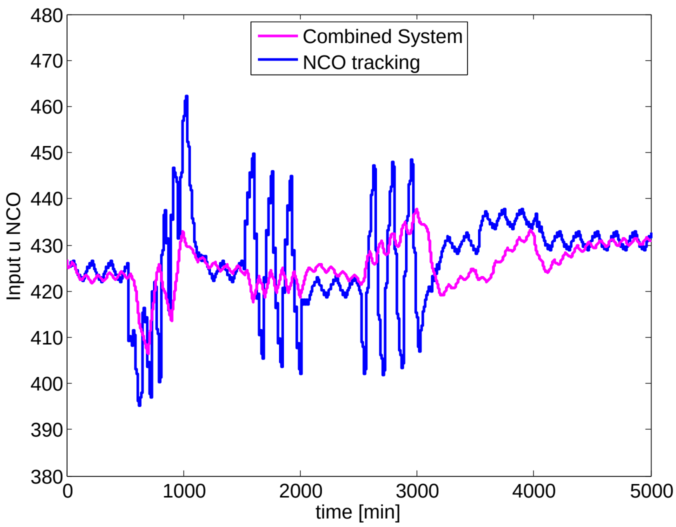


Figure 16: Input, combined NCO/SOC and direct NCO tracking

modelled (unexpected) disturbances are not rejected in online RTO either

## 6. Discussion

There has been some confusion about the relationship between "self-optimizing" control approach of Skogestad and coworkers and the NCO tracking approach of Bonvin and coworkers. The reason for the confusion is that both approaches seek to optimize operation and make the gradient zero ($J_u = 0$), but there are significant differences:

In self-optimizing control the idea is to use offline calculations to obtain good controlled variables, typically as linear combination of the measurements, $\mathbf{c} = \mathbf{Hy}$, where $\mathbf{c}$ may be considered an estimate of the gradient. It is critical to have a model of the expected disturbances when obtaining $\mathbf{c}$. One does not compute the optimal inputs explicitly; they are generated by a feedback controller to make $\mathbf{c} = \mathbf{Hy} = \mathbf{c}_s$ (constant).

In NCO tracking one aims at obtaining the optimal inputs $\mathbf{u}$ that drive the measured or estimated gradient to zero. It is not necessary to know the disturbances in advance, and no model is needed.

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 let NCO tracking adjust the setpoints, $\mathbf{c} = \mathbf{c}_{opt}$ based on online estimates of the gradient.

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 $\beta$, 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. Especially the non-steady state periods make it difficult to find parameters which optimize the cost with acceptable input usage.

The NCO tracking procedures 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. Other more advanced gradient and update methods may give better overall performance, especially in terms of input usage, because poor update steps caused by wrong gradient estimates would be avoided.

However, we chose to apply the simplest method in this work, because our purpose is to demonstrate that the basic concepts of NCO tracking and self-optimizing control 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.

## 7. Conclusion

The different characteristics of the two methods studied in this paper suggest to consider 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 dynamic control system in the layer below, using a self-optimizing control structure in the lower layer, improves performance and can significantly reduce need for RTO updates. For NCO tracking as implemented in this paper, this means less 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 and the null space method

Consider the unconstrained optimization problem

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

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

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

The linear model (5) can be rewritten as:

$$\mathbf{y} = \tilde{\mathbf{G}}^{\mathbf{y}} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix} \tag{A.3}$$

If we assume that we have a sufficient number of measurements, $n_y = n_u + n_d$, then the model may be inverted, and substitution into (A.2) gives

$$J_u = \begin{bmatrix} \mathbf{J_{uu}} & \mathbf{J_{ud}} \end{bmatrix} [\tilde{\mathbf{G}}^{\mathbf{y}}]^{-1}\mathbf{y}. \tag{A.4}$$

At the optimum, we have $J_u = 0$, or equivalently $\mathbf{c} = \mathbf{Hy} = 0$, where $\mathbf{H} = [\mathbf{J_{uu}}\ \mathbf{J_{ud}}][\tilde{\mathbf{G}}^{\mathbf{y}}]^{-1}$. This is the same expression for $\mathbf{H}$ as derived in [14]. And indeed, if we evaluate $\mathbf{HF}$ using $\mathbf{F}$ in (9), we get $\mathbf{HF} = 0$. This follows since $\mathbf{F}$ in (9) may be rewritten as

$$\mathbf{F} = \tilde{\mathbf{G}} \begin{bmatrix} -\mathbf{J_{uu}^{-1}J_{ud}} \\ \mathbf{I} \end{bmatrix}. \tag{A.5}$$

Also note that the loss $L$ and gradient are related by

$$L = \frac{1}{2} J_u \mathbf{J_{uu}^{-1}} J_u, \tag{A.6}$$

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

## References


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

[2] 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. analsysis of the optimizing control structures, AIChE Journal 26 (1980) 220–232.

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

[4] 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.

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

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

[7] V. Alstad, S. Skogestad, Null space method for selecting optimal measurement combinations as controlled variables, Ind. Eng. Chem. Res. 46 (2007) 846–853.

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

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

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

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

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

[13] V. Kariwala, Y. Cao, S. Janardhanan, Local self-optimizing control with average loss minimization, Ind. Eng. Chem. Res. 47 (2008) 1150–1158.

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

[15] J.-H. Cheng, E. Zafiriou, Robust model-based iterative feedback optimization of steady state plant operations, Industrial & Engineering Chemistry Research 39 (2000) 4215–4227.

[16] 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.

[17] 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.

[18] W. Gao, S. Engell, Iterative set-point optimization of batch chromatography, Computers & Chemical Engineering 29 (2005) 1401 – 1409. Selected Papers Presented at the 14th European Symposium on Computer Aided Process Engineering.

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

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

[21] S. Skogestad, Control structure design for complete chemical plants, Computers and Chemical Engineering 28 (2004).

[22] S. Engell, Feedback control for optimal process operation, Journal of Process Control 17 (2007) 203 – 219. Special Issue ADCHEM 2006 Symposium.

[23] C. G. Economou, M. Morari, Internal model control. 5. extension to nonlinear systems, Ind. Eng. Chem. Process Des. Dev. 25 (1986) 403–411.

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