Published version of this article: https://dx.doi.org/10.1177/0959651811426062

Optimizing the Convergence of Data-Based Controller Tuning

Diego Eckhard and Alexandre Sanfelice Bazanella

***Abstract*—Data-based control design methods most often consist of iterative adjustment of the controller's parameters towards the parameter values which minimize an H_2 performance criterion. Typically, batches of input-output data collected from the system are used to feed directly a gradient descent optimization algorithm - no process model is used. Two topics are important regarding this algorithm: the convergence rate and the convergence to the global minimum. This paper discusses these issues and provides a method for choosing the step size to ensure convergence with high convergence rate, as well as a test to verify at each step whether or not the algorithm is converging to the global minimum.**

***Index Terms*—Control systems, Optimization methods, Gradient methods.**

I. Introduction

Data-based controller tuning methods have drawn considerable attention in the control community in the last fifteen years. Several methods have appeared such as *Controller Unfalsification* [20], *Iterative Feedback Tuning* [10], [8], *Virtual Reference Feedback Tuning* [4], [15], *Correlation based Tuning* [13], [14] and *Frequency Domain Tuning* [12]. Several of these data-based control design methods utilize iterative algorithms to find the minimum of an H_2 performance criteria; this is the case of *Iterative Feedback Tuning*, *Correlation based Tuning* and *Frequency Domain Tuning*. These iterative data-based methods utilize data obtained from one or more experiments in order to compute estimates of the cost function gradient and maybe of its Hessian. These estimates are then used to feed an iterative optimization algorithm, such as steepest descent or Newton-Raphson method. The methods *Virtual Reference Feedback Tuning* and *Noniterative Correlation based Tuning* are "direct" (meaning not iterative) but they are also based on the optimization of the same H_2 performance criteria. Both direct and iterative methods have overlapping, yet different particular application fields, dictated not only by their performance and requirements, but also by operational constraints and the users' preferences and background.

Regarding the experiments used in data-based design methods, they tend to be more complex as more information is required - it requires more data to obtain a good enough estimate of the Hessian than to estimate the gradient, for instance. A major credential for a data-based design is simplicity, both in terms of the information required for the optimization and in the number of algorithm parameters to be commissioned. Then an algorithm of choice is naturally the steepest descent algorithm, which requires estimation of the cost function's gradient only - and not of the cost function's Hessian, as other algorithms do - and whose only design parameter is the step size. So, in this paper we focus on the gradient algorithm but extending the reasoning and the results for other descent methods is in sight.

Although very good results have been reported over the years [10], [12], [13], little is said about the convergence of the algorithms. The convergence of the steepest descent algorithm depends on the tuning of the size of each algorithm step, and this tuning tends to involve expert intervention by a designer on a case-by-case basis, which is highly undesired in the contexts in which data-based methods are intended to be applied. Indeed, what is desirable in adaptive and data-based control design is to have general policies for the step size choice, involving no human intervention whatsoever, so that the method can be applied in a fully automated manner. Moreover, a limiting factor in the application of these iterative data-based design methods is the possible (and quite common) convergence to local (nonglobal) minima of the criterion. It would be highly desirable to verify at each step of the algorithm whether or not it is converging to the global minimum.

This contribution advances in two directions. First, a test is provided for checking at each iteration whether or not the algorithm is converging to the global minimum. Second, whenever this test yields a positive answer, the step size at each iteration can be chosen to speed up convergence. If, on the other hand, this test yields a negative answer, then measures can be taken to enforce convergence to the global minimum [1], but these are outside the scope of this paper. Initial results of this work were presented in [5], however the current results are less conservative. It is assumed in this paper that the data collected from the process is not corrupted by noise; the resulting properties of the iterative data-based control design algorithms are proven under this assumption. We have proven elsewhere [6] that the relevant properties which are obtained when the data are noiseless are robust to the presence of noise, so this issue is not discussed in this paper.

The paper is organized as follows. Section 2 presents the definitions and the problem statement, and in section 3 iterative algorithms are presented. Section 4 presents the main result of this work. Case studies are presented in section 5 and concluding remarks are given in section 6.

D. Eckhard and A. S. Bazanella are with Department of Electrical Engineering, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil

II. Preliminaries

A. System definitions and notation

Consider a linear time-invariant discrete-time single-input single-output process

$$y(t) = G(z)u(t) + \nu(t), \tag{1}$$

where z is the forward time-shift operator, $G(z)$ is the process transfer function, assumed rational and proper, $u(t)$ is the control input and $\nu(t)$ is the process noise. The noise is a quasi-stationary process which can be written as $\nu(t) = H(z)e(t)$, where $e(t)$ is white noise with variance σ_e^2. This process is controlled by a linear time-invariant controller $C(z,\rho)$ such that $C(z,\rho)G(z)$ has positive relative degree for all $\rho \in \mathcal{D}_\rho$. It is further assumed that the controller has a parametric structure as specified below.

Assumption 1: Linear Parametrization

$$C(z,\rho) = \rho^T \bar{C}(z), \tag{2}$$

where $\rho \in \mathcal{D}_\rho \subseteq \mathbb{R}^p$ and $\bar{C}(z)$ is a column vector of fixed rational functions.

Some of the most common controller structures are linearly parametrized, PID with fixed derivative pole being the most popular. Indeed, a PID can be written as

$$C(z,\rho) = \left[\begin{array}{ccc} k_p & k_i & k_d \end{array}\right] \left[\begin{array}{ccc} 1 & \frac{z}{z-1} & \frac{z-1}{z} \end{array}\right]^T.$$

In addition, any transfer function with any dependence on ρ can be approximated by a transfer function of the form (2). Indeed, one can always choose a set of basis functions such that a transfer function - say $F(z)$ - can be represented as

$$F(z) = \sum_{i=1}^{\infty} \alpha_i f_i(z) \tag{3}$$

where $\alpha_i \in \mathbb{R}$ and $f_i(z)$ are the basis transfer functions. One example of a celebrated choice for this basis are the Laguerre functions:

$$f_i(z) = \frac{\sqrt{1-a^2}}{(z-a)} \left[\frac{1-az}{z-a}\right]^i$$

where $a \in (-1,\ 1)$. A truncation of the series in (3) yields

$$F(z) \cong \sum_{i=1}^{p} \alpha_i f_i(z)$$

which can approximate the transfer function to any degree of accuracy desired by choosing sufficiently large p [2]. For these practical and theoretical reasons, Assumption 1 does not represent a relevant loss of generality.

The control action $u(t)$ is given by

$$u(t) = C(z,\rho)(r(t) - y(t)), \tag{4}$$

where $r(t)$ is the reference signal, which is assumed to be quasi-stationary and uncorrelated with the noise, that is

$$\bar{E}\left[r(t)e(s)\right] = 0 \ \forall t, s \tag{5}$$

where $\bar{E}[f(t)] = lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N} E[f(t)]$ and $E[\cdot]$ denotes expectation [17]. The system (1)-(4) in closed-loop becomes

$$y(t,\rho) = T(z,\rho)r(t) + S(z,\rho)\nu(t)$$

$$T(z,\rho) \triangleq \frac{C(z,\rho)G(z)}{1 + C(z,\rho)G(z)} = C(z,\rho)G(z)S(z,\rho)$$

where we have now made the dependence on the controller parameter ρ explicit in the output signal $y(t,\rho)$. Let us define the set Γ of all control parameters values that render the closed-loop system BIBO-stable, that is, $\Gamma \triangleq \{\rho : T(z,\rho) \text{ is BIBO-stable}\}$.

B. The Model Reference Control Problem

We want the closed-loop to achieve a given performance which is specified by a "desired" closed-loop transfer function

$$y_d(t) = T_d(z)r(t),$$

called *reference model* in the literature. We thus search for the controller parameters that make the output of the system the closest to the desired one, by solving the following optimization problem.

$$\min_\rho J(\rho)$$

$$J(\rho) \triangleq \bar{E}\left[(y(t,\rho) - y_d(t))^2\right]. \tag{6}$$

Under the hypothesis (5), the cost function (6) can be divided into two terms [1]:

$$J(\rho) = J_y(\rho) + J_e(\rho) \tag{7}$$

where each term represents a conceptually different control objective:

- reference tracking:

$$J_y(\rho) = \bar{E}\left[((T_d(z) - T(z,\rho))r(t))^2\right]; \tag{8}$$

- noise rejection:

$$J_e(\rho) = \bar{E}\left[(S(z)\nu(t))^2\right]. \tag{9}$$

The method *Frequency Domain Tuning* (FDT) [12] minimizes the performance criterion $J_e(\rho)$. It assumes that there is no reference and the gradient of the criterion is obtained by a frequency domain approach. On the other hand, the method *Correlation based Tuning* (CbT) [13] optimizes the performance criterion $J_y(\rho)$. The method *Iterative Feedback Tuning* (IFT) uses data from two closed-loop experiments to obtain an estimate to the gradient of $J(\rho)$, so both reference tracking and noise rejection are taken into account.

In this work only the reference tracking problem (J_y) is considered; the development for noise rejection is very similar. Once the desired closed-loop transfer function $T_d(z)$ is chosen, it could be exactly achieved with the ideal controller

$$C_d(z) = \frac{T_d(z)}{G(z)(1 - T_d(z))}. \tag{10}$$

This controller is the one that minimizes the tracking error criterion J_y, causing this performance criterion to be evaluated at zero. Note however, that this controller can be unstable and/or non-causal if the reference model $T_d(z)$ is not chosen properly. To avoid these problems the relative degree of the reference model $T_d(z)$ can not be smaller than the relative degree of the process $G(z)$, and all the nonminimum phase zeros of the process must be present in the reference model. These are constraints that must be taken into account at an earlier stage of the design, when an appropriate reference model must be chosen. Of course, such a choice requires some minimum knowledge about the process, but certainly not a model.

If and only if the ideal controller $C_d(z)$ lies within the class of controllers considered the closed-loop system can behave exactly as specified by the reference model. We will assume in this work that it is the case. Let us formalize this assumption, which is referred to as the Matched Control Case:

Assumption 2:

$$\exists \rho_d \in \mathcal{D}_\rho : C(z,\rho_d) = C_d(z) = \rho_d^T \bar{C}(z).$$

This hypothesis is equivalent to the standard assumption in system identification that the process being identified belongs to a given model class [17] [7] and as such implies similar properties and similar difficulties. In order to have Assumption 2 satisfied, it is necessary to chose the reference model such that the ideal controller is causal, which requires knowledge of an upper bound for the relative degree of the process' transfer function. On the other hand, for nonminimum phase processes Assumption 2 requires the reference model to possess the nonminimum phase zeroes of the process in order to avoid unstable pole-zero cancelations, which in principle would require an earlier identification stage to identify these zeroes. Alternatives to this previous identification have been proposed for IFT and for VRFT in [16] and [3] respectively.

Under Assumptions 1 and 2, a direct method such as VRFT or CbT could be used to tune the controller's parameters, and avoiding iterations is in principle advantageous. Still, iterative adjustments of the controller's parameters of an operating controller may be preferred over an abrupt change of the parameters in many practical situations, mainly for operational reasons, even under these Assumptions. Moreover, the direct methods rely on Assumption 2 to yield the correct answer, which is the global minimum of the performance criterion J_y [4], [13], whereas the iterative methods only rely on these hypothesis to optimize convergence.

We know from [1] that if Assumption 2 is ensured then the gradient can be written as

$$\nabla J_y(\rho) = \frac{\partial J_y(\rho)}{\partial \rho} = M(\rho)(\rho - \rho_d) \tag{11a}$$

$$M(\rho) = M_s(\rho) + M_a(\rho) \tag{11b}$$

$$\begin{aligned} M_s(\rho) &= \frac{1}{\pi}\int_{-\pi}^{\pi} \Phi_r(e^{j\omega}) \left|G(e^{j\omega})S(e^{j\omega},\rho)\right|^2 \times \quad (11c) \\ &\quad \Re\left\{S_d^*(e^{j\omega})S(e^{j\omega},\rho)\right\} \Re\left\{\bar{C}(e^{j\omega})\bar{C}^*(e^{j\omega})\right\} d\omega \\ M_a(\rho) &= -\frac{1}{\pi}\int_{-\pi}^{\pi} \Phi_r(e^{j\omega}) \left|G(e^{j\omega})S(e^{j\omega},\rho)\right|^2 \times \quad (11d) \\ &\quad \Im\left\{S_d^*(e^{j\omega})S(e^{j\omega},\rho)\right\} \Im\left\{\bar{C}(e^{j\omega})\bar{C}^*(e^{j\omega})\right\} d\omega \end{aligned}$$

The matrix $M_s(\rho)$ is symmetric, whereas $M_a(\rho)$ is anti-symmetric, $\Re\{\cdot\}$ denotes the real part and $\Im\{\cdot\}$ the imaginary part of a complex number and the desired sensitivity function is defined as $S_d(z) = 1 - T_d(z)$.

III. Iterative Algorithms

In adaptive control and data-based control design a model for the process is not known *a priori*, so neither is the cost function $J(\rho)$. Only local information about the cost function can be obtained from data collected on the system, so iterative gradient-based methods are used to find a minimum of $J(\rho)$. The algorithms have the following structure

$$\rho_{i+1} = \rho_i - \gamma_i R_i^{-1} \nabla J(\rho_i) \tag{12}$$

where $R_i \in \mathbb{R}^{p\times p}$ and $\gamma_i > 0\ \forall i$. When R_i is the identity matrix for all i the algorithm is called steepest descent and the updates are made in the opposite direction of the gradient, so, at least for sufficiently small γ_i, at each iteration a smaller value for the cost is achieved. If R_i is an estimate of the Hessian of the criterion and $\gamma_i = 1$ then the algorithm is called Newton-Raphson method.

Several papers on data-based control give the following classical result [19] as a theoretical foundation for the analysis of the convergence of algorithms like (12).

Theorem 1: [10], [9] Consider a twice-differentiable function $J(\rho) : \mathbb{R}^n \to \mathbb{R}^+$. If the optimization problem is unconstrained, the estimate of the gradient is uniformly bounded and the step size γ_i of the steepest descent algorithm respects the following conditions:

$$\sum_{i=1}^{\infty} \gamma_i = \infty \qquad \sum_{i=1}^{\infty} \gamma_i^2 < \infty$$

then

$$\lim_{i\to\infty} \rho_i = \{\rho : \nabla J(\rho) = 0\}.$$

A "classical" choice of the step sizes, which satisfies these conditions, is the harmonic series [10], [11]:

$$\gamma_i = \frac{\gamma_1}{i};\ i > 1. \tag{13}$$

This step sequence choice is theoretically justified, and as such is often presented as a benchmark, although it tends to provide poor convergence rates and it leaves open the issue of how to chose the first step size γ_1. Moreover, this result only ensures that the algorithm will converge to a stationary point of the cost function - we cannot ensure that this will be the global minimum.

A major concern in optimization, and particularly in the H_2 control design formulation, is the convergence to the global minimum of the performance criterion. This can rarely be achieved globally, that is, regardless of the initialization of the algorithm, so we define the following.

Definition 1: A set $\Omega \subset \mathbb{R}^n$ is a domain of attraction (DOA) of an algorithm $\rho_{i+1} = f(\rho_i)$ for the function $J_y(\rho)$ (8) if $\lim_{i\to\infty} \rho_i = \rho_d, \forall \rho_0 \in \Omega$.

The algorithm can only converge to the global minimum if the parameters are inside the domain of attraction. Therefore, to ensure convergence, it is necessary to check if the parameters are inside this set. It is also necessary to choose the step size sequence properly, in order to have a good convergence rate. In the next section we will present the main results of this work: a test to verify if the algorithm is converging to the global minimum and a step size policy to ensure a high convergence rate.

IV. Main Results

In this section, relevant properties of the steepest descent algorithm are explored to design a step size sequence with high convergence rate and the test to verify if the algorithm is converging to the global minimum. The next theorem presents the main result of this work.

Theorem 2: [1] Consider a twice-differentiable function $J_y(\cdot) : \mathbb{R}^n \to \mathbb{R}^+$. Assume that this function has an isolated global minimum ρ_d, and that Assumptions 1 and 2 are satisfied. Define

$$\begin{aligned} \gamma_i^{max} &= \max_{\gamma_i} \gamma_i \text{ subject to} \\ &\quad \gamma_i > 0 \\ &\quad \gamma_i M^T(\rho_i) M(\rho_i) - 2M_s(\rho_i) < 0 \end{aligned} \tag{14}$$

where $M(\rho_i)$ and $M_s(\rho_i)$ are defined in (11). If at each iteration the step size of the steepest descent algorithm satisfies

$$\gamma_i < \gamma_i^{max} \tag{15}$$

then the algorithm (12) with $R_i = I$ converges to the global minimum ρ_d.

Proof: Let $V(\rho) = (\rho - \rho_d)^T(\rho - \rho_d)$ be a Lyapunov function for the discrete-time system (12) with $R_i = I$. Then the set

$$\mathcal{B}_\alpha(\rho_d) = \left\{ \rho : (\rho - \rho_d)^T(\rho - \rho_d) < \alpha \right\}.$$

is a domain of attraction if the Lyapunov function $V(\rho)$ is strictly decreasing in time, that is, if

$$V(\rho_{i+1}) - V(\rho_i) < 0;\ \forall \rho_i \in \mathcal{B}_\alpha(\rho_d), \rho_i \neq \rho_d \tag{16}$$

Replacing (12) into (16) with $R_i = I$ we have

$$\begin{aligned} &V(\rho_{i+1}) - V(\rho_i) = \\ &(\rho_i - \gamma_i \nabla J_y(\rho_i) - \rho_d)^T (\rho_i - \gamma_i \nabla J_y(\rho_i) - \rho_d) \\ &-(\rho_i - \rho_d)^T(\rho_i - \rho_d) = -2\gamma_i(\rho_i - \rho_d)^T \nabla J_y(\rho_i) \\ &+\gamma_i^2 \nabla J_y^T(\rho_i) \nabla J_y(\rho_i) < 0 \end{aligned}$$

which is ensured if

$$0 < \gamma_i < 2\frac{(\rho_i - \rho_d)^T \nabla J_y(\rho_i)}{\nabla J_y^T(\rho_i) \nabla J_y(\rho_i)}. \tag{17}$$

Replacing (11a) in (17) we get:

$$\gamma_i < 2\frac{(\rho_i - \rho_d)^T M(\rho_i)(\rho_i - \rho_d)}{(\rho_i - \rho_d)^T M^T(\rho_i) M(\rho_i)(\rho_i - \rho_d)}$$

Now, $M_a(\rho)$ is an anti-symmetric matrix, so the above expression can be simplified to

$$\gamma_i < 2\frac{(\rho_i - \rho_d)^T M_s(\rho_i)(\rho_i - \rho_d)}{(\rho_i - \rho_d)^T M^T(\rho_i) M(\rho_i)(\rho_i - \rho_d)}$$

which is equivalent to

$$(\rho_i - \rho_d)^T(\gamma_i M^T(\rho_i) M(\rho_i) - 2M_s(\rho_i))(\rho_i - \rho_d) < 0$$

which is quadratic and is satisfied by the conditions of the theorem.

Recall that any connected and bounded level set of a Lyapunov function is a domain of attraction if the Lyapunov difference is strictly negative in its interior. Then the proof is completed by noting that $\mathcal{B}_\alpha(\rho_d)$ is a connected and bounded level set of the Lyapunov function $V(\rho)$. ∎

The theorem ensures that the steepest descent algorithm converges to the global minimum if the step size respects the conditions of the theorem. If the step size is smaller than γ_i^{max}, then at each iteration the parameters of the controller are closer to the global minimum ρ_d, because $V(\rho_{i+1}) < V(\rho_i)$. Any value of the step size that respects $0 < \gamma_i < \gamma_i^{max}$ can be used to ensure convergence, however some values make the convergence rate higher. Someone could think that the larger step size possible should be used, but this is not always the best option. As we can see in Figure 1, sometimes the larger step size puts the next parameter ρ_{i+1} further from the global minimum ρ_d than smaller step sizes. This can happen because the rule (15) only ensures that the next iteration will be in a ball (centered in ρ_d) with smaller radius. From the authors experience, most of the times the use $\gamma_i = \gamma_i^{max}/2$ generates very high convergence rates, but any value between $\gamma_i^{max}/2$ and γ_i^{max} can be used as well. It would be necessary to know the parameter ρ_d to compute exactly the step size that generates the higher convergence rate, however we assume that it is unknown. On the other hand, it is necessary to have an estimate of $M(\rho)$ to compute γ_i^{max}, an aspect that will be discussed later.

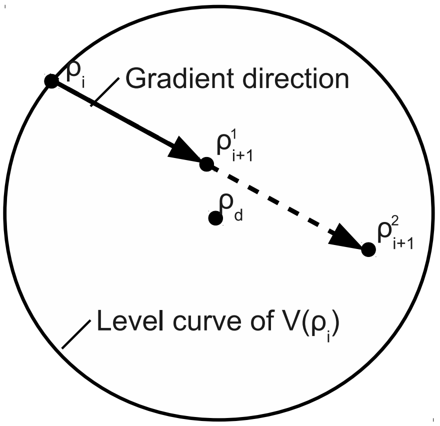

Fig. 1. Choice of the step size: ρ_{i+1}^1 obtained with $\gamma_i = \gamma_i^{max}/2$ and ρ_{i+1}^2 obtained with $\gamma_i = \gamma_i^{max}$.

A. The convergence test

Every time the step size is computed using (15), this calculation indicates whether or not the algorithm is converging to the global minimum. Indeed, if the problem (14) has a feasible solution, then we know that the algorithm is pointing to the right a direction, such that the parameters of the next iteration are going to be closer to the global minimum than the actual parameters. It can also happen that the problem (14) doesn't have solution, which means that the algorithm will be pointing to the wrong direction in the parameter space - that is, away from the global minimum. So, the procedure used for the calculation of the step size also serves as a test to verify if the algorithm is converging to the global minimum.

The problem (14) doesn't have solution if the parameters of the algorithm are outside the candidate domain of attraction. In this case the matrix $M_s(\rho)$ is not positive definite and the steepest descent algorithm can not converge to the global minimum. Every time that the optimization problem is infeasible some action must be taken. Whenever such situation occurs, one can resort to the *Cost Function Shaping* methodology [1]. The *Cost Function Shaping* makes use of the variables that the designer has at his disposal (mainly the reference spectrum and the reference model) to shape the cost function such that it becomes well-behaved as desired. It has been shown, among other things, that it is always possible to manipulate these variables such that any given (stabilizing) controller be inside the candidate domain of attraction. However, this is not in the scope of this article. The combination of proposed step sizes and cost function shaping makes it possible to converge to the global minimum from any stabilizing) controller.

B. *Estimate of* $M(\rho)$

The computation of γ_i^{max} requires an estimate of $M(\rho)$, a matrix which is given by:

$$\begin{aligned} M(\rho) &= \frac{1}{\pi}\int_{-\pi}^{\pi} \Phi_r(e^{j\omega})\left|G(e^{j\omega})S(e^{j\omega},\rho)\right|^2 \times \\ &\Re\left\{S_d^*(e^{j\omega})S(e^{j\omega},\rho)\bar{C}(e^{j\omega})\bar{C}^*(e^{j\omega})\right\}d\omega \end{aligned} \quad (18)$$

It is observed that this matrix depends on the reference signal, on the controller, on the reference model and on the frequency response of the process. The reference signal, the controller and the reference model are data known by the designer. The only information unknown concerns the frequency response of the process. If a precise model of the process were available, it would be possible to compute the controller using (10), and the data-based methods would be useless.

On the other hand, if nothing is known about the process, then it is impossible to choose appropriately the reference model and complexity of the controller. At least some very basic information about the process must be available in order to set up a data-based control design, and in most of the cases this basic information about the process is indeed available a priori, such as approximations of the dominant time constant, the static gain and the delay. These informations can be used to obtain a rough model of the process, which can serve the purpose of estimating the matrix $M(\rho)$ using equation (18).

The model does not need to be precise to the method ensure convergence because it is not used to compute the gradient direction, which is done based solely on input-output the data, but only to determine the step sizes. If an error on the model makes the step size smaller than it should be, then the convergence is still achieved, but the convergence rate will be lower than expected. On the other hand, if an error on the model makes the step size larger, then it could in principle happen that the algorithm will leave the domain of attraction of the global minimum, but the step size used is half the one that would cause this problem, but there is quite a large margin for error here.

V. Case studies

In this section we present simulationand experimental results that illustrate the effectiveness of the step size policy proposed.

A. *Simulation result - first example*

The following case study illustrates the large improvements that can be obtained with the step size sequence proposed when compared to a benchmark, as well as the dangers of using the Newton-Raphson algorithm, which is potentially much faster.

Consider the following system

$$y(t) = \frac{0.05}{z-0.95}u(t). \quad (19)$$

This system is controlled by a PI controller

$$C(z,\rho) = \left[\begin{array}{cc} k_p & k_i \end{array}\right]\left[\begin{array}{cc} 1 & \frac{z}{z-1} \end{array}\right]^T. \quad (20)$$

The following reference model is specified:

$$y_d(t) = \frac{0.1}{z-0.9}r(t). \quad (21)$$

From (10) it follows that

$$C_d(z) = \left[\begin{array}{cc} 1.9 & .1 \end{array}\right]\left[\begin{array}{cc} 1 & \frac{z}{z-1} \end{array}\right]^T. \quad (22)$$

This controller lies in the class of PI controllers in the form (20), so Assumption 2 is satisfied.

We consider that the system is initially in closed-loop with the controller

$$C(z,\rho) = \left[\begin{array}{cc} 1 & 2 \end{array}\right]\left[\begin{array}{cc} 1 & \frac{z}{z-1} \end{array}\right]^T. \quad (23)$$

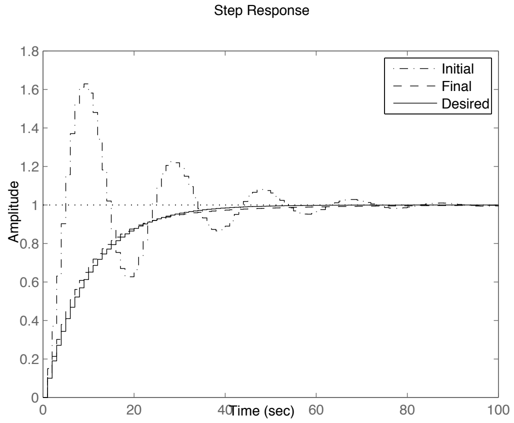

Fig. 2. Closed-loop step response: desired response, achieved response with the initial controller, and with the final controller.

Figure 2 shows the step response of the closed-loop system. Note that the initial response is very different from the desired one. We want now to improve the closed-loop performance utilizing the algorithm (12). We then use the algorithm (12) where $\nabla J(\rho)$ is substituted by the estimate based on data obtained from the Iterative Feedback Tuning (IFT) method. The following reference signal, which is conceived to guarantee persistence of excitation, is used to obtain the data:

$$r(t) = square\left(\frac{2\pi t}{200}\right).$$

where $square\left(\frac{2\pi t}{T}\right)$ stands for a square wave with period T. The period of the reference is larger than the settling time of the process, so the reference can be viewed as a sequence of step signals. 1000 samples of the above signal were used to compute the estimates.

Figure 3 shows the evolution of the controller parameters for 15 iterations using the proposed sequence of step sizes and the "classical" choice of the step sizes (13). The first step size of the classical step sizes sequence was chosen to be the same as the one in the proposed approach.

Table I shows the cost $J(\rho_i)$ at each iteration. By using the "classical" steps, after 15 iterations the cost is still 0.22753, whereas with the proposed step size sequence at iteration 15 the cost is 0.00080, indicating that the closed-loop response is very close to the desired response. Figure 2 also shows the step response with the final controller (iteration 15) of the proposed step size sequence.

After iteration 15 the cost has already achieved a very small value, so the convergence rate becomes also small, which suggests the application of the Newton-Raphson iterations from this point on. The application of this scheme results in the convergence presented in Figure 4 and Table II. We can see that in this configuration the algorithm achieves the global minimum within a few iterations, which strongly suggest this combination of Newton-Raphson and steepest descent with our proposed step size sequence followed by Newton-Raphson iterations.

From these results one may be tempted to apply the Newton-Raphson algorithm from the very beginning, discarding the steepest descent all along; it turns out that this is not a good idea because the Newton-Raphson method typically presents a smaller *domain of attraction* than the steepest descent [18], which implies that the Newton-Raphson method requires a better initialization (that is, closer to the minimum) in order to perform adequately. Also, the Newton-Raphson is more costly, because it requires estimating the Hessian and the gradient of the cost function, whereas the steepest descent algorithm requires only the gradient. Figure 5 and Table III show the result of applying the N-R method to the initial condition (23). In this case, because the initial condition is far from the global minimum the Newton-Raphson method diverges until arriving at an unstable closed-loop at iteration 4.

TABLE II
CONTROLLER PARAMETERS EVOLUTION - NEWTON-RAPHSON METHOD INITIALIZED WITH STEP 15 OF TABLE I.

i	$J(\rho_i)$	$[\,k_p$	$k_i\,]$
1	0.00080687	[2.1750	0.0999]
2	$3.8667e-06$	[1.8828	0.1002]
3	$6.0009e-11$	[1.8999	0.1000]
4	0.0000	[1.9000	0.1000]

Another possibility is to modify the Newton-Raphson method to allow a variable step size. Several articles suggest

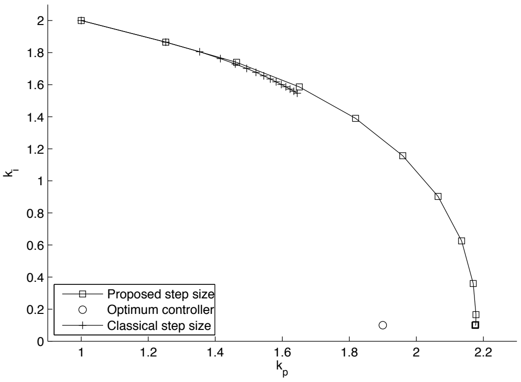

Fig. 3. Controller parameters: evolution with the proposed and classical step size, both starting in $[1\ \ 2]^T$.

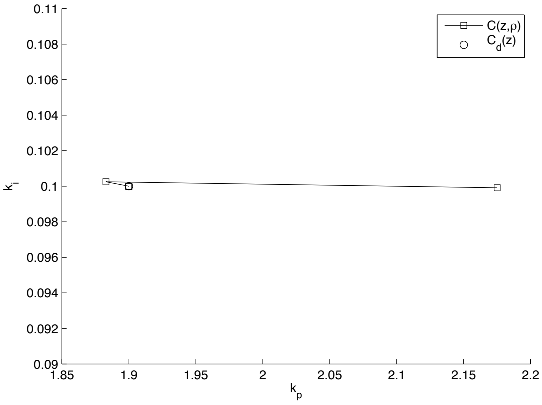

Fig. 4. Controller parameters: evolution with the Newton-Raphson method.

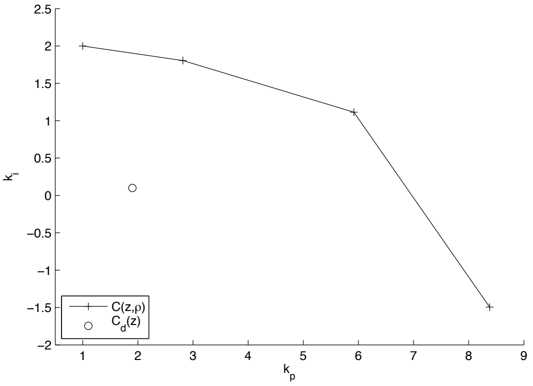

Fig. 5. Controller parameters: evolution with the Newton-Raphson method.

TABLE III
CONTROLLER PARAMETERS EVOLUTION - NEWTON-RAPHSON METHOD.

i	$J(\rho)$	$[\,k_p$	$k_i\,]$
1	0.28958	[1.0000	2.0000]
2	0.20295	[2.8159	1.8053]
3	0.14209	[5.9171	1.1141]
4	∞	[8.3799	-1.4947]

TABLE I
CONTROLLER PARAMETERS EVOLUTION - STEEPEST DESCENT.

	Proposed Step Size			Classical Step Size		
i	$J(\rho_i)$	$[\,k_p$	$k_i\,]$	$J(\rho_i)$	$[\,k_p$	$k_i\,]$
1	0.28958	[1.0000	2.0000]	0.28958	[1.0000	2.0000]
2	0.26532	[1.2523	1.8654]	0.26532	[1.2523	1.8654]
3	0.24700	[1.4635	1.7396]	0.25639	[1.3532	1.8053]
4	0.22948	[1.6509	1.5865]	0.25082	[1.4153	1.7629]
5	0.21028	[1.8186	1.3899]	0.24674	[1.4596	1.7293]
6	0.18819	[1.9596	1.1568]	0.24350	[1.4938	1.7011]
7	0.16067	[2.0650	0.9030]	0.24080	[1.5215	1.6767]
8	0.12162	[2.1353	0.6254]	0.23847	[1.5446	1.6551]
9	0.06691	[2.1697	0.3600]	0.23643	[1.5644	1.6357]
10	0.01251	[2.1779	0.1657]	0.23460	[1.5817	1.6179]
11	0.00087	[2.1774	0.1034]	0.23294	[1.5970	1.6016]
12	0.00081	[2.1768	0.0999]	0.23143	[1.6107	1.5865]
13	0.00081	[2.1762	0.0999]	0.23003	[1.6231	1.5725]
14	0.00081	[2.1756	0.0999]	0.22874	[1.6344	1.5593]
15	0.00080	[2.1750	0.0999]	0.22753	[1.6447	1.5469]

the use of the classical step size (13) rule with the Newton-Raphson search direction given by the cost function's Hessian. Table IV shows the result of applying this variable steps size Newton-Raphson algorithm to the present example. The corresponding reduction of the step size makes the algorithm more conservative, which tends to make convergence slower, yet safer. The algorithm does not diverge, as it was seen before, but after 30 iterations the parameters are still farther away from the global minimum than what is achieved at iteration 15 with the step size proposed in this article, and the corresponding value of the cost function is also significantly larger.

TABLE IV
CONTROLLER PARAMETERS EVOLUTION - NEWTON-RAPHSON METHOD WITH CLASSICAL STEP SIZE

i	$J(\rho)$	$[\,k_p$	$k_i\,]$
1	0.2896	[1.0000	2.0000]
5	0.0819	[5.9890	0.3976]
10	0.0140	[3.2846	0.6603]
15	0.0058	[2.6905	0.7259]
20	0.0031	[2.4530	0.1078]
25	0.0020	[2.3253	0.1060]
30	0.0013	[2.2455	0.1049]

B. Simulation result - second example

This subsection presents an example in which Assumptions 1 and 2 are mildly violated, comparing the performance of direct design and iterative design in this case. In this example the output of the system is corrupted by white noise and the structure of the controller does not permit a perfect match of the desired closed loop.

Consider the flexible transmission system presented in [14] as a benchmark for digital control design. The plant's transfer function is described by

$$G(z) = \frac{0.28261z + 0.50666}{z^4 - 1.41833z^3 + 1.58939z^2 - 1.31608z + 0.88642}. \tag{24}$$

This system is controlled by a linearly parametrized controller

$$C(z,\rho) = [\rho_0\ \rho_1\ \rho_2\ \rho_3\ \rho_4\ \rho_5] \left[\frac{z^5}{z^5 - z^4}\ \frac{z^4}{z^5 - z^4}\ \frac{z^3}{z^5 - z^4}\right.$$
$$\left.\frac{z^2}{z^5 - z^4}\ \frac{z}{z^5 - z^4}\ \frac{1}{z^5 - z^4}\right]^T \tag{25}$$

The following reference model is specified:

$$T_d(t) = \frac{(1 - 0.6065)^2}{z(z - 0.6065)^2} \tag{26}$$

This reference model cannot be achieved with the proposed controller structure; the controller is not flexible enough to make the output of the closed loop system identically to the desired output, even if there were no noise corrupting the data.

In [14], two controllers designed to this problem have been presented. The first one was designed with the method VRFT and its parameters are given by:

$$\rho_{VRFT} = [0.2383\ \ -0.3387\ 0.3261\ \ -0.2547$$
$$0.1991\ \ -0.02382]^T \tag{27}$$

The second one was designed with the method nonitereative correlation based tuning:

$$\rho_{nCbT} = [0.2315\ \ -0.3286\ 0.3192\ \ -0.2610$$
$$0.2107\ \ -0.03225]^T \tag{28}$$

We have used the parameters of the controller obtained with the VRFT ρ_{VRFT} as the initial condition for the method Iterative Feedback Tuning in order to verify if the closed loop performance could be improved using the proposed step size policy.

The reference used was 511 samples of the PRBS signal, and the noise level was chosen such that the signal-to-noise ration was of order 10.

In order to compute the step sizes it is necessary to have an estimate model of the system. The data from one closed loop experiment was used to compute the following estimate to the process using the least square estimate (LSE):

$$\hat{G}(z) = \frac{0.1017z^3 + 0.2562z^2 + 0.2908z + 0.2250}{1.0000z^4 - 0.6418z^3 + 0.1544z^2 - 0.0093z + 0.2951} \tag{29}$$

After 10 iterations the performance of the closed loop was improved and the cost was reduced from 0.0243 to 0.0170, as it can be seen in Table V.

TABLE V
CONTROLLER PARAMETERS EVOLUTION - STEEPEST DESCENT INITIALIZED WITH VRFT CONTROLLER.

i	$J(\rho_i)$	ρ_i^T	γ_i
1	0.0243	[0.2383 -0.3387 0.3261 -0.2547 0.1991 -0.0238]	1.3042
2	0.0179	[0.3126 -0.5439 0.6092 -0.5505 0.4231 -0.1143]	1.5062
3	0.0169	[0.3325 -0.5995 0.7087 -0.6344 0.4561 -0.1104]	1.4732
4	0.0178	[0.3229 -0.5863 0.6972 -0.6321 0.4558 -0.1149]	1.4969
5	0.0180	[0.3251 -0.5906 0.6955 -0.6414 0.4774 -0.1250]	1.4760
6	0.0162	[0.3300 -0.6016 0.7189 -0.6552 0.4742 -0.1191]	1.4847
7	0.0171	[0.3332 -0.6017 0.7110 -0.6461 0.4710 -0.1178]	1.4851
8	0.0171	[0.3270 -0.5932 0.6995 -0.6414 0.4740 -0.1263]	1.5226
9	0.0185	[0.3137 -0.5647 0.6627 -0.5991 0.4349 -0.1057]	1.4419
10	0.0170	[0.3138 -0.5649 0.6682 -0.6040 0.4385 -0.1093]	1.4681

Table VI show the cost $J(\rho)$ obtained with each one of the methods and also present norm of the difference between the desired and achieved closed loop.

TABLE VI
COMPARISON BETWEEN METHODS.

	$J(\rho)$	$\|T_d(z) - T(z,\rho)\|$
IFT	0.0170	0.0736
VRFT	0.0243	0.1077
nCbT	0.0233	0.1083

We can see in this example that proposed step size policy can be used even when the system output is corrupted with noise, and when the controller is not flexible enough to satisfy the Assumption 2. Although the theoretical development in this paper does not cover this case, and relies on Assumptions 1 and 2, the method has shown to be robust against mild violations of these hypothesis. The direct methods - nCbT and VRFT - also provide reasonable results in this case, but the iterative design can further improve the performance with respect to the best performance achievable with a direct design.

It is also worth mentioning that the model $\hat{G}(z)$ used to compute the step sizes is by no means a precise one. If we compute a model-based controller using (10) with the model $\hat{G}(z)$, we obtain a controller that makes the closed loop unstable.

C. Experimental results

This section shows experiments that verify the theoretical results presented previously. The experimental objective is to control the temperature of a process. A resistance is used to heat the system and a thermocouple sensor is used to measure the temperature. We would like to control this process with zero steady-state error for constant references, which is a typical industrial specification.

The quality of the controller can be measured utilizing the following estimate of the cost function

$$\hat{J}(\rho) = \frac{1}{N}\sum_{t=1}^{N}(y(t,\rho) - y_d(t))^2$$

where N is the number of samples utilized to compute the estimate. We chose to utilize the following quality criterion, because it has a physical meaning, namely the average error per sample, in units of the process output (degree Celsius):

$$\bar{J}(\rho) = \sqrt{\hat{J}(\rho)}.$$

TABLE VII
PARAMETERS EVOLUTION.

i	$[k_p \;\; k_i \;\; k_d]$	$J(\rho)$
1	[0.60000 0.10000 0.90000]	3.99
2	[0.68351 0.02539 0.87074]	0.96
3	[0.69349 0.01549 0.86740]	0.74
4	[0.69021 0.01883 0.86848]	0.63

The reference model chosen is

$$T_d(z) = \frac{0.1}{z - 0.9}.$$

The closed-loop system is running initially with the controller:

$$C(z,\rho_1) = [0.6 \;\; 0.1 \;\; 0.9] \begin{bmatrix} 1 & \frac{z}{z-1} & \frac{z-1}{z} \end{bmatrix}^T.$$

The closed-loop system with this controller has an unsatisfactory behavior, as can be seen by the resulting high cost value $\bar{J}(\rho_1) = 3.99\ ^oC$. This means that in average, the response is $3.99\ ^oC$ different from what is desired at each sample, which represents almost 20% of the amplitude of the reference step. The system response is shown in Figure 6.

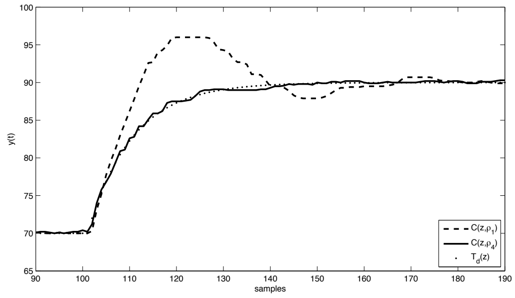

Fig. 6. Reference model and step response with $C(z,\rho_1)$ and $C(z,\rho_4)$.

Let us now apply Iterative Feedback Tuning and the step size sequence computed as previously presented. The reference signal utilized to run the experiments to collect the input-output data was

$$r(t) = 80 + 10square\left(\frac{2\pi t}{64}\right).$$

Again this reference was conceived to guarantee persistence of excitation and its period is larger than the settling time of the process, so the reference can be viewed as a sequence of step signals.

With the controller $C(z,\rho_1)$, the cost value was $\check{J}(\rho_1) = 3.99\ ^oC$. After 4 iterations, the cost value was reduced to $\check{J}(\rho_4) = 0.63\ ^oC$, 6 times smaller than the initial cost value. The evolution of the parameters is shown in the Table VII.

The controller achieved at the $4th$ iteration is given by

$$C(z,\rho_4) = [0.69021 \; 0.01883 \; 0.86848] \begin{bmatrix} 1 & \frac{z}{z-1} & \frac{z-1}{z} \end{bmatrix}^T.$$

The step closed-loop response with the controller $C(z, \rho_4)$ is also shown in the Figure 6.

But how good is this performance in absolute terms, that is, how close to the global optimum have we arrived? We have tested several controller parameters obtained with many different methods. The best controller that was ever achieved is

$$C(z, \rho_d) = [0.675\ 0.020\ 0.793] \left[\begin{array}{ccc} 1 & \frac{z}{z-1} & \frac{z-1}{z} \end{array} \right]^T .$$

The cost of the closed-loop system with the controller $C(z, \rho_d)$, to a step reference signal of 20 ^oC, is measured as $\check{J}(\rho_d) = 0.22\ ^oC$.

VI. Conclusion

Previous publications tend to resort to ad hoc solutions to choose the step sizes in the steepest descent algorithm when tuning the parameters of a controller through data-based methods. We have given a systematic procedure to compute these step sizes. Assuming that the initial condition is inside the candidate domain of attraction of the global minimum of the performance criterion, this step size policy ensures that the algorithm converges to this global minimum. Also, at each iteration the calculation informs us whether or not the algorithm is converging to the global optimum. In the examples given, which include experimental results, convergence speed is radically improved when compared to a classical systematic choice of the step size that is often given as a benchmark. The examples also illustrate the role of the Newton-Raphson algorithm, which we recommend to use in combination with the steepest descent, but not by itself, and that the proposed method can still be applied if some assumptions of the method are slightly violated. Indeed, even if there is noise corrupting the data and/or the Assumption 2 is not satisfied the method still performs as appropriately; however, the implication of these violations is a topic deserving further research effort. The practical experiments also indicate that the method can be easily applied in a practical way.

VII. Acknowledgment

This work was supported in part by Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq - Brazil.

References

[1] A. S. Bazanella, M. Gevers, L. Miskovic, and B. D. O. Anderson. Iterative minimization of H_2 control performance criteria. *Automatica*, 44(10):2549–2559, 2008.

[2] Ricardo J. G. B. Campello, Gerard Favier, and Wagner C. do Amaral. Optimal expansions of discrete-time volterra models using laguerre functions. *Automatica*, 40(5):815 – 822, 2004.

[3] L. Campestrini, Diego Eckhard, Michel Gevers, and Alexandre S. Bazanella. Virtual reference feedback tuning for non-minimum phase plants. *Automatica*, 47(8):1778–1784, 2011.

[4] M. C. Campi, A. Lecchini, and S. M. Savaresi. Virtual reference feedback tuning: a direct method for the design of feedback controllers. *Automatica*, 38(8):1337–1346, 2002.

[5] D. Eckhard and A. S. Bazanella. Optimizing the convergence of data-based controller tuning. In *Proc. European Control Conference*, pages 910–915, Budapest, Hungary, 2009. Budapest: EUCA.

[6] D. Eckhard and A.S. Bazanella. Robust convergence of the steepest descent method for data-based control. *International Journal of Systems Science*, 99999(1):1–7, 2011.

[7] D. Eckhard, L. Campestrini, M. E. Bergel, and A. S. Bazanella. Data-based control design for a process class with guaranteed convergence to the globally optimum controller. In *Proc. European Control Conference*, Budapest, Hungary, 1999. Budapest: EUCA.

[8] M. Gevers. A decade of progress in iterative control design: from theory to practice. *Journal of process control*, 12(4):519–531, 2002.

[9] H. Hjalmarsson. Iterative feedback tuning - an overview. *International journal of adaptive control and signal processing*, 16(5):373–395, 2002.

[10] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin. Iterative feedback tuning: theory and applications. *IEEE control systems magazine*, 18(4):26–41, 1998.

[11] J. K. Huusom, N. K. Poulsen, and S. B. Jorgensen. Improving convergence of iterative feedback tuning. *Journal of process control*, 19(4):570–578, 2009.

[12] L. C. Kammer, R. R. Bitmead, and P. L. Bartlett. Direct iterative tuning via spectral anaysis. *Automatica*, 36(9):1301–1307, 2000.

[13] A. Karimi, L. Miskovic, and D. Bonvin. Iterative correlation-based controller tuning. *International journal of adaptive control and signal processing*, 18(8):645–664, 2004.

[14] A. Karimi, K. van Heusden, and D. Bonvin. Noniterative data-driven controller tuning using the correlation approach. In *European Control Conference 2007*, 2007.

[15] A. Lecchini, M. C. Campi, and S. M. Savaresi. Virtual reference feedback tuning for two degrees of freedom controllers. *International journal of adaptive control and signal processing*, 16:355–371, 2002.

[16] A. Lecchini and M. Gevers. On iterative feedback tuning for non-minimum phase plants. In *41st IEEE Conference on Decision and Control*, pages 4658–4663, 2002.

[17] L. Ljung. *System identification: theory for the user*. Prentice Hall, Englewood Cliffs, NJ, EUA, 1987.

[18] A. Ralston and P. Rabinowitz. *A First Course in Numerical Analysis*. Dover Publications, Mineola, USA, 2001.

[19] H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, 22(3):400–407, 1951.

[20] M. G. Safonov and Tung-Ching Tsao. The unfalsified control concept and learning. *IEEE Transactions on Automatic Control*, 42(6):843–847, June 1997.