

# Iterative Controller Tuning by Real-Time Optimization

Gene A. Bunin, Grégory François, Dominique Bonvin \*

\* *Laboratoire d'Automatique, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne (e-mail: dominique.bonvin@epfl.ch).*

**Abstract:** The present article looks at the problem of iterative controller tuning, where the parameters of a given controller are adapted in an iterative manner to bring a user-defined performance metric to a local minimum for some repetitive process. Specifically, we cast the controller tuning problem as a real-time optimization (RTO) problem, which allows us to exploit the available RTO theory to enforce both convergence and performance guarantees. We verify the effectiveness of the proposed methodology on an experimental torsional system and note that the results are particularly promising considering the simplicity of the method.

*Keywords:* Optimization; Control system design; Iterative methods; Self-tuning control

## 1. INTRODUCTION

We consider the problem of iteratively tuning the parameters  $\boldsymbol{\rho} \in \mathbb{R}^{n_\rho}$  of some general control law,  $G(\boldsymbol{\rho})$ , so as to achieve optimal control performance for a repetitive setpoint trajectory. Concretely,  $G(\boldsymbol{\rho})$  can take the form of any of the controllers in use today, as we can easily use it to express the PID, a general fixed-order controller, or even the optimization-based model-predictive control (MPC).

Iterative tuning usually follows the initial controller design stage, where the latter may be carried out in a number of ways and may be characterized based on, for example, whether it is data-driven or model-based, or whether or not it uses simple heuristics (e.g., Ziegler-Nichols) or more advanced design methodologies. The goal of the design stage is often to find an initial set  $\boldsymbol{\rho}_0$  that meets certain stability criteria while at the same time obtaining good performance. However, regardless of the method used, it is a general fact that  $\boldsymbol{\rho}_0$  will not be the best choice in actual application due to:

- assumptions made at the design stage (e.g., linearity, time invariance),
- modeling errors and simplifications,
- conservatism of a robust design,
- practical arguments that favor a rough, quick design over a rigorous method.

When the process is “repetitive”, the control task is usually the same from one realization (one “run”, “batch”, “cycle”, or “window”) to another, and some sort of learning may be used to adapt  $\boldsymbol{\rho}$  from one run to the next so as to reduce the suboptimality of  $\boldsymbol{\rho}_0$  (Fig. 1) by taking advantage of the fact that the resulting differences in performance will be, on average, a solely deterministic reflection of the change in  $\boldsymbol{\rho}$ . While there are several classes of methods for carrying out such adaptations, the focus in this paper will be on the optimization-based approach of minimizing directly some controller performance metric as an unknown function of the parameters. Here, the choice of

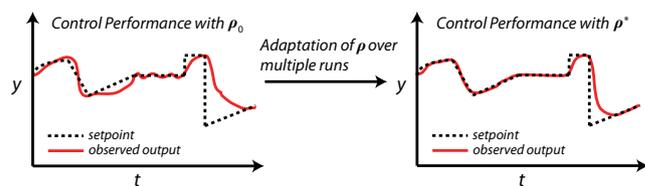


Fig. 1. The basic premise of iterative controller tuning.

the metric may vary depending on the strategy used – e.g., it may be a correlation function in correlation-based tuning (Karimi et al., 2004) or a general performance criterion that is a weighted sum of subcriteria (e.g., tracking error or control effort) in iterative feedback tuning (Hjalmarsson et al., 1998) or extremum-seeking methods (Killingsworth and Krstić, 2006). A standard optimization method for these algorithms, which attempts to find parameters  $\boldsymbol{\rho}^*$  that locally minimize the performance metric, is the gradient descent, with the way in which the gradient is estimated representing an additional trait that differentiates one method from another.

Our contribution in this work is to generalize this class of methods by treating them in the real-time optimization (RTO) framework, where the minimization of an unknown function is a standard problem (see, e.g., Box and Draper (1969), Brdys and Tatjewski (2005), Conn et al. (2009), Myers et al. (2009)), and where recent contributions by the authors (Bunin et al., 2013d,e) have provided a unified methodology for solving such a problem with optimality guarantees. As will be shown, such a generalization presents many new possibilities. Namely, one is not forced to use a gradient-descent method in adapting  $\boldsymbol{\rho}$ , constraints on closed-loop stability may be incorporated directly into the RTO problem, and the issue of robustness against gradient uncertainty may be handled in a relatively simple manner.

The formulation of the tuning problem in the RTO framework and the performance guarantees that are possible

therein are discussed in Sections 2 and 3, respectively. Section 4 then takes the reader through an experimental case study of a laboratory torsional system where a fixed-order controller with three degrees of freedom is tuned by the proposed method to achieve better performance. We conclude the paper in Section 5.

## 2. RTO-BASED ITERATIVE CONTROLLER TUNING

In order to apply RTO theory and methods to the tuning problem, a basic repeatability assumption, previously stated in Bunin et al. (2012) for the problem of run-to-run MPC tuning, is required.

*Assumption 1.* Let  $J_k$  denote the observed control performance at run  $k$ . Process *repeatability* implies that:

$$J_k = J(\boldsymbol{\rho}_k) + \delta_k, \quad (1)$$

where  $\delta_k$  is an unknown stochastic element while  $J : \mathbb{R}^{n_\rho} \rightarrow \mathbb{R}$  is purely deterministic.

Assumption 1 has two crucial characteristics. First, it makes the very idea of improving (minimizing)  $J_k$  sensible, as it guarantees that applying a given set  $\boldsymbol{\rho}$  will always return the same deterministic component  $J(\boldsymbol{\rho})$ , together with some “non-repeatable noise” component  $\delta_k$  that is independent of  $\boldsymbol{\rho}$ . Assuming  $J(\boldsymbol{\rho})$  to be differentiable, it follows that  $\nabla J(\boldsymbol{\rho}) = \mathbf{0}$  implies first-order optimality of the observed performance  $J_k$ .

The other important aspect of (1) is that it renders any additional assumptions on the system and controller structure unnecessary. The control law  $G(\boldsymbol{\rho})$  may be parameterized in almost any way – in MPC, for example,  $\boldsymbol{\rho}$  may correspond to the parameters of an optimization problem, and may affect control performance in a highly nontrivial manner (Bunin et al., 2012). Likewise, the process being controlled may itself be nonlinear and of unknown structure. An alluring trait of Assumption 1 is that it lumps all of these analytically unknown (but deterministic) characteristics into the single function  $J(\boldsymbol{\rho})$ .

Some caution is necessary, however, as Assumption 1 is only an approximation of reality, the validity of which depends on the amount of non-repeatable effects in the system, which may originate from measurement noise, equipment degradation, and disturbances. In the ideal case where such effects are completely absent,  $\delta_k = 0$  since applying the same  $\boldsymbol{\rho}$  will always yield the same input/output trajectories and thus the same performance. When these effects are present but minor, we expect their influence on the controller performance to be small, thereby leading to approximately the same performance for the same  $\boldsymbol{\rho}$  (with the variations approximated by the stochastic term  $\delta$ ). If the non-repeatable effects become significant, Assumption 1 will no longer be valid and the methods proposed in this work may not be appropriate.

Supposing hereafter that the repeatability assumption is satisfied, we may minimize  $J_k$  by minimizing  $J$ , and so the task of iterative controller tuning may be formulated in RTO form as:

$$\begin{aligned} & \underset{\boldsymbol{\rho}}{\text{minimize}} && J(\boldsymbol{\rho}) \\ & \text{subject to} && \mathbf{R}(\boldsymbol{\rho}) \preceq \mathbf{0} \text{ ,} \\ & && \underline{\boldsymbol{\rho}} \preceq \boldsymbol{\rho} \preceq \bar{\boldsymbol{\rho}} \end{aligned} \quad (2)$$

where  $\mathbf{R}(\boldsymbol{\rho})$  may be a set of model-based stability constraints, with  $\underline{\boldsymbol{\rho}}$  and  $\bar{\boldsymbol{\rho}}$  representing lower and upper limits, respectively, on the tunable parameters. Note that any general RTO algorithm of the form

$$\boldsymbol{\rho}_{k+1}^* = \Gamma(\boldsymbol{\rho}_0, \dots, \boldsymbol{\rho}_k, J_0, \dots, J_k) \quad (3)$$

may be used to solve (2), with  $\Gamma$  some law that uses past data to generate the next “best” iterate,  $\boldsymbol{\rho}_{k+1}^*$ . Usually, the solution of (3) is filtered by an adaptable step  $K_k$  (Brdys and Tatjewski, 2005):

$$\boldsymbol{\rho}_{k+1} = \boldsymbol{\rho}_k + K_k (\boldsymbol{\rho}_{k+1}^* - \boldsymbol{\rho}_k), \quad (4)$$

and the calculated parameters  $\boldsymbol{\rho}_{k+1}$  are used in the next run.

We remark that choosing  $\Gamma(\cdot) = \boldsymbol{\rho}_k - \nabla J(\boldsymbol{\rho}_k)$  and  $K_k = 1/k$  yields the well-known gradient descent method with a diminishing step size, but it should be evident that better performing schemes that employ *all* of the previous data may be used instead. In fact, any RTO method, from a simple, gradient-free approach like evolutionary operation (Box and Draper, 1969) to advanced model-based methods (Brdys and Tatjewski, 2005; Marchetti et al., 2010) may be used to minimize  $J$ .

Although the methodology for solving (2) with theoretical guarantees exists (Bunin et al., 2013d), we choose here to work with the simpler unconstrained problem of

$$\underset{\boldsymbol{\rho}}{\text{minimize}} J(\boldsymbol{\rho}) \text{ ,} \quad (5)$$

as this allows us to state the relevant RTO theory without going into the full detail of the constrained case. An in-depth treatment of the latter may be found in Bunin et al. (2013b).

## 3. GUARANTEES AND PRACTICAL CONSIDERATIONS

When applying an RTO algorithm to autotune a controller, it is desired that the algorithm in question have some guarantee of convergence and optimality. As such, we will now outline the rigorous conditions for finding  $\boldsymbol{\rho}^*$  such that the stationarity condition  $\nabla J(\boldsymbol{\rho}^*) \approx \mathbf{0}$  is satisfied, as well as discuss the practical aspects that must be considered when applying these conditions.

### 3.1 Conceptual Convergence Conditions

We proceed to derive the sufficient conditions for the convergence of an RTO algorithm (3) to a local stationary point of (5). The following assumption on  $J$  is required:

*Assumption 2.* Let  $J$  be twice continuously differentiable, coercive, and have bounded second derivatives  $\forall \boldsymbol{\rho} \in \mathbb{R}^{n_\rho}$ .

This allows us to state the following lemma and theorem: *Lemma 1.* (Quadratic Upper Bound). There exists a diagonal matrix  $\bar{\mathbf{Q}} \succ 0$  such that:

$$\begin{aligned} J(\boldsymbol{\rho}_{k+1}) - J(\boldsymbol{\rho}_k) & \leq \nabla J(\boldsymbol{\rho}_k)^T (\boldsymbol{\rho}_{k+1} - \boldsymbol{\rho}_k) \\ & + \frac{1}{2} (\boldsymbol{\rho}_{k+1} - \boldsymbol{\rho}_k)^T \bar{\mathbf{Q}} (\boldsymbol{\rho}_{k+1} - \boldsymbol{\rho}_k), \quad \forall \boldsymbol{\rho}_k, \boldsymbol{\rho}_{k+1}. \end{aligned} \quad (6)$$

**Proof.** The proof follows from Assumption 2 and the mean-value theorem (Bunin et al., 2013d). ■

*Theorem 1.* (Sufficient Conditions for Convergence to a Stationary Point). Consider the RTO algorithm  $\rho_{k+1}^* = \Gamma(\cdot)$  and filter law (4) that either satisfy the conditions:

$$\nabla J(\rho_k)^T(\rho_{k+1}^* - \rho_k) \leq -\delta_J, \quad \delta_J > 0 \quad (7)$$

$$K_k = -\gamma \frac{\nabla J(\rho_k)^T(\rho_{k+1}^* - \rho_k)}{(\rho_{k+1}^* - \rho_k)^T \bar{\mathbf{Q}}(\rho_{k+1}^* - \rho_k)}, \quad 0 < \gamma < 2 \quad (8)$$

$$-\Delta \bar{\rho}^* \preceq \rho_{k+1}^* - \rho_k \preceq \Delta \bar{\rho}^* \quad (9)$$

when this is possible or yield  $\rho_{k+1} := \rho_k$  otherwise.

It follows that such an algorithm will:

- (i) Converge to a point  $\rho^*$  with:

$$\|\nabla J(\rho^*)\|_1 < \frac{\delta_J}{\min \Delta \bar{\rho}^*}. \quad (10)$$

- (ii) Converge monotonically in no more than:

$$\frac{\left[ \min_{\rho} J(\rho) - J(\rho_0) \right] (\Delta \bar{\rho}^*)^T \bar{\mathbf{Q}} \Delta \bar{\rho}^*}{(0.5\gamma^2 - \gamma)\delta_J^2} \quad (11)$$

iterations.

**Proof.**

- (i) Only (7) is in danger of not being satisfied at  $\rho^*$ , which may only occur when:

$$\nabla J(\rho^*)^T(\rho_{k+1}^* - \rho^*) > -\delta_J \quad (12)$$

$$\forall \rho_{k+1}^* \in \rho^* \pm \Delta \bar{\rho}^*,$$

or, equivalently, when:

$$\begin{aligned} & \min_{\rho_{k+1}^* \in \rho^* \pm \Delta \bar{\rho}^*} \nabla J(\rho^*)^T(\rho_{k+1}^* - \rho^*) > -\delta_J \\ \Leftrightarrow & \nabla J(\rho^*)^T \Delta \bar{\rho}^* > -\delta_J, \quad (13) \\ \Leftrightarrow & -\nabla J(\rho^*)^T \Delta \bar{\rho}^* < \delta_J \end{aligned}$$

where we have assumed, without loss of generality, that all of the elements in  $\nabla J(\rho^*)$  are negative. We complete the proof by noting that:

$$\begin{aligned} & \|\nabla J(\rho^*)\|_1 \min \Delta \bar{\rho}^* \leq -\nabla J(\rho^*)^T \Delta \bar{\rho}^* \\ \Rightarrow & \|\nabla J(\rho^*)\|_1 < \frac{\delta_J}{\min \Delta \bar{\rho}^*}. \quad (14) \end{aligned}$$

- (ii) Substituting the filter law (4) into the bound (6) and setting  $K_k$  as given in (8) yields:

$$\begin{aligned} J(\rho_{k+1}) - J(\rho_k) & \leq \\ & \frac{(0.5\gamma^2 - \gamma) [\nabla J(\rho_k)^T(\rho_{k+1}^* - \rho_k)]^2}{(\rho_{k+1}^* - \rho_k)^T \bar{\mathbf{Q}}(\rho_{k+1}^* - \rho_k)}, \quad (15) \end{aligned}$$

which achieves its worst-case maximal value for  $[\nabla J(\rho_k)^T(\rho_{k+1}^* - \rho_k)] = -\delta_J$  and  $(\rho_{k+1}^* - \rho_k)^T \bar{\mathbf{Q}}(\rho_{k+1}^* - \rho_k) = (\Delta \bar{\rho}^*)^T \bar{\mathbf{Q}} \Delta \bar{\rho}^*$ :

$$J(\rho_{k+1}) - J(\rho_k) \leq \frac{(0.5\gamma^2 - \gamma)\delta_J^2}{(\Delta \bar{\rho}^*)^T \bar{\mathbf{Q}} \Delta \bar{\rho}^*} < 0. \quad (16)$$

This proves that there will be a monotonic improvement from iteration to iteration. Since  $J$  is coercive by assumption, it must achieve its global minimum on  $\mathbb{R}^{n_\rho}$ , thereby allowing us to divide the largest possible initial suboptimality gap  $J(\rho_0) - \min_{\rho} J(\rho)$  by

the worst-case improvement to obtain the maximal possible number of iterations for which Conditions (7) and (8) can be met, as given in (11). Meeting the conditions for more than the number of iterations given by (11) would guarantee decreasing the cost past its global minimum, which is impossible. ■

Conditions (7) and (8) may thus be viewed as the *conceptual* sufficient conditions to bring  $\nabla J(\rho)$  close to  $\mathbf{0}$  in some finite number of iterations, where the degree of accuracy and the worst-case convergence time are functions of  $\delta_J$ ,  $\gamma$ , and  $\Delta \bar{\rho}^*$ , which may be set by the user.

### 3.2 Implementation Issues

While we do not consider Assumption 2 as being practically limiting, the proper choice of  $\bar{\mathbf{Q}}$  may be. When a model of the system that captures the system's nonlinear trends is available, one may estimate  $\bar{\mathbf{Q}}$  by numerical trials and then, for example, multiply this estimate by an additional safety factor (Bunin et al., 2013c, Sec. 4.1.2). Alternatively, experimental data and past measurements of  $J_k$  may be used to construct an estimate of  $\bar{\mathbf{Q}}$  either offline or online.

Another crucial difficulty arises when we want to satisfy the sufficient conditions despite having only an estimate of the gradient  $\nabla \hat{J}(\rho_k)$ , as  $\nabla \hat{J}(\rho_k)^T(\rho_{k+1}^* - \rho_k) \leq -\delta_J \not\Rightarrow \nabla J(\rho_k)^T(\rho_{k+1}^* - \rho_k) \leq -\delta_J$ . Assuming that the elements in  $\nabla \hat{J}(\rho_k)$  are well scaled and defining  $\sigma = \mathbf{1}$ , we may add robustness to the sufficient conditions by building an uncertainty box,  $\mathcal{E}_k$ , around  $\nabla \hat{J}(\rho_k)$ :

$$\mathcal{E}_k = \left\{ \nabla \tilde{J}(\rho_k) : \nabla \hat{J}(\rho_k) - m_k \sigma \preceq \nabla \tilde{J}(\rho_k) \preceq \nabla \hat{J}(\rho_k) + m_k \sigma \right\}, \quad (17)$$

with  $m_k \geq 0$  a scalar used to define the size of the box. We then attempt to satisfy the robust versions of Conditions (7) and (8):

$$\max_{\nabla \tilde{J}(\rho_k) \in \mathcal{E}_k} \nabla \tilde{J}(\rho_k)^T(\rho_{k+1}^* - \rho_k) \leq -\delta_J \quad (18)$$

$$K_k = -\gamma \frac{\max_{\nabla \tilde{J}(\rho_k) \in \mathcal{E}_k} \nabla \tilde{J}(\rho_k)^T(\rho_{k+1}^* - \rho_k)}{(\rho_{k+1}^* - \rho_k)^T \bar{\mathbf{Q}}(\rho_{k+1}^* - \rho_k)}, \quad (19)$$

which have a tractable reformulation and may be handled without introducing any computational difficulties (Bunin et al., 2013e). If the true gradient  $\nabla J(\rho_k) \in \mathcal{E}_k$ , then it is clear that (7) and (8) will be satisfied as well. Increasing  $m_k$  represents adding robustness at the price of speed, as the numerator of (19) will simply approach 0 due to growing  $\mathcal{E}_k$ , thereby leading to smaller steps and slower progress. Increasing  $m_k$  too much will lead to infeasibility in (18) as  $\mathcal{E}_k$  cannot be made arbitrarily large (the condition cannot hold for all possible gradients).

Finally, some additional constraints on the steps  $\rho_{k+1} - \rho_k$  are generally desired in practice. As RTO schemes often try to estimate the gradient from past measurements (Brdys and Tatjewski, 2005, Sec. 4.3-4.4), (Marchetti et al., 2010) and as many of these estimations are finite-difference based, it is of interest to maintain a sufficient minimal gap between consecutive sets of parameters so as to help reject

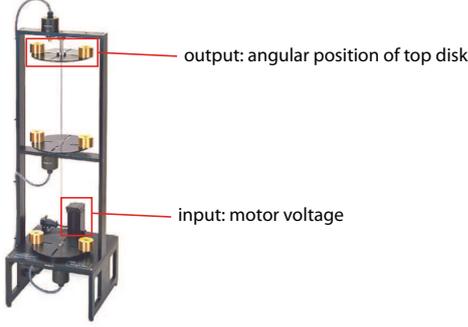


Fig. 2. The ECP 205 torsional system.

the noise term  $\delta_k$ . Additionally, since improper choices of  $\bar{\mathbf{Q}}$  or bad estimates of  $\nabla \hat{J}(\boldsymbol{\rho}_k)$  may lead to an adaptation that would *increase*  $J$  (or even make the closed-loop system detrimentally unstable), some maximum allowable change in the parameters from one run to the next is also desired. These two requirements may be summarized as:

$$\Delta \boldsymbol{\rho} \leq \|\boldsymbol{\rho}_{k+1} - \boldsymbol{\rho}_k\|_\infty \leq \Delta \bar{\boldsymbol{\rho}}, \quad (20)$$

and may be met by simply tuning  $K_k$ . Note, however, that the lower constraint may force (8) to not be met, whereas the upper constraint may lead to slower progress. As such, the (*ad hoc*) choices of  $\Delta \underline{\boldsymbol{\rho}}$  and  $\Delta \bar{\boldsymbol{\rho}}$  – guided by the size of  $\delta_k$  and our uncertainty in  $\bar{\mathbf{Q}}$  and  $\nabla \hat{J}(\boldsymbol{\rho}_k)$  – should be as small and large, respectively, as possible.

#### 4. APPLICATION TO A TORSIONAL SYSTEM

We illustrate the proposed ideas on an experimental torsional system in our laboratory.

##### 4.1 Experimental Setup

The physical system, pictured in Fig. 2, consists of a motor and three disks in series along a torsionally flexible shaft, with the motor generating a torque that rotates the shaft. Up to four removable weights may be placed on each of the disks to vary the inertia of the system – the configuration with all four weights on each disk is used here. A Macintosh PC with a National Instruments’ LabVIEW<sup>®</sup> interface and data acquisition system is used to adjust the voltage to the motor and to collect the angular position measurements of the top and middle disks with a sampling time of 60 ms. A detailed description of the system is available from the vendor (Educational Control Products, 2008).

##### 4.2 Formulation of the Controller Tuning Problem

We consider a single-input-single-output control problem, treating the motor voltage as the input and the angular position of the top disk as the output (Fig. 2). Two different repetitive setpoint trajectories are considered:

$$\begin{aligned} y_{d,1}(t, A) &= A \operatorname{sign} \left( \sin \frac{\pi t}{6} \right), \\ y_{d,2}(t, A) &= -A \cos \frac{\pi t}{6} \end{aligned}, \quad (21)$$

with  $t$  expressed in seconds and  $A$  denoting the amplitude that is set to either 1 or 2 radians, thereby creating a

total of four test cases.  $y_{d,1}(t, A)$  is a square wave while  $y_{d,2}(t, A)$  is a sinusoid, both with a period of 12 s that corresponds, essentially, to one “run”. An adaptation of the controller parameters is carried out in the sampling time gap between each pair of consecutive periods.

The controller function  $G(\boldsymbol{\rho})$  takes the form of a fixed-order discrete-time transfer function, where the denominator coefficients have been fixed (by prior pole placement) and the numerator coefficients are adapted:

$$G(\boldsymbol{\rho}) = \frac{\rho_1 z^2 + \rho_2 z + \rho_3}{z^2 + z + 0.5}. \quad (22)$$

For simplicity, we only consider the tracking error in the performance metric:

$$J_k := \sum_{T=200k}^{200(k+1)-1} [y_d(0.06T, A) - y(0.06T)]^2, \quad (23)$$

with  $T$  used to denote the (absolute) sample counter and  $k = 0, 1, 2, \dots$  the adaptation iteration counter.

##### 4.3 Choice and Configuration of RTO Algorithm

As only 60 ms are available to evaluate  $J_k$ , to obtain a gradient estimate  $\nabla \hat{J}(\boldsymbol{\rho}_k)$ , and to calculate the new parameters  $\boldsymbol{\rho}_{k+1}$ , we choose the (estimated) gradient descent with a unit step size for the RTO adaptation, so that  $\boldsymbol{\rho}_{k+1}^* := \boldsymbol{\rho}_k - \nabla \hat{J}(\boldsymbol{\rho}_k)$ . This is done not only because the computational requirements of such an update are minimal, but also because Condition (7) – or, equivalently, (18) for  $m_k := 0$  – is satisfied by construction, with  $-\nabla \hat{J}(\boldsymbol{\rho}_k)^T \nabla \hat{J}(\boldsymbol{\rho}_k) \leq -\delta_J$ , where the value of  $\delta_J$  is implicitly inferior to  $\|\nabla \hat{J}(\boldsymbol{\rho}_k)\|_2^2$ .

$\nabla \hat{J}(\boldsymbol{\rho}_k)$  is obtained from a least-squares regression of a quadratic model (without interaction terms) for the full set of measured data up to run  $k$ :

$$J_k \approx a_{11}\rho_{k,1}^2 + a_{22}\rho_{k,2}^2 + a_{33}\rho_{k,3}^2 + a_1\rho_{k,1} + a_2\rho_{k,2} + a_3\rho_{k,3} + a_0 \quad (24)$$

$$\nabla \hat{J}(\boldsymbol{\rho}_k) := \begin{bmatrix} 2a_{11}\rho_{k,1} + a_1 \\ 2a_{22}\rho_{k,2} + a_2 \\ 2a_{33}\rho_{k,3} + a_3 \end{bmatrix}. \quad (25)$$

Such an estimation, though fairly rudimentary in that it supposes the noise term  $\delta_k$  to be white Gaussian and the performance function  $J(\boldsymbol{\rho})$  to be quadratic, was found to be satisfactory in this study. More advanced methods of estimating the gradient from discrete measurements are certainly possible (Yeow et al., 2010; Bunin et al., 2013a), but would require greater computational effort, whereas a least-squares regression is easily carried out within the 60-ms time constraint.

Condition (18) is implemented to reduce the negative effect of uncertainty in  $\nabla \hat{J}(\boldsymbol{\rho}_k)$ . An equivalent version of this condition with  $\delta_J \approx 0$  may be written as follows<sup>1</sup>:

$$\sum_{i=1}^3 \max \left[ - \left( \frac{\partial \hat{J}}{\partial \rho_i} \Big|_{\boldsymbol{\rho}_k} \pm m_k \right) \frac{\partial \hat{J}}{\partial \rho_i} \Big|_{\boldsymbol{\rho}_k} \right] < 0, \quad (26)$$

<sup>1</sup> Each component is a maximum of two elements. See Bunin et al. (2013e).

with the left-hand side easily computed for a given value of  $m_k$ . As mentioned, this condition is satisfied when  $m_k = 0$ , but must be violated at some point as  $m_k$  is increased. The value of  $m_k$  at which (26) becomes equality, denoted by  $\bar{m}$ , is approximated online by bisection. Since using  $m_k := \bar{m}$  would lead to very conservative adaptations, the heuristic choice of  $m_k := 0.5\bar{m}$  is used. The adaptation step is then set as:

$$\bar{K}_k := -2 \frac{\sum_{i=1}^3 \max \left[ - \left( \frac{\partial \hat{J}}{\partial \rho_i} \Big|_{\rho_k} \pm m_k \right) \frac{\partial \hat{J}}{\partial \rho_i} \Big|_{\rho_k} \right]}{\nabla \hat{J}(\rho_k)^T \bar{\mathbf{Q}} \nabla \hat{J}(\rho_k)}, \quad (27)$$

where we choose  $\gamma = 2$  for the largest step possible<sup>2</sup>.

The matrix  $\bar{\mathbf{Q}}$  is approximated from the quadratic regression (and updated at each run  $k$ ) by a diagonal matrix with entries:

$$\bar{Q}_{ii} := 2 \max(2a_{ii}, 10^{-3}), \quad (28)$$

i.e., by projecting the Hessian of the regression in (24) onto the space of positive definite matrices and multiplying by an additional safety factor of 2.

The algorithm is initialized with  $\rho_0 := (1, 2.77, -2.6)$ , which correspond to a stable controller found by simple hand tuning. As no *a priori* data are assumed to be available, runs corresponding to  $k = 1, 2$ , and 3 consist of simply perturbing each parameter, in turn, by  $+0.05$ , so that  $\rho_1 := (1.05, 2.77, -2.6)$ ,  $\rho_2 := (1.05, 2.82, -2.6)$ , and  $\rho_3 := (1.05, 2.82, -2.55)$ , which allows for an estimate of the gradient based on a linear regression (the quadratic terms in (24) are simply ignored until enough measurements to form an overdefined regression become available).  $\bar{\mathbf{Q}}$  is initialized as  $500\mathbf{I}$  (an *ad hoc* choice) until it is possible to carry out the estimation in (28).  $\Delta\rho$  and  $\Delta\bar{\rho}$  are chosen as 0.01 and 0.05, respectively, and  $\bar{K}_k$  is clipped or extended accordingly in the case that (27) leads to adaptations that fail to meet these limits.

#### 4.4 Experimental Results

The RTO algorithm was launched 10 times for each of the four trajectories  $y_{d,1}(t, 2)$ ,  $y_{d,1}(t, 1)$ ,  $y_{d,2}(t, 2)$ ,  $y_{d,2}(t, 1)$  – making for a total of 40 experiments, each of which lasted for 50 runs (10 minutes). For each case, we also asked three members of our lab to tune, by hand, the controller so as to obtain the best possible performance that they could in the same 10-minute window. This was done mainly so as to put the RTO results in some perspective, with all of the results obtained by the tuners repeated for an additional 50 runs so as to filter out the effect of the non-repeatability noise term  $\delta_k$ . It is also worth noting that both the RTO and human testers started with no real *a priori* knowledge of the system.

Figs. 3 and 4 provide results for all four cases, both in terms of  $J_k$  and in terms of the visual improvement due to reduced tracking error. For the former, we provide the results of all of the experiments together with an average of each ten so as to give a general idea of the effect that using the RTO algorithm has on control performance. We

<sup>2</sup> Though needed theoretically, we do not insist on  $\gamma < 2$  with strict inequality in practice.

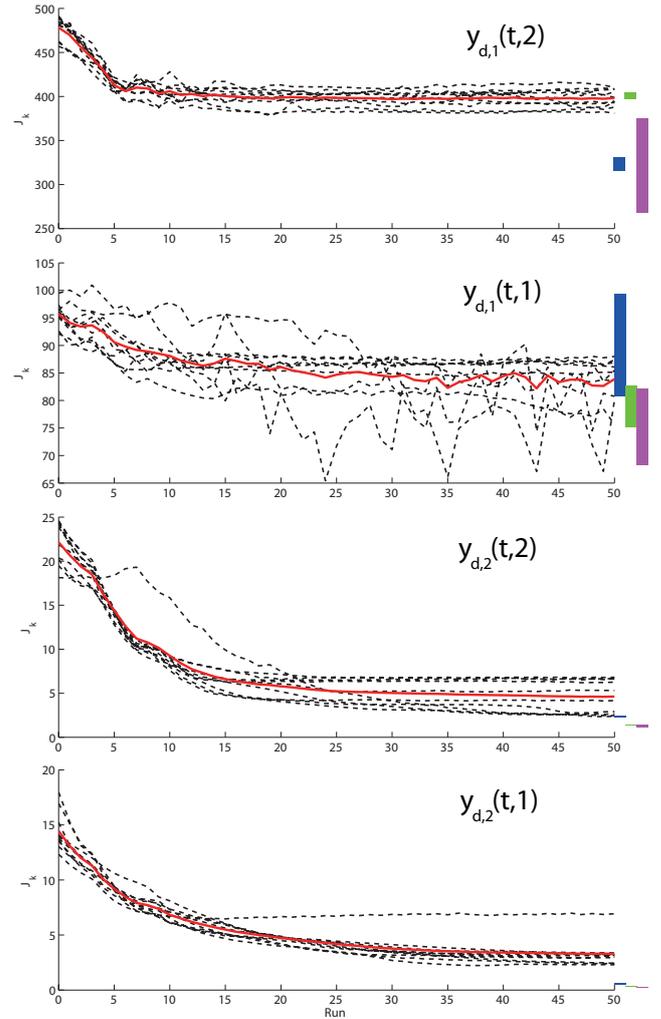


Fig. 3. Improvement in the performance metric due to RTO-based controller tuning. Black lines denote the 10 individual cases, while the red line denotes their average. The vertical bars on the right denote the average performance ( $\pm$  twice the standard deviation) obtained by the three human tuners.

see that, in each of the four cases, the average effect is quite good despite a significant  $\delta_k$  (consider the differences in  $J_0$  for the different experiments as well as the particularly noisy nature of  $y_{d,1}(t, 1)$ , illustrated by the large variations in performance both for the controllers obtained by the RTO and those obtained by the human tuners). We see that, as would be expected in theory, the average tracking error is reduced, almost monotonically, until a stationary point with  $\nabla \hat{J}(\rho_k) \approx \mathbf{0}$  is reached. Furthermore, while we allow the RTO to operate for fifty runs, we see that the majority of the improvement in each case is attained within the first ten, of which three are used to initialize the algorithm due to lack of prior knowledge – with the exception of the  $y_{d,1}(t, 2)$  case, one sees a noticeably faster improvement starting from the fourth run. For the visual results in Fig. 4, we note that the improvement is less obvious for the  $y_{d,1}(t, 2)$  case (though clearly present as indicated by the performance metric in Fig. 3). For the other three cases, it is clear that better tracking is achieved, particularly for the two sinusoidal trajectories.

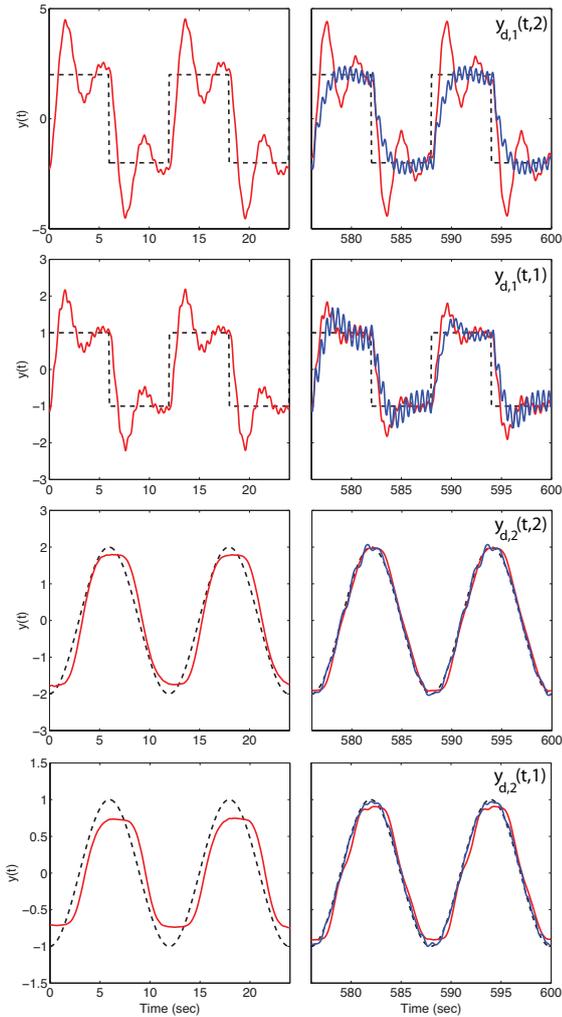


Fig. 4. Visual difference between the tracking during the first two runs and the final two runs, with the dashed black lines denoting the setpoint, the red lines denoting the average output, and the blue denoting the best performance obtained by a human tuner.

We also note that the human tuners almost always do better than the RTO, but this should not be surprising given the low complexity of the controller. It is clear that the innate advantage of RTO lies in its automatic nature, as it may be run continuously without human supervision and can be applied to high-complexity controllers where a good manual approach may not be obvious.

## 5. CONCLUSIONS

This paper has proposed an approach to iterative controller tuning based on real-time optimization (RTO) theory, with the argument that such an approach may enrich the tuning problem by allowing the use of various approaches and algorithms from the RTO domain. A case study on a laboratory torsional system confirmed that even a fairly simple scheme designed to enforce RTO convergence gave, on average, the kind of performance that would be expected when applying a reliable RTO algorithm. It is expected that using more advanced gradient estimation techniques and RTO algorithms would lead to even better performance. We end by noting that a significant extension

of this work, which uses more powerful RTO tools, incorporates constraints, and examines a number of different controllers, is available in Bunin et al. (2013b).

## ACKNOWLEDGEMENT

The authors would like to thank Fernando Fraire Tirado for the initial experimental work that led to this publication, Dr. Alireza Karimi for helpful discussions, and doctoral students Sriniketh Srinivasan, Sean Costello, and Zlatko Emedji for acting as the human tuners.

## REFERENCES

- Box, G. and Draper, N. (1969). *Evolutionary Operation: A Statistical Method for Process Improvement*. John Wiley & Sons.
- Brdys, M. and Tatjewski, P. (2005). *Iterative Algorithms for Multilayer Optimizing Control*. Imperial College Press.
- Bunin, G.A., Fraire, F., François, G., and Bonvin, D. (2012). Run-to-run MPC tuning via gradient descent. *Comput. Aided Chem. Eng.*, 30, 927–931.
- Bunin, G.A., François, G., and Bonvin, D. (2013a). From discrete measurements to bounded gradient estimates: A look at some regularizing structures. *Ind. Eng. Chem. Res.* doi:10.1021/ie303309a.
- Bunin, G.A., François, G., and Bonvin, D. (2013b). A real-time optimization framework for the iterative controller tuning problem. *Processes (submitted)*.
- Bunin, G.A., François, G., and Bonvin, D. (2013c). *The SCFO Real-Time Optimization Solver: Users' Guide (version 0.9.4)*. Ecole Polytechnique Fédérale de Lausanne. <http://infoscience.epfl.ch/record/186672>.
- Bunin, G.A., François, G., and Bonvin, D. (2013d). Sufficient conditions for feasibility and optimality of real-time optimization schemes - I. Theoretical foundations. *arXiv:1308.2620v1 [math.OC]*.
- Bunin, G.A., François, G., and Bonvin, D. (2013e). Sufficient conditions for feasibility and optimality of real-time optimization schemes - II. Implementation issues. *arXiv:1308.2625v1 [math.OC]*.
- Conn, A., Scheinberg, K., and Vicente, L. (2009). *Introduction to Derivative-Free Optimization*. Cambridge University Press.
- Educational Control Products (2008). *Manual for Model 205/205a: Torsional Control System*.
- Hjalmarsson, H., Gevers, M., Gunnarsson, S., and Lequin, O. (1998). Iterative feedback tuning: Theory and applications. *Control Systems, IEEE*, 18(4), 26–41.
- Karimi, A., Mišković, L., and Bonvin, D. (2004). Iterative correlation-based controller tuning. *Int. J. Adapt. Control Signal. Process.*, 18, 645–664.
- Killingsworth, N.J. and Krstić, M. (2006). PID tuning using extremum seeking: Online, model-free performance optimization. *Control Systems, IEEE*, 26(1), 70–79.
- Marchetti, A., Chachuat, B., and Bonvin, D. (2010). A dual modifier-adaptation approach for real-time optimization. *J. Process Control*, 20(9), 1027–1037.
- Myers, R., Montgomery, D., and Anderson-Cook, C. (2009). *Response Surface Methodology*. John Wiley & Sons.
- Yeow, Y., Isac, J., Khalid, F., Leong, Y., and Lubansky, A. (2010). A method for computing the partial derivatives of experimental data. *AIChE J.*, 56(12), 3212–3224.