

---

# Comparison of six implicit real-time optimization schemes

Grégory François<sup>1</sup>, Bala Srinivasan<sup>2</sup>, Dominique Bonvin<sup>3</sup>

1. Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne,  
CH-1015 Lausanne, Switzerland

gregory.francois@epfl.ch

2. École Polytechnique de Montréal, Montreal, H3C 3A7 Canada

bala.srinivasan@polymtl.ca

3. Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne,  
CH-1015 Lausanne, Switzerland

dominique.bonvin@epfl.ch

---

*ABSTRACT.* Real-time optimization (RTO) is a class of methods that use measurements to reject the effect of uncertainty on optimal performance. This article compares six implicit RTO schemes, that is, schemes that implement optimality not through numerical optimization but rather via the control of appropriate variables. For unconstrained processes, the ideal controlled variable is the cost gradient. It is shown that, because of their structural differences, model-free and model-based techniques exhibit different features in terms of required excitation, convergence, scalability with the number of inputs and rejection of uncertainty. This comparison is illustrated through a simulated CSTR.

*RÉSUMÉ.* L'optimisation en temps réel (RTO) est une classe de méthodes où les mesures sont utilisées pour rejeter l'effet de l'incertitude. Cet article compare six techniques de RTO implicites qui optimisent un procédé en contrôlant certaines variables. En l'absence de contraintes, la grandeur commandée idéale est le gradient de la fonction coût. A cause de leurs différences structurelles, les méthodes sans modèle et les méthodes basées sur le modèle se comportent différemment en termes de besoin d'excitation, de temps de convergence, de capacité de mise à l'échelle et d'aptitude à rejeter l'effet d'incertitudes. Cette comparaison est illustrée en simulation au moyen d'un réacteur chimique à marche continue.

*KEYWORDS:* Real-time optimization, gradient estimation, finite differences, extremum-seeking control, neighboring-extremal control, self-optimizing control.

*MOTS-CLÉS:* Optimisation en temps réel, estimation des gradients, différences finies, extremum-seeking control, neighboring-extremal control, self-optimizing control.

---

DOI:10.3166/JESA.x.1-?? © 2013 Lavoisier

## 1. Introduction

The use of process optimization has become so widespread (Rotava, Zanin, 2005) that optimization is seen today as a viable *technology* (Boyd, Vandenberghe, 2004). One of the main difficulties associated with process optimization regards the use of a possibly inaccurate process model. A process model is typically a set of differential and algebraic equations, which invariably represents an oversimplification of the reality. Model prediction will differ from plant behavior because of uncertainty in the form of plant-model mismatch, parametric uncertainty and process disturbances. Model uncertainty is detrimental, as model-based optimization leads to the computation of inputs that are optimal for the model and not for the plant. This can be seen as the inability of the model to predict the necessary conditions of optimality (NCO) of the plant. Because the plant NCO include conditions on both plant constraints and plant sensitivities, the inability of model-based optimal inputs to predict the plant NCO will lead to constraint violations and/or suboptimal performance.

An efficient way to combat the detrimental effect of model uncertainty on the plant performance is to incorporate process measurements in the optimization framework. There are two wide classes of methods that use measurements. With *explicit optimization* methods, the optimization problem is solved repeatedly, while measurements are used to either adapt the model parameters, thus leading to an updated model that is used for optimization (Marlin, Hrymak, 1997), or update correction terms that are added to the cost and constraint functions of the optimization problem (Marchetti *et al.*, 2009). The second class of measurement-based optimization methods is referred to as *implicit optimization* and proposes to use measurements to adapt the process inputs directly, that is, without repeating the optimization. This can be done in three different manners. In (zeroth-order) techniques labeled evolutionary optimization (Box, Draper, 1987), a simplex-type algorithm is used to approach the optimum, with the cost function being measured experimentally for every visited operating condition. In perturbation (first-order) techniques, such as extremum-seeking control (Ariyur, Krstic, 2003), the gradients are estimated experimentally using sinusoidal excitation; this scheme uses only the measurement of the cost function. The third family of schemes includes NCO tracking (Francois *et al.*, 2005) and self-optimizing control (Skogestad, 2000), which formulate the optimization as the control of certain variables, whose optimal values are (approximately) invariant to uncertainty; in contrast to perturbation methods, these schemes use output information.

This paper compares several model-free and model-based implicit optimization schemes. Optimality implies meeting the plant NCO, that is, both constraint and sensitivity requirements. To simplify the presentation, and also since it is widely accepted that experimental gradients are more difficult to evaluate than constraint values, this paper will only consider the case of unconstrained optimization. Hence, real-time optimization will amount to controlling the gradient to zero. Three model-based techniques, for which the gradient is estimated on the basis of a process model, are compared to three model-free techniques, where the gradient information is constructed from process measurements. The comparison is performed using several criteria, with

the objective to evaluate qualitatively the main features of the techniques and not to rank them in any way. Hence, the contribution of this paper is to provide guidelines that help choose an appropriate implicit optimization scheme for the problem at hand, while highlighting potential difficulties that could be faced during implementation.

The paper is organized as follows. In Section 2, the problem of unconstrained RTO using gradient control is formulated. Six implicit RTO schemes are described in Section 3 and compared in Section 4. The illustrative example of a simulated continuous stirred-tank reactor is presented in Section 5. Finally, Section 6 concludes the paper.

## 2. Unconstrained Real-time Optimization via Gradient Control

### 2.1. Problem Formulation

In its general formulation, RTO addresses the minimization or maximization of some steady-state cost function. It is useful to distinguish between static RTO, for which only steady-state measurements are used, and RTO for the steady-state optimization of a dynamical plant, where online measurements are used. In the latter case the inputs will lead the system to settle at some optimal steady-state operating point.

An unconstrained steady-state optimization problem can be formulated as follows:

$$\min_{\mathbf{u}} J_p(\mathbf{u}) := \phi_p(\mathbf{u}, \mathbf{y}_p) \quad (1)$$

where the subscript  $p$  stands for "plant",  $J_p$  is the scalar performance index to be minimized,  $\mathbf{u} \in \mathfrak{R}^m$  are the inputs,  $\mathbf{y}_p \in \mathfrak{R}^p$  are the plant outputs, and  $\phi_p : \mathfrak{R}^m \times \mathfrak{R}^p \rightarrow \mathfrak{R}$  is the scalar cost function to be minimized.

Problem [1] can be solved using a steady-state model of the plant. The model is typically inaccurate since the plant mapping  $\mathbf{y}_p(\mathbf{u})$  is unlikely to be perfectly known. Assuming that the model outputs  $\mathbf{y}$  can be expressed explicitly as functions of the inputs  $\mathbf{u}$  and the parameters  $\boldsymbol{\theta}$ , i.e.  $\mathbf{y} = \mathbf{H}(\mathbf{u}, \boldsymbol{\theta})$ , the following model-based optimization problem can be written:

$$\min_{\mathbf{u}} J(\mathbf{u}) = \Phi(\mathbf{u}, \boldsymbol{\theta}) \quad (2)$$

where  $J$  is the scalar performance index of the model and  $\boldsymbol{\theta} \in \mathfrak{R}^q$  are uncertain model parameters. A solution to Problem [2] can be found by solving  $\frac{\partial \Phi}{\partial \mathbf{u}} = \mathbf{0}$ .

However, solving these equations for the model does not necessary lead to *plant* optimality. The difficulty arises from the fact that plant-model mismatch, parametric uncertainty and process disturbances lead to an inaccurate prediction of the plant gradient  $\frac{\partial \phi_p}{\partial \mathbf{u}}$ . Measurement-based RTO techniques propose to use measurements to satisfy the *plant* NCO. With implicit optimization, this will be performed by driving the plant gradient to zero instead of repeating model-based optimization.

**Remark:** Although the focus is here on unconstrained optimization, the RTO schemes discussed in this paper can also be applied to constrained optimization problems. For example, when there are more inputs than active constraints, some input

directions can be computed to meet the active constraints, while the other directions are used to force the reduced gradients to zero (Francois *et al.*, 2005).

## 2.2. Gradient Control

For unconstrained optimization, the NCO indicate that the plant gradient  $\mathbf{g}_p := \frac{\partial J_p}{\partial \mathbf{u}}$  should be zero for optimality. Implicit RTO methods will adapt the inputs to force the estimated plant gradient to zero, with the methods differing in the way the gradient is computed. When only steady-state measurements are used, the following discrete adaptation law can be used:

$$\mathbf{u}_{c,i+1} = \mathbf{u}_{c,i} - \kappa \mathbf{P} \mathbf{g}_i \quad (3)$$

where  $\mathbf{u}_c$  are the *computed* optimal inputs (by opposition to  $\mathbf{u}$ , which represent the *applied* inputs, as will be seen later),  $\mathbf{g}$  is the estimated gradient,  $\kappa$  is the adaptation gain,  $\mathbf{P}$  is an approximation of the inverse of the Hessian, and  $i$  is the iteration number.

Theoretically, the gradient  $\mathbf{g}$  is only defined when the system is at steady state. In practice, however,  $\mathbf{g}(t)$  corresponds to an estimated quantity, whose limiting value is the gradient  $\frac{\partial J}{\partial \mathbf{u}}$  when the system reaches steady state. For the sake of simplicity,  $\mathbf{g}(t)$  will also be referred to as the "*gradient*" hereafter. When continuous-time measurements are available, the following integral adaptation law can be used:

$$\dot{\mathbf{u}}_c(t) = -\kappa \mathbf{P} \mathbf{g}(t) \quad (4)$$

## 3. Gradient Estimation Techniques

This section describes 3 model-free and 3 model-based gradient estimation techniques.

### 3.1. Model-free Techniques

The model-free techniques assume that the cost function can be directly measured, i.e. there is the single output  $y = J$ ,  $p = 1$ . The gradient is obtained by exciting the system through its inputs and calculating the gradient from the corresponding output values. The model-free techniques are presented hereafter for the monovariate case, with an indication on how the scheme extends to the multivariable case.

**Gradient from finite differences (FD):** Two different constant input values are applied, each for the duration  $T$ , which is sufficiently long to allow the system to reach steady state; the gradient is computed using the finite-difference approximation:

$$\begin{aligned} u(t) &= \begin{cases} u_{c,i} & 2iT \leq t < (2i+1)T \\ u_{c,i} + \Delta & (2i+1)T \leq t < (2i+2)T \end{cases} \\ g_i &= \frac{J((2i+2)T) - J((2i+1)T)}{\Delta} \end{aligned} \quad (5)$$

Note that FD is the only method discussed in this paper, for which only steady-state measurements are used. Hence,  $g_i$  corresponds to the gradient *stricto senso* at the  $i^{th}$  iteration. In the multivariable case, each input is perturbed individually in a similar way, and the corresponding perturbed cost is measured once steady state has been reached.

**Gradient from extremum-seeking control (ESC):** Excitation is added in the form of a dither sinusoidal signal, and the gradient is estimated by correlation. The approach includes high-pass filtering to remove the constant term in the cost and low-pass filtering to help compute the derivative of the gradient (Krstic, Wang, 2000). The various operations can be formulated in the following compact mathematical form:

$$\begin{aligned} u(t) &= u_c(t) + \Delta \sin(\omega t), & \frac{d\bar{J}(t)}{dt} &= \alpha(J(t) - \bar{J}(t)) \\ \frac{dg(t)}{dt} &= \beta \left( \frac{2(J(t) - \bar{J}(t)) \sin(\omega t)}{\Delta} - g(t) \right) \end{aligned} \quad (6)$$

where  $\bar{J}$  is a filtered cost, and  $\alpha$  and  $\beta$  represent filter coefficients. The gradient is then obtained by numerical integration of Equation 6 as shown in Figure 1.

Extremum-seeking control has three time scales; the fastest scale corresponds to the controlled plant, the medium scale is associated with the periodic excitation, while the slowest scale deals with the two filters. The reason for using a slow periodic excitation is to avoid that the system dynamics interfere with the computation of the gradient; furthermore, with slow excitation, the plant can be seen as a static map. To extend this scheme to the multivariable case, one needs to excite each input individually, i.e. with its own excitation frequency.

**Gradient from multiple units (MU):** This method assumes the availability of multiple similar units. The inputs to the various units differ by an offset, and the gradient is estimated delay-free from the difference in the measured costs (Srinivasan, 2007). With the units labeled 'a' and 'b', one has:

$$u_a(t) = u_c(t) + \frac{\Delta}{2}, \quad u_b(t) = u_c(t) - \frac{\Delta}{2}, \quad g(t) = \frac{J_a(t) - J_b(t)}{\Delta} \quad (7)$$

as illustrated in Figure 1. The typical configuration for the multivariable case is to work with  $m + 1$  units. Only one input direction is perturbed for each of the first  $m$  units, while the last unit has no input offset. The gradient is computed in the  $m$  directions by comparing the cost of each of the first  $m$  units to the last one, as in the mono-dimensional case.

### 3.2. Model-based Techniques

Model-based techniques include the following assumptions: (i) the model is *structurally* correct; (ii) there are no process disturbances. Hence, only parametric uncertainty is considered, that is, the source of uncertainty is known (so-called expected

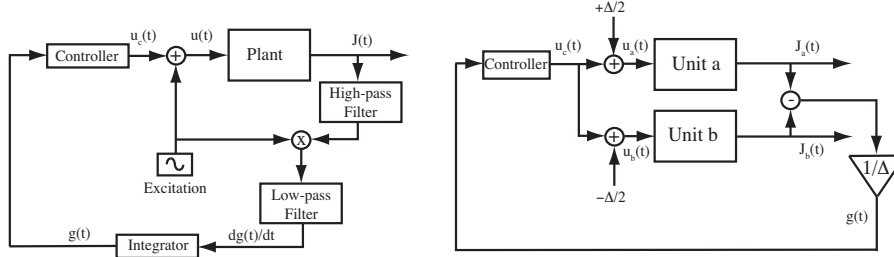


Figure 1. Extremum-seeking control (left) Multiple-unit approach (right).

uncertainty). It is furthermore assumed that there are at least as many measurements as there are uncertain parameters, i.e.  $p \geq q$ , which provides sufficient information to estimate the uncertain parameters without having to excite the system. The model is then used to compute the gradient and subsequently the optimal inputs.

**Gradient from adapted model (AM):** This corresponds to the classical two-step approach, whereby the model parameters are estimated and the updated model is used for optimization. In this work, the recursive parameter identification algorithm is written in continuous time as the associated differential equation (Ljung, 1999), which is justified by the difference in dynamics between the plant and the (slower) parameter estimation. Numerical optimization is implemented by forcing the gradient to zero. Hence, the scheme can be written as (Figure 2):

$$\begin{aligned} \dot{\hat{\theta}}(t) &= k_{\theta} \left( \frac{\partial \mathbf{H}(\mathbf{u}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)^+ \Big|_{\mathbf{u}(t), \hat{\boldsymbol{\theta}}(t)} \left( \mathbf{y}_p(t) - \mathbf{H}(\mathbf{u}(t), \hat{\boldsymbol{\theta}}(t)) \right), \quad \hat{\boldsymbol{\theta}}(0) = \boldsymbol{\theta}_0 \\ \mathbf{g}(t) &= \frac{\partial \Phi(\mathbf{u}, \boldsymbol{\theta})}{\partial \mathbf{u}} \Big|_{\mathbf{u}(t), \hat{\boldsymbol{\theta}}(t)} \end{aligned} \quad (8)$$

where  $\hat{\boldsymbol{\theta}}$  represents the parameter estimates and  $k_{\theta}$  is the gain used for parameter estimation. The deviations between predicted and measured outputs are used to adapt the values of the uncertain parameters. The gradient estimate is computed from the adapted model and used in the gradient control law (4).

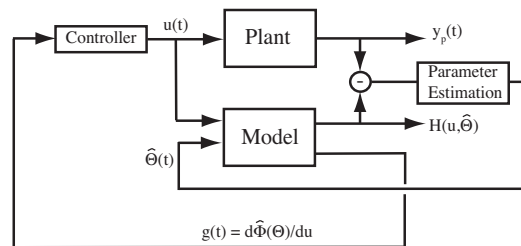


Figure 2. Gradient calculation from adapted model.

**Gradient from neighboring-extremal control (NEC):** Uncertainty causes the optimal inputs and outputs as well as the gradient to deviate from their nominal values. NEC proposes to estimate the gradient on the basis of a variational analysis around the *nominal* operating point for which the nominal gradient  $\mathbf{g}_0 = \mathbf{0}$  (Gros *et al.*, 2009).

The variation of the outputs at steady state can be written as:

$$\delta \mathbf{y} = \frac{\partial \mathbf{H}}{\partial \mathbf{u}} \delta \mathbf{u} + \frac{\partial \mathbf{H}}{\partial \boldsymbol{\theta}} \delta \boldsymbol{\theta} \quad (9)$$

with  $\delta \mathbf{y} := \mathbf{y} - \mathbf{y}_0$ ,  $\delta \mathbf{u} := \mathbf{u} - \mathbf{u}_0$  and  $\delta \boldsymbol{\theta} := \boldsymbol{\theta} - \boldsymbol{\theta}_0$ . For  $p \geq q$ , the variation of the uncertain parameters can be estimated using output and input measurements as  $\delta \boldsymbol{\theta} = \left(\frac{\partial \mathbf{H}}{\partial \boldsymbol{\theta}}\right)^+ \left[\delta \mathbf{y} - \frac{\partial \mathbf{H}}{\partial \mathbf{u}} \delta \mathbf{u}\right]$ .

Similarly, the gradient can be written as:

$$\frac{\partial \Phi}{\partial \mathbf{u}} = \frac{\partial^2 \Phi}{\partial \mathbf{u}^T \partial \mathbf{u}} \delta \mathbf{u} + \frac{\partial^2 \Phi}{\partial \mathbf{u}^T \partial \boldsymbol{\theta}} \delta \boldsymbol{\theta} \quad (10)$$

which, with the expression for  $\delta \boldsymbol{\theta}$ , allows writing the gradient in terms of  $\delta \mathbf{y}$  and  $\delta \mathbf{u}$ . Hence, the use of the measured values  $\delta \mathbf{y}(t)$  and  $\delta \mathbf{u}(t)$  leads to the following expression for the gradient estimate  $\mathbf{g}(t)$ :

$$\mathbf{g}(t) = \frac{\partial^2 \Phi}{\partial \mathbf{u}^T \partial \boldsymbol{\theta}} \left(\frac{\partial \mathbf{H}}{\partial \boldsymbol{\theta}}\right)^+ \delta \mathbf{y}(t) + \left(\frac{\partial^2 \Phi}{\partial \mathbf{u}^T \partial \mathbf{u}} - \frac{\partial^2 \Phi}{\partial \mathbf{u}^T \partial \boldsymbol{\theta}} \left(\frac{\partial \mathbf{H}}{\partial \boldsymbol{\theta}}\right)^+ \frac{\partial \mathbf{H}}{\partial \mathbf{u}}\right) \delta \mathbf{u}(t) \quad (11)$$

**Gradient from self-optimizing control (SOC):** This method uses the sensitivity of the *optimal* inputs and outputs with respect to the uncertain model parameters. Noting that the optimal inputs  $\mathbf{u}_{opt}$  have to satisfy Equation 10 with  $\frac{\partial \Phi}{\partial \mathbf{u}} = \mathbf{0}$ , and using Equation 9 written for  $\mathbf{y}_{opt}$  and  $\mathbf{u}_{opt}$ , the  $[(p+m) \times q]$  variational matrix becomes:

$$\mathbf{S}_\theta = \begin{bmatrix} \frac{\partial \mathbf{y}_{opt}}{\partial \boldsymbol{\theta}} \\ \frac{\partial \mathbf{u}_{opt}}{\partial \boldsymbol{\theta}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{H}}{\partial \boldsymbol{\theta}} - \frac{\partial \mathbf{H}}{\partial \mathbf{u}} \left(\frac{\partial^2 \Phi}{\partial \mathbf{u}^T \partial \mathbf{u}}\right)^{-1} \frac{\partial^2 \Phi}{\partial \mathbf{u}^T \partial \boldsymbol{\theta}} \\ - \left(\frac{\partial^2 \Phi}{\partial \mathbf{u}^T \partial \mathbf{u}}\right)^{-1} \frac{\partial^2 \Phi}{\partial \mathbf{u}^T \partial \boldsymbol{\theta}} \end{bmatrix} \quad (12)$$

One way to perform SOC, referred to as the null-space method, relies on the computation of the  $[(p+m) \times (p+m-q)]$  matrix  $\mathcal{N}$  that describes the left null space of  $\mathbf{S}_\theta$  (Alstad, Skogestad, 2007). This matrix is guaranteed to exist from the assumption  $p \geq q$ . Since there are  $m$  manipulated variables, it has been proposed to choose the  $m$  controlled variables as  $\mathbf{c} = \mathbf{N}^T \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \end{bmatrix}$ , or  $\delta \mathbf{c} = \mathbf{N}^T \begin{bmatrix} \delta \mathbf{y} \\ \delta \mathbf{u} \end{bmatrix}$  upon subtracting the relationship at the optimal nominal operating point, where  $\mathbf{N}$  is any arbitrary full-rank  $[(p+m) \times m]$  submatrix of  $\mathcal{N}$ . Since  $\mathbf{N}$  lies in the left null space of  $\mathbf{S}_\theta$ , the optimal values of  $\mathbf{c}$  are locally insensitive to the expected uncertainty, and keeping them constant

at their nominal optimal values will reject the effect of uncertainty. For  $p = q$ , these controlled variables represent the gradient (Jaeschke, Skogestad, 2010) and therefore  $\frac{\partial \delta c}{\partial \mathbf{u}} = \mathbf{N}^T \begin{bmatrix} \frac{\partial \mathbf{y}}{\partial \mathbf{u}} \\ \frac{\partial \mathbf{H}}{\partial \mathbf{u}} \end{bmatrix}$  represents an estimate of the Hessian. Hence, choosing

$$\mathbf{P} = \left( \mathbf{N}^T \begin{bmatrix} \frac{\partial \mathbf{H}}{\partial \mathbf{u}} \\ \mathbf{I}_m \end{bmatrix} \right)^{-1} \quad (13)$$

enforces local decoupling between the controlled variables.

Figure 3 illustrates the similarity in philosophy behind NEC and SOC. Input and output measurements are used to compute either the gradient  $\mathbf{g}$  or the CVs  $\delta c$ , which are controlled to zero to enforce optimality.

**Remark:** For  $p = q$  and with only parametric uncertainty, NEC and SOC are equivalent. For  $p > q$ , there are several possible choices of  $\mathbf{N}$ , but this choice is not important to reject expected uncertainty at steady state. However, the convergence paths might be different for NEC and SOC. The situation is different in the presence of process disturbances, say  $\mathbf{d}$ . In this case,  $\mathbf{N}$  will still lie in the left null space of  $\mathbf{S}_\theta$ , but not in the null space of the sensitivity matrix  $\mathbf{S}_d$ . In general, the CVs determined using  $\mathbf{S}_\theta$  will have different sensitivities with respect to  $\mathbf{d}$ , and the steady-state performance of SOC will be affected by the choice of  $\mathbf{N}$ .

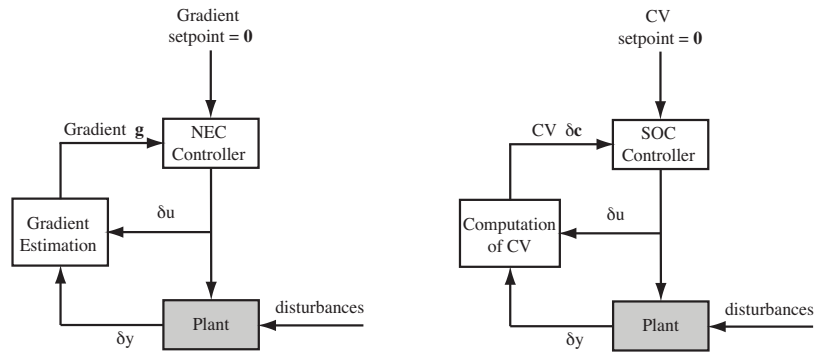


Figure 3. Optimizing control via NEC (left) and SOC (right).

#### 4. Comparison of Implicit RTO Schemes

The goal of this section is to compare six RTO schemes in terms of accuracy, required excitation, convergence properties, implementation issues and sensitivity to measurement noise. The results are given in Table 1 and discussed next.

1. This condition becomes  $p \geq q + m$  if  $\mathbf{S}_\theta$  does not include the term  $\frac{\partial \mathbf{u}_{opt}}{\partial \theta}$ .
2. SOC is affected if  $\mathbf{S}_\theta$  does not include the term  $\frac{\partial \mathbf{u}_{opt}}{\partial \theta}$ . In this case,  $p$  needs to be larger than  $q + m$  and thus increases linearly with  $m$ .



Table 1. Comparison of six gradient estimation techniques (FD: finite difference; ESC: extremum-seeking control; MU: multiple units; AM: adapted model; NEC: neighboring-extremal control; SOC: self-optimizing control)

	Model-free methods			Model-based methods		
	FD	ESC	MU	AM	NEC	SOC
Accurate with plant-model mismatch?	yes	yes	yes	no	no	no
Handles large perturbations?	yes	yes	yes	yes	no	no
Number of outputs	$p = 1$	$p = 1$	$p = 1$	$p \geq q$	$p \geq q$	$p \geq q^1$
Excitation	temporal	temporal	multiple units	none	none	none
Convergence	slow	slow	fast	slow	fast	fast
Scalability with number inputs	slower	even slower	more units	unaffected	unaffected	unaffected <sup>2</sup>

**Plant model:** Model-based methods require a plant model, while no such model is needed for model-free methods. Another important difference between the two classes regards the measurements. Model-free methods require direct measurement of the objective function, whereas the model creates a link between the outputs  $\mathbf{y}$  and the objective function  $J$ , thereby making the measurement of  $J$  unnecessary. Among the different model-based techniques studied here, only the AM scheme uses the model on-line, while NEC and SOC use the model off-line to design the controller.

**Accuracy:** This is clearly an area where the model-free techniques excel. Model-based techniques work well if the uncertainty can be anticipated (the model is structurally correct, there are no process disturbances, and the user knows which parameters can vary). With structural plant-model mismatch or unmeasured process disturbances, convergence will not be to the plant optimum (Chachuat *et al.*, 2009). Furthermore, since NEC and SOC are based on a linearization around the nominal operating point, these two methods give good results around that nominal point, while they cannot cope with large variations.

**Excitation:** In principle, no temporal excitation is necessary for model-based techniques since one can get the gradient information from the  $p$  outputs. However, note that excitation is required to estimate the model parameters in the AM scheme. On the other hand, model-free techniques with  $p = 1$  require excitation to estimate the gradient. Temporal excitation is provided in the FD and ESC schemes, while the use of multiple units provides the necessary excitation in the MU approach.

**Convergence:** Model-based techniques have a clear edge when it comes to convergence speed. Except for MU, all other model-free methods are slow since the

excitation has to respect a time-scale separation and be slower than the system settling time. MU is faster than the other model-free techniques since the dynamics of the units cancel out between similar units. Among the model-based techniques, AM is slower due to the dynamics of the embedded parameter estimation. On the other hand, the gradients in NEC and SOC are computed directly from  $\delta y$  and  $\delta u$ , which makes the techniques fast.

**Implementation issues:** The performance of model-free techniques rely heavily on the tuning parameters. In particular, in addition to the controller gains, the amplitude  $\Delta$  for FD, ESC and MU needs to be selected in such a way that the system is sufficiently excited, while ensuring that the finite-difference approximation is a good estimate of the gradient. Each input has to be perturbed individually for FD and ESC, which limits the use of such methods for multi-input systems. In addition, for FD, constant inputs are applied during transients and reaching steady state has to be detected (Jiang *et al.*, 2002). Despite these difficulties, the FD and ESC schemes have been applied successfully to a wide range of systems in various fields, generally at the level of a single unit. The MU method requires that several identical units be available, with the number of units increasing with the number of inputs. This is the case for processes assembled in stacks such as fuel cells and micro reactors. In contrast, the MU technique has little chance to be applicable for plant-wide optimization. Also, reducing the effect of differences between units is still an issue, although it has been investigated in two recent papers for both the static (Woodward *et al.*, 2009) and dynamic (Reney *et al.*, 2009) cases.

In contrast, model-based techniques do not suffer from these difficulties and are typically more likely to be implemented on large-scale systems as (i) no temporal excitation is necessary, (ii) convergence is relatively fast, and (iii) good scalability with the number of inputs can be ensured. The key issues relate to the presence or absence of a reliable model and the characterization of the main sources of uncertainty. Clearly, in the absence of a reasonably accurate process model, model-based techniques will be of little help, as their accuracy decreases with the size of the uncertainty.

**Sensitivity to measurement noise:** Measurement noise will affect all schemes, though those based on numerical differentiation will be more affected. Noise will also affect the MU method, as adding units also implies adding sensors that may not be identical, thus leading to potentially biased gradient estimates. Furthermore, handling measurement noise for MU is still an open issue. The use of NEC and SOC in the presence of noise is questionable, as noisy measurements are used for gradient estimation. Although this point requires further investigation, it has been shown that NEC should only be considered when the level of noise is small, i.e. the potential cost improvement should not be buried in the measurement noise (Gros *et al.*, 2009). Also, for the case  $p > q$ ,  $(p - q)$  output measurements can be discarded as only  $q$  measurements are required. These  $(p - q)$  outputs can be chosen as the most sensitive to measurement noise as suggested in (Alstad, Skogestad, 2007).

## 5. Illustrative Example

This section illustrates by means of a simple MIMO example the general trends discussed in the previous section. It may well happen that, for other examples, a quantitative comparison would give slightly different results. For instance, in many cases, the ESC method could converge faster than FD. Keep in mind that the goal of this paper is not to rank the six methods on the basis of a single example but rather to illustrate the discussion of Section 4. The example involves the steady-state optimization of an isothermal continuous stirred-tank reactor, where the reactions  $A+B \rightarrow C$ ,  $2B \rightarrow D$  take place. There are two manipulated variables, the feed rates of  $A$  and  $B$ . The goal is to maximize the productivity of  $C$ .

### 5.1. Problem Formulation

The problem can be formulated mathematically as follows:

$$\max_{u_A, u_B} J = \frac{c_C^2 (u_A + u_B)^2}{u_A c_{Ain}} - w(u_A^2 + u_B^2) \quad (14)$$

$$\dot{c}_A = -k_1 c_A c_B + \frac{u_A}{V} c_{Ain} - \frac{u_A + u_B}{V} c_A \quad c_A(0) = c_{A,s} \quad (15)$$

$$\dot{c}_B = -k_1 c_A c_B - 2 k_2 c_B^2 + \frac{u_B}{V} c_{Bin} - \frac{u_A + u_B}{V} c_B \quad c_B(0) = c_{B,s} \quad (16)$$

$$\dot{c}_C = k_1 c_A c_B - \frac{u_A + u_B}{V} c_C \quad c_C(0) = c_{C,s} \quad (17)$$

$$\dot{c}_D = k_2 c_B^2 - \frac{u_A + u_B}{V} c_D \quad c_D(0) = c_{D,s} \quad (18)$$

where  $c_X$  denotes the concentration of species  $X$  and  $c_{X,s}$  the corresponding steady-state value,  $V$  is the reactor volume,  $u_A$  and  $u_B$  are the feed rates of  $A$  and  $B$ ,  $c_{Ain}$  and  $c_{Bin}$  are the inlet concentrations,  $k_1$  and  $k_2$  are the rate constants of the two chemical reactions, and  $w$  a weighting parameter. The numerical values of the model parameters are given in Table 2. Two scenarios are investigated. In Scenario A, the plant differs from the model by the values of the rate constants,  $k_{1plant} = 1.4 \frac{1}{\text{mol min}}$  and  $k_{2plant} = 0.4 \frac{1}{\text{mol min}}$ . Hence, there is only parametric uncertainty, and the vector of uncertain parameters is known to be  $\theta = [k_1 \ k_2]^T$ . In scenario B, there is, in addition, the possibility of having an unexpected (unmodeled and unmeasured) disturbance in the form of a step variation of  $c_{Ain}$ , with  $c_{Ain,plant} = 2.5 \frac{\text{mol}}{1}$ . Note also that the values of the plant rate constants and of the disturbance are not known to the RTO schemes. The plant settling time  $\tau$  is approximately 50 min. This example has 2 manipulated feed rates ( $m = 2$ ), 4 measured concentrations ( $p = 4$ ) and two uncertain kinetic parameters ( $q = 2$ ).

Table 2. Nominal model parameters

$k_1$	0.75	$\frac{1}{\text{mol}\cdot\text{min}}$	$c_{Ain}$	2	$\frac{\text{mol}}{1}$
$k_2$	1.5	$\frac{1}{\text{mol}\cdot\text{min}}$	$c_{Bin}$	1.5	$\frac{\text{mol}}{1}$
$V$	500	1	$w$	0.004	$\frac{\text{mol}\cdot\text{min}}{1^2}$

### 5.2. Performance Comparison of Six Implicit RTO Schemes

The adaptation gains for the various schemes are 0.003, 0.001, 0.02, 0.1, 1 and  $1 \frac{l}{min}$  for FD, ESC, MU, AM, NEC and SOC, respectively. These gains were tuned manually and chosen rather aggressively, that is, somewhat larger values would either induce large oscillations without leading to significant reduction in convergence time, or worse, would not guarantee convergence.

For gradient control, all methods use  $\mathbf{P} = \left( \frac{\partial^2 \mathbf{J}}{\partial \mathbf{u}^T \partial \mathbf{u}} \right)^{-1}$  evaluated at the optimal nominal operating point, except for SOC that uses [13], which is also an estimate of the inverse of the Hessian for  $p = q$ . All model-free methods use  $\Delta = 0.4 \frac{l}{min}$ . The ESC scheme uses  $\omega_1 = \frac{2\pi}{150}$ ,  $\omega_2 = \frac{2\pi}{200}$ , and  $\alpha = \beta_1 = \beta_2 = \frac{1}{200} \text{min}^{-1}$ . These frequencies were chosen in such a way that they respect a time-scale separation with the natural frequency of the system, as suggested in (Krstic, Wang, 2000). The parameter estimation uses the gain  $k_\theta = 1 \text{min}^{-1}$ . Table 3 summarizes the results obtained with the various approaches in terms of both convergence time and accuracy for two situations, namely without and with unexpected disturbance in  $c_{Ain}$ . Figure 4 depicts the time evolution of the normalized cost for the six schemes when the disturbance is present. Note that MU has been plotted together with model-based techniques because of its fast convergence. Note also that, although it has been normalized with respect to the steady-state plant optimal cost, the running cost can have values larger than 1 before steady state is reached.

Table 3. Convergence time, convergence time relative to the plant settling time, and accuracy of the six RTO schemes, without (scenario A) and with unexpected disturbance in  $c_{Ain}$  (scenario B).

	Convergence time [min]		Relative convergence time		Optimality loss [%]	
	A	B	A	B	A	B
No adaptation	-	-	-	-	19	26
FD	1200	1200	$24\tau$	$24\tau$	0.2	0.3
ESC	3000	4000	$60\tau$	$80\tau$	0.6	0.5
MU	150	150	$3\tau$	$3\tau$	0	0
AM	75	75	$1.25\tau$	$1.25\tau$	0	7
NEC	45	50	$0.9\tau$	$\tau$	0.5	6
SOC	45	60	$0.9\tau$	$1.2\tau$	0.8	18

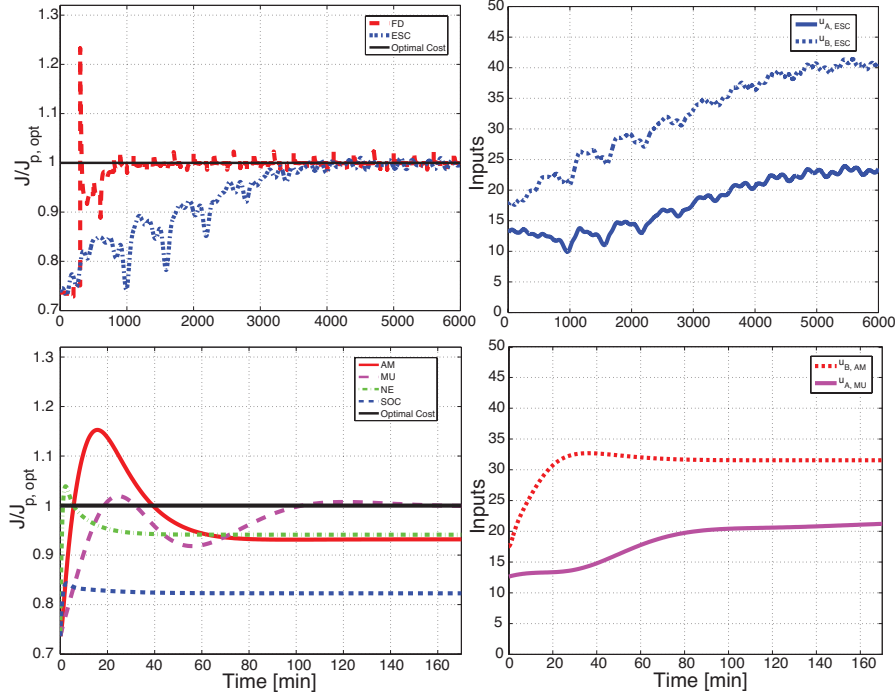


Figure 4. Evolution of the cost ratio  $\frac{J(t)}{J_{p,opt}}$  and of the inputs for six RTO schemes in the presence of an unexpected disturbance in  $c_{Ain}$ . Note the large difference in time scales between FD and ESC on the one side, and the other schemes on the other. For the sake of clarity, only the four input profiles with the widest ranges are plotted.

As seen from Table 3, all methods are quite accurate in the absence of unexpected disturbances, although FD and ESC exhibit slow convergence. Optimality loss is nearly zero with the MU and AM schemes, while small errors persist with NEC and SOC because of the linearization introduced in the control design step. For  $p > q$ , the observed small difference between NEC and SOC can be explained by the different paths that the two schemes follow to convergence. In terms of convergence time, AM is slightly inferior to both NEC and SOC due to the time taken for parameter estimation.

The situation is quite different in the presence of the unexpected disturbance in  $c_{Ain}$ . In this example, NEC outperforms SOC, but it could also happen that, for a specific choice of CVs, the performance of SOC would be better (for instance if the CVs happen to be in the null space of  $S_d$ ). The model-free methods FD, ESC and MU are able to reject the effect of both parametric uncertainty and unexpected disturbances. This is due to the fact that these methods use only measurements and not a (possibly inaccurate) plant model. The price to pay for this improved accuracy

is slower convergence. Note that MU is the method with the best overall performance, as the optimal solution can be reached in  $3\tau$  with nearly perfect accuracy.

## 6. Conclusions

This paper has investigated the use of measurements to drive a plant towards optimality in the presence of both expected and unexpected uncertainty. Six implicit schemes that implement optimality via gradient control have been considered; three of them are model-free (FD, ESC and MU) and rely exclusively on measurements, whereas the other three are model-based and rely both on a process model and measurements to estimate the plant gradient and compute the optimal inputs. Although some of these techniques could be combined (for instance, all techniques could be modified to benefit from the presence of multiple units or the possibility of repeating the optimization), the schemes have been investigated in their original formulation so as to focus the discussion around the strengths and weaknesses of the individual approaches.

It has been observed that model-based techniques based on linearization are able to quickly reject the effect of expected parametric uncertainty. The AM method exhibits the slowest convergence, while NEC and SOC are very similar. However, the AM scheme is appropriate when the uncertainty is large since, in contrast to NEC and SOC, there is no underlying linearization. All three model-based techniques require the availability of at least as many independent output measurements as there are uncertain parameters in order to compute the gradients. The slow convergence of the AM scheme is due to the dynamics of the parameter estimation, whereas NEC and SOC use input and output measurements to infer the gradients.

In contrast, model-free techniques are able to reject the effect of both expected and unexpected disturbances. However, gradient computation requires input excitation, which is of temporal nature in the FD and ESC schemes, thus requiring a time-scale separation, which penalizes the rate of convergence. This even worsens when the number of inputs increases, as each input has to be excited individually. In contrast, the MU scheme does not require time-scale separation as the excitation is not temporal but between units. In comparison, model-based techniques do scale better with the number of inputs. Note also that, regardless of the number of uncertain parameters, model-free techniques only need the cost measurement, while, for model-based techniques, an increase in the number of uncertain parameters requires more output measurements. The only technique that combines the advantages of model-free and model-based techniques is the MU approach. A single measurement suffices and fast convergence is obtained. However, MU relies on the strong assumption that several identical units are available. Since the number of units increases with the number of inputs, this assumption will tend to become unrealistic for systems with a large number of inputs.

This paper has investigated and compared the basic features of various model-free and model-based RTO schemes. Future work needs to include an analysis of

the impact of measurement noise, although it is quite clear that techniques based on numerical differentiation will be more affected.

## References

- Alstad V., Skogestad S. (2007). Null space method for selecting optimal measurement combinations as controlled variables. *Ind. Eng. Chem. Res.*, Vol. 46, No. 3, pp. 846-853.
- Ariyur K., Krstic M. (2003). *Real-time optimization by extremum-seeking control*. New York, John Wiley.
- Box G. E. P., Draper N. R. (1987). *Empirical model-building and response surfaces*. New York, John Wiley.
- Boyd S., Vandenberghe L. (2004). *Convex optimization*. U.K., Cambridge University Press.
- Chachuat B., Srinivasan B., Bonvin D. (2009). Adaptation strategies for real-time optimization. *Comput. Chem. Engng.*, Vol. 33, No. 10, pp. 1557-1567.
- Francois G., Srinivasan B., Bonvin D. (2005). Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty. *J. Process Contr.*, Vol. 15, No. 6, pp. 701-712.
- Gros S., Srinivasan B., Bonvin D. (2009). Optimizing control based on output feedback. *Comput. Chem. Engng.*, Vol. 33, No. 1, pp. 191-198.
- Jaeschke J., Skogestad S. (2010). Self-optimizing control and NCO tracking in the context of real-time optimization. In *Int. symp. dycops*, p. 123-128. Leuven, Belgium.
- Jiang T., Chen B., He X., Stuart P. (2002). Application of steady-state detection method based on wavelet transform. *Comput. Chem. Engng.*, Vol. 27, pp. 569-578.
- Krstic M., Wang H.-H. (2000). Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica*, Vol. 36, pp. 595-601.
- Ljung L. (1999). *System identification – Theory for the user (2nd ed.)*. Upper Saddle River, N.J., Prentice Hall.
- Marchetti A., Chachuat B., Bonvin D. (2009). Modifier-adaptation methodology for real-time optimization. *Ind. Eng. Chem. Res.*, Vol. 48, pp. 6022-6033.
- Marlin T., Hrymak A. (1997). Real-time operations optimization of continuous processes. In *AIChE symposium series - CPC-V*, Vol. 93, p. 156-164.
- Reney F., Perrier M., Srinivasan B. (2009). Influence of differences in system dynamics in the context of multi-unit optimization. In *Int. symp. ADCHEM*, p. 557-562. Istanbul (Turkey).
- Rotava O., Zanin A. C. (2005). Multivariable control and real-time optimization – An industrial practical view. *Hydrocarbon Processing*, Vol. 84, No. 6, pp. 61-71.
- Skogestad S. (2000). Plantwide control: The search for the self-optimizing control structure. *J. Process Contr.*, Vol. 10, pp. 487-507.
- Srinivasan B. (2007). Real-time optimization of dynamic systems using multiple units. *Int. J. Robust Nonlinear Control*, Vol. 17, pp. 1183-1193.
- Woodward L., Perrier M., Srinivasan B. (2009). Improved performance in the multi-unit optimization method with non-identical units. *J. Process Contr.*, Vol. 19, No. 2, pp. 205-215.