Iterative Feedback Tuning for cascade systems

Douglas Tesch[1], Diego Eckhard[2] and Alexandre S. Bazanella[3]

***Abstract*— Iterative Feedback Tuning (IFT) is a data-driven method used to tune parameters of feedback controllers minimising an H_2 criterion. The method uses data from experiments to estimate the gradient of the criterion, and uses iterative quasi-newton algorithms to adjust the controllers. When the method is used in cascade systems, usually the inner loop is firstly adjusted, and after the outer loop. In this article we describe an extension to the IFT method that adjusts both inner and outer loop at the same time using only data from closed-loop experiments at each iteration.**

I. INTRODUCTION

Iterative Feedback Tuning (IFT) has been first proposed roughly two decades ago [1] and is now a well established design methodology [2], [3]. IFT has pioneered the Data-Driven approach (DD, also called Data-Based) for control design, which has become an important paradigm with sound design methodologies [4]–[8]. The most general formulation of IFT is valid for the so-called two-degree-of-freedom controller structure - a controller in the loop plus a set-point filter - and requires, at each iteration, the realisation of three experiments to guarantee statistical convergence to the optimal controller.

Cascade control is a very common control configuration (see, for instance, [9]) that is not explicitly encompassed by the IFT formulation. Cascade controllers consist of an outer (main) control loop, and an inner (secondary) control loop. The secondary loop controls an intermediate variable, with the output of the main controller as its set-point. IFT could, in principle, be applied to tune cascade controllers by adjusting each controller separately, but there are at least two major inconveniences in doing so. First, this would demand twice as many experiments at each iteration, since there are two controllers to be tuned independently. Second, it would require the choice of a reference model for the secondary loop. In conventional practice the secondary loop is designed to be as fast as possible, in an attempt to make its dynamics negligible with respect to the dynamics of the main loop. This idea could be applied to conceive a reference model for the secondary loop. However, as we will show by means of an example, this does not usually yield the best result in terms of the performance of the main loop - which is the sole objective of the cascade control.

In this paper we develop an extension of the IFT formulation that applies to the cascade control structure, allowing the simultaneous tuning of both controllers. In this new formulation, no extra experiments are required and there is no need to define a reference model for the inner loop. The paper is organised as follows. Section II presents the classical Iterative Feedback Tuning for non-cascade systems. In Section III the cascade feedback system is described. The proposed methodology to apply Iterative Feedback Tuning for cascade systems is presented in Section IV. Numerical examples are given in Section V and final conclusions in Section VI.

II. ITERATIVE FEEDBACK TUNING

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

$$y(t) = G(q)u(t) + v(t) = G(q)u(t) + H(q)e(t), \qquad (1)$$

where q is the forward-shift operator, $G(q)$ is the process transfer function, $u(t)$ is the control input, $H(q)$ is the noise model, and $e(t)$ is zero mean white noise with variance σ_e^2. Both transfer functions, $G(q)$ and $H(q)$, are rational and causal.

The system is in feedback control where control action $u(t)$ can be written as

$$u(t) = C(q, \rho)(r(t) - y(t)), \qquad (2)$$

where $C(q, \rho)$ is the transfer function of the controller which is parameterized by $\rho \in \mathbb{R}^n$.

The system (1)-(2) in closed loop becomes

$$y(t, \rho) = T(q)r(t) + S(q)v(t) \qquad (3)$$

$$T(q) = \frac{C(q)G(q)}{1 + C(q)G(q)} = C(q)G(q)S(q) \qquad (4)$$

where we have made explicit the dependence of the output on the parameter vector ρ.

We also assume that the model of the process is unavailable to the user such that the transfers functions $G(q)$ and $H(q)$ are unknown. However, we assume that the user can collect a batch of data from the process

$$Z^N = [y(1), \dots, y(N)].$$

The Iterative Feedback Tuning is an iterative method which solves the following H_2 optimisation problem:

$$\min_{\rho} \quad J(\rho)$$

where

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

*This work was supported by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and CNPq/Brasil.

[1]Douglas Tesch is with Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil. E-mail: douglasatesch@gmail.com.

[2]Diego Eckhard is with Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil. E-mail: diegoeck@ufrgs.br.

[3]Alexandre S. Bazanella is with Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil. E-mail: bazanella@ufrgs.br.

The optimisation criterion $J(\rho)$ is the mean square error between the output of the closed loop system $y(t,\rho)$ and the *desired output* $y_d(t)$. The desired output is calculated using a *reference model* $T_d(q)$:

$$y_d(t) = T_d(q)r(t)$$

where the closed-loop performance is specified by the choice of the transfer function $T_d(q)$. Usually the DC gain of $T_d(q)$ is one, such that the desired closed-loop system presents zero steady-state error.

The optimisation problem can be solved iteratively by *Quasi-Newton Algorithms*

$$\rho^{i+1} = \rho^i - \gamma_i R_i \nabla J(\rho^i) \tag{5}$$

where $\rho^i \in \mathbb{R}^n$ is the controller parameters at iteration i, $\nabla J(\rho^i)$ is the gradient of $J(\rho)$ with respect to the parameter vector ρ, $R_i \in \mathbb{R}^{n\times n}$ is a matrix which defines (together with $\nabla J(\rho^i)$) the direction of the updates and $\gamma_i \in \mathbb{R}$ is used to tune the step size. When R_i is the identity matrix the algorithm is called *Steepest Descent* and the updates are done in the opposite direction of the gradient of the cost function. The convergence of this algorithm to a local minimum of the cost function $J(\rho)$ depends on two factors according to ([1]–[4], [10]): the use of an unbiased estimate to the gradient of the cost function and a correct choice for the step size γ_i. In this article we will present an unbiased estimate to the gradient and we recommend the above articles to a deep understanding of the step size choice.

The method described until here can be classified as an optimum controller design with H_2 criterion. The optimisation can be performed with or without knowledge of the system models; the objective of Iterative Feedback Tuning method is optimize the criterion using only input/output data as information from the process.

So, the method is an algorithm to obtain both the gradient of the cost function $\nabla J(\rho^i)$ and the matrix R_i from closed-loop data from the system. Observe that the gradient can be calculated as

$$\nabla J(\rho^i) = \frac{2}{N}\sum_{t=1}^{N}\left(y_d(t) - y(t,\rho)\right)\frac{\partial y}{\partial \rho}$$

where

$$\frac{\partial y}{\partial \rho} = \frac{T(q,\rho)}{C(q,\rho)}\frac{\partial C(q,\rho)}{\partial \rho}\left[r(t) - T(q,\rho)r(t)\ - S(q,\rho)v(t)\right].$$

Observe that to compute the gradient, the signal $\frac{\partial y}{\partial \rho}$ is needed, which depends on the model process what is unknown. So the main idea of IFT method is use collected data from two specific closed loop experiments (in the case of one degree of freedom controller) to obtain $\frac{\partial y}{\partial \rho}$ and estimate the gradient of the cost function. The two experiments are given by:

- First experiment: $r^1(t) = r(t)$ and the output signal $y^1(t)$ is collected; the superscript means that the data is collected in the first experiment.
- Second experiment: $r^2(t) = r(t) - y^1(t)$ and the output of this experiment $y^2(t)$ is collected.

The method suggest the following estimate for the gradient of the output with respect to the parameter vector:

$$\widehat{\frac{\partial y}{\partial \rho}} = \frac{1}{C(q,\rho)}\frac{\partial C}{\partial \rho}y^2(t)$$

such that the gradient of the criterion is estimated as

$$\widehat{\nabla J(\rho^i)} = \frac{2}{N}\sum_{t=1}^{N}\left(y_d(t) - y^1(t,\rho)\right)\widehat{\frac{\partial y}{\partial \rho}}.$$

The reference signal $r(t)$ is assumed to be quasi-stationary and uncorrelated with the noise, that is $\bar{E}\left[r(t)e(s)\right] = 0\ \forall t, s$, and

$$\bar{E}[f(t)] \triangleq \lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N}E[f(t)]$$

with $E[\cdot]$ denoting expectation [11]. Then it is possible to show that

$$E\left[\widehat{\nabla J(\rho^i)}\right] = \nabla J(\rho^i)$$

such that the estimate is unbiased.

A common choice for the the matrix R_i is

$$\widehat{R_i} = \left(\frac{2}{N}\sum_{t=1}^{N}\widehat{\frac{\partial y}{\partial \rho}}^T\widehat{\frac{\partial y}{\partial \rho}}\right)^{-1} \tag{6}$$

which is a biased approximation to the inverse of Hessian of the criterion $J(\rho)$.

Using the estimates $\widehat{\nabla J(\rho^i)}$ and $\widehat{R_i}$ the user can run the *Quasi-Newton Algorithm* (5) and then optimize the criterion $J(\rho)$. For each iteration of the algorithm, the user should run two closed loop experiments to collect the data $y^1(t)$ and $y^2(t)$. Observe that all the procedure is performed without the use of the process model.

The method Iterative Feedback Tuning can also be used to tune two degree of freedom controllers (2DOF) as described in theIFT. In this case, the estimate of the gradient of the cost function is obtained using data from three specific experiments. The first two experiments are exactly the same as described above, but the method also needs data from a third experiment with the same reference signal as the first experiment. In order to obtain unbiased estimates to the Hessian of the cost function, one should run a fourth specific experiment, as described in [12].

III. Cascade Systems

Let us assume a linear time-invariant discrete-time single-input-single-output cascade process

$$\begin{aligned} y_1(t) &= G_1(q)u_1(t) + v_1(t) \\ &= G_1(q)u_1(t) + H_1(q)e_1(t), \end{aligned} \tag{7}$$

$$\begin{aligned} y_2(t) &= G_2(q)u_2(t) + v_2(t) \\ &= G_2(q)u_2(t) + H_2(q)e_2(t), \end{aligned} \tag{8}$$

where $G_1(q)$ and $G_2(q)$ are the inner and outer process transfer function respectively, $H_1(q)$ and $H_2(q)$ are the noise models, $e_1(t)$ and $e_2(t)$ are zero mean white noise with variances σ_{e1}^2 and σ_{e2}^2. All transfer functions are rational and causal.

The cascade feedback control system is shown in Figure 1. The control action can be written as

$$u_1(t) = C_1(q, \rho_1)(r_1(t) - y_1(t)), \tag{9}$$
$$r_1(t) = C_2(q, \rho_2)(r_2(t) - y_2(t)), \tag{10}$$

where $C_1(q, \rho_1)$ and $C_2(q, \rho_2)$ are respectively the transfer functions of the inner and outer controllers, which are parameterized by $\rho_1 \in \mathbb{R}^{n1}$ and $\rho_2 \in \mathbb{R}^{n2}$. The signals $r_1(t)$ and $r_2(t)$ are the references of the *inner* and *outer* system, $r_2(t)$ is assumed to be quasi-stationary and uncorrelated with the noises, that is $\bar{E}\left[r_2(t)e_1(s)\right] = 0$ and $\bar{E}\left[r_2(t)e_2(s)\right] = 0\ \forall t, s$.

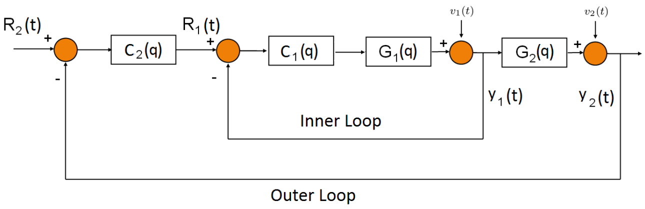

Fig. 1. Cascade system.

The system in closed loop becomes

$$\begin{aligned} y_2(t, \rho_1, \rho_1) &= T_2(q, \rho_1, \rho_2)r_2(t) + S_2(q, \rho_1, \rho_2)v_2(t) \\ &\quad + S_1(q, \rho_1, \rho_2)v_1(t) \end{aligned} \tag{11}$$
$$\begin{aligned} y_1(t, \rho_1, \rho_2) &= T_1(q, \rho_1, \rho_2)(r_2(t) - v_2(t)) \\ &\quad + \frac{S_1(q, \rho_1, \rho_2)}{G_2(q)} v_1(t) \end{aligned} \tag{12}$$

where

$$\begin{aligned} T_i(q, \rho_1) &= \frac{y_1}{r_1} = \frac{C_1(q, \rho_1)G_1(q)}{1 + C_1(q, \rho_1)G_1(q)} \\ T_2(q, \rho_1, \rho_2) &= \frac{y_2}{r_2} = \frac{C_2(q, \rho_2)T_i(q, \rho_1)G_2(q)}{1 + C_2(q, \rho_2)T_i(q, \rho_1)G_2(q)} \\ T_1(q, \rho_1, \rho_2) &= \frac{y_1}{r_2} = \frac{T_2(q, \rho_1, \rho_2)}{G_2(q)} \end{aligned}$$

and

$$\begin{aligned} S_2(q, \rho_1, \rho_2) &= \frac{y_2}{v_2} = 1 - T_2(q, \rho_1, \rho_2) \\ S_1(q, \rho_1, \rho_2) &= G_2(q)S_2(q, \rho_1, \rho_2)(1 - T_i(q, \rho_1)). \end{aligned}$$

IFT can be directly used to tune cascade systems, by tuning each loop separately. The inner loop is firstly adjusted tuning ρ_1, using two experiments for each iteration, until the algorithm converges to a good inner performance. The outer loop is then adjusted tuning ρ_2, using also data from two experiments at each iteration. So, a typical cascade adjust uses twice as much experiments than a non-cascade adjust. In this article we describes an extension of the Iterative Feedback Tuning method that adjust both controllers (inner and outer) at the same time.

IV. Cascade Iterative Feedback Tuning

The proposed *Cascade Iterative Feedback Tuning* is an iterative method which solves the following optimisation problem

$$\min_{\rho_1, \rho_2} J(\rho_1, \rho_2) \tag{13}$$

where the objective function $J(\rho_1, \rho_2)$ depends on both controller parameters ρ_1 and ρ_2 and is defined as

$$J(\rho_1, \rho_2) = \frac{1}{N} \sum_{t=1}^{N} (y_2(t, \rho_1, \rho_2) - y_d(t))^2. \tag{14}$$

The *desired output* $y_d(t)$ is defined as

$$y_d(t) = T_d(q)r_2(t)$$

where $T_d(q)$ is a desired transfer function from the *outer loop* reference to the output.

The output signal $y_2(t, \rho_1, \rho_2)$ depends on the parameters ρ_1 and ρ_2. To simplify the notation, we will consider that

$$\rho = \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} \quad \nabla J(\rho) = \begin{bmatrix} \frac{\partial J}{\partial \rho_1} \\ \frac{\partial J}{\partial \rho_2} \end{bmatrix}. \tag{15}$$

The gradients can be written as

$$\frac{\partial J}{\partial \rho_1} = \frac{2}{N} \sum_{t=1}^{N} (y_2(t, \rho) - y_d(t)) \frac{\partial y_2}{\partial \rho_1} \tag{16}$$

$$\frac{\partial J}{\partial \rho_2} = \frac{2}{N} \sum_{t=1}^{N} (y_2(t, \rho) - y_d(t)) \frac{\partial y_2}{\partial \rho_2} \tag{17}$$

which depend on the partial derivative of $y_2(t)$ with relation to ρ_1 and ρ_2.

In the following equations we will drop dependence on q and ρ to save space. Observe that

$$\frac{\partial y_2}{\partial \rho_1} = \frac{\partial}{\partial \rho_1} \left(T_2 r_2(t) + S_2 v_2(t) + S_1 v_1(t)\right)$$

so that, after computing the derivatives, it follows that

$$\begin{aligned} \frac{\partial y_2}{\partial \rho_1} = \frac{\nabla C_1}{C_1} \{ & T_2(r_2(t) - y_2(t)) \\ & - \frac{T_1}{C_2} (y_2(t) - v_2(t)) \}. \end{aligned} \tag{18}$$

Also, the gradient with relation to ρ_2 is given by

$$\frac{\partial y_2}{\partial \rho_2} = \frac{\partial}{\partial \rho_2} \left(T_2 r_2(t) + S_2 v_2(t) + S_1 v_1(t)\right)$$

such that

$$\frac{\partial y_2}{\partial \rho_2} = \frac{\nabla C_2}{C_2} \{T_2(r_2(t) - y_2(t))\}. \tag{19}$$

In order to compute the gradient (15) one need to compute (16)-(17) using equations (18) and (19). However, the signals $y_2(t)$, $\frac{\partial y_2}{\partial \rho_1}$ and $\frac{\partial y_2}{\partial \rho_1}$ depend on the model of the process which is unknown. So, in the sequence we will establish a method to estimate the gradient of the criterion using only input/output data collected from closed-loop experiments.

The estimates will be computed using data from three specific experiments, the superscript means that the data is collected at that experiment:

- First Experiment:
  The reference signal is $r_2(t) = r^1(t)$ and we collect $y_1^1(t)$ and $y_2^1(t)$. Observe that

$$\begin{aligned} y_1^1(t) &= T_1 r^1(t) + \frac{S_1}{G_2} v_1^1(t) - T_1 v_2^1(t) \\ y_2^1(t) &= T_2 r^1(t) + S_1 v_1^1(t) + S_2 v_2^1(t). \end{aligned}$$

- Second experiment:
  The reference signal is $(r^1(t) - y_2^1(t))$ (the error of the collected signals in the first experiment) and we again collect $y_1^2(t)$ and $y_2^2(t)$. Observe that

$$\begin{aligned} y_1^2(t) &= T_1(r^1 - y_2^1) + \frac{S_1}{G_2} v_1^2(t) - T_1 v_2^2(t) \\ y_2^2(t) &= T_2(r^1 - y_2^1) + S_1 v_1^2(t) + S_2 v_2^2(t). \end{aligned}$$

- Third experiment:
  The reference signal is again $r^1(t)$ and we again collect $y_1^3(t)$ and $y_2^3(t)$. Observe that

$$\begin{aligned} y_1^3(t) &= T_1 r^1(t) + \frac{S_1}{G_2} v_1^3(t) - T_1 v_2^3(t) \\ y_2^3(t) &= T_2 r^1(t) + S_1 v_1^3(t) + S_2 v_2^3(t). \end{aligned}$$

Three signals $y_1^1(t)$, $y_1^2(t)$ and $y_2^2(t)$ are then used to estimate the gradient, using the following formulas:

$$\widehat{\frac{\partial y_2}{\partial \rho_1}} = \frac{\nabla C_1}{C_1}\left[y_2^2(t) - \frac{1}{C_2}(y_1^1(t) - y_1^2(t))\right] \quad (20)$$

$$\widehat{\frac{\partial y_2}{\partial \rho_2}} = \frac{\nabla C_2}{C_2}\left[y_2^2(t)\right]. \quad (21)$$

These estimates are computed using only input/output data from the process and they are exactly the correct gradients when there is no noise, as we can observe in the following equations.

$$\begin{aligned} \widehat{\frac{\partial y_2}{\partial \rho_1}} = \frac{\partial y_2}{\partial \rho_1} + \frac{\nabla C_1}{C_1}\left\{S_2 v_2^2(t) + S_1 v_1^2(t)\right\} \\ - \frac{\nabla C_1}{C_1}\frac{1}{C_2}\left\{\frac{S_1}{G_2}(v_1^1 - v_1^2) + T_1 v_2^2\right\} \end{aligned}$$

$$\widehat{\frac{\partial y_2}{\partial \rho_2}} = \frac{\partial y_2}{\partial \rho_2} + \frac{\nabla C_2}{C_2}\left\{S_2 v_2^2(t) + S_1 v_1^2(t)\right\}$$

Notice that the estimate $\widehat{\frac{\partial y_2}{\partial \rho_1}}$ is corrupted by noise that comes from the two experiments $v_1^1(t)$, $v_2^1(t)$, $v_1^2(t)$ and $v_2^2(t)$. However, the estimate $\widehat{\frac{\partial y_2}{\partial \rho_2}}$ is corrupted by noise only from the second experiment $v_1^2(t)$ and $v_2^2(t)$.

The data from the third experiment is only used to compute the gradient of the criterion, using the formula

$$\widehat{\nabla J(\rho^i)} = \frac{2}{N}\sum_{t=1}^{N}\left(y_d(t) - y_2^3(t,\rho)\right)\widehat{\frac{\partial y(t)}{\partial \rho}}.$$

Observe that the term $\left(y_d(t) - y_2^3(t,\rho)\right)$ contains noise only from the third experiment, while $\widehat{\frac{\partial y(t)}{\partial \rho}}$ contains noise from the first and second experiments. Since the noise from an experiment is uncorrelated with the noise of another experiment, we can show that

$$E\left[\widehat{\nabla J(\rho^i)}\right] = \nabla J(\rho^i).$$

The matrix R_i is computed as

$$R_i = \left(\frac{2}{N}\sum_{t=1}^{N}\widehat{\frac{\partial y_2}{\partial \rho}}^T \widehat{\frac{\partial y_2}{\partial \rho}}\right)^{-1}$$

V. Numerical Example

The aim of this section is to illustrate the Cascade Iterative Feedback Tuning where this method is compared to the conventional method.

The process is defined as

$$G_1(q) = \frac{1}{q - 0.8} \quad G_2(q) = \frac{1}{q - 0.9}$$

and there is noise in both systems loops conform (12) and (11), being $v_1(t)$ and $v_2(t)$ Gaussian noises with zero mean and variance of 10^{-4}.

The controllers $C_1(q,\rho_1)$ and $C_2(q,\rho_2)$ are both PIs (Proportional-Integral). This kind of cascade PI controllers widely used in practical situations involving cascade control [13] [14]. Each PI controller is parameterized as

$$C(q,\rho) = \rho^T \overline{C}(q) \quad (22)$$

where ρ and $\overline{C}(q)$ are defined as

$$\rho = \begin{bmatrix} k_p \\ k_i \end{bmatrix} \quad \overline{C}(q) = \begin{bmatrix} 1 \\ \frac{q}{q-1} \end{bmatrix}. \quad (23)$$

The reference model was chosen as

$$T_d(q) = \frac{0.016q^2 - 0.0246q + 0.00923}{q^4 - 3.5q^3 + 4.626q^2 - 2.738q + 0.6122}$$

such that it can be exactly achieved by the controllers

$$C_{d1}(q) = \begin{bmatrix} 0.2 \\ 0.07 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{1}{q-1} \end{bmatrix} \quad (24)$$

$$C_{d2}(q) = \begin{bmatrix} 0.08 \\ 0.009 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{1}{q-1} \end{bmatrix}. \quad (25)$$

in a noise free environment. Since the level of the noise affecting the data is low, we expect that the controllers related to global minimum of $J(\rho)$ be close to $C_{d1}(q)$ and $C_{d2}(q)$.

The process is initially controlled by the PI controllers given by

$$C_1(q) = \begin{bmatrix} 0.1 \\ 0.05 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{1}{q-1} \end{bmatrix} \quad (26)$$

$$C_2(q) = \begin{bmatrix} 0.05 \\ 0.004 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{1}{q-1} \end{bmatrix}. \quad (27)$$

Figure 2 shows the output of the outer loop $y_2(t)$ obtained with the initial controllers (26)-(27) and a square wave as reference ($r_2(t)$), with unitary amplitude and 2 seconds period as

$$r_2(t) = square\left(\frac{2\pi t}{2}\right). \tag{28}$$

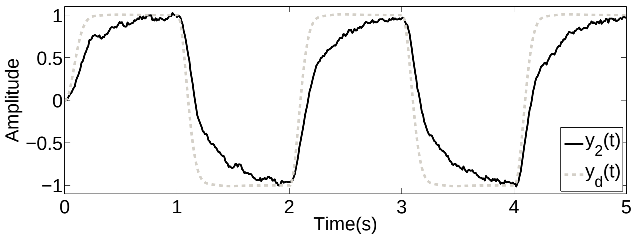

Fig. 2. System response $y_2(t)$ with controllers (26)-(27) (continuous line); desired response $y_d(t)$ (dashed line).

A. Quasi-Newton Algorithm

The *Steepest Descent Algorithm* (matrix $R_i(t)$ as identity) usually has a large region of attraction, but it tends to provide low convergence rates. On the other hand, the Quasi-Newton algorithm with matrix $R_i(t)$ as (6) usually has fast convergence rate and small region of attraction [15]. Therefore, we suggest the use of *Steepest Descent* in the first iterations and then the *Quasi-Newton algorithm*, in order to boost the convergence rate without losing the large domain of attraction. The used step size sequence for *Steepest Descent* and *Quasi-Newton algorithm* were chosen as

$$Steepest\ Descent : \gamma_i = \frac{\mu}{\|\nabla J\|} \tag{29}$$

$$Quasi\text{-}Newton : \gamma_i = \frac{i}{L} \tag{30}$$

where μ is a constant with 0.001 value and L is the maximum iterations number.

B. Conventional Methodology

We used the conventional tuning to cascade controllers, where the tuning is carried out in stages. In the first stage the inner loop is adjusted tuning $C_1(q,\rho_1)$. This stage is very difficult for the user, because there is no closed loop specification for the inner loop response. In the second stage, the outer loop is adjusted tuning the $C_2(q,\rho_2)$.

The IFT algorithm is executed in the inner loop with reference signal defined as (28). The $T_d(t)$ is chosen as an unitary transfer function, in other words the secondary loop is design to be as fast as possible. Table I show the controller parameters convergence of $C_1(q,\rho_1)$. At the end of controller tuning, we obtain as a result the PI controller given by

$$C_1(q,\rho_1) = \begin{bmatrix} 0.9950 \\ 0.1992 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{1}{q-1} \end{bmatrix} \tag{31}$$

Table II show the controller parameters convergence of $C_2(q,\rho_2)$. At the end of the outer loop tuning, we obtain as a result the PI controller given by

$$C_2(q,\rho_2) = \begin{bmatrix} 0.0578 \\ 0.0099 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{1}{q-1} \end{bmatrix} \tag{32}$$

TABLE I

INNER LOOP IFT RESULTS.

int	$C_1(q)$ k_p	k_d	R_i
1	0.1000	0.0500	Identity
2	0.1006	0.0508	Identity
3	0.1012	0.0516	Identity
4	0.1018	0.0524	Identity
5	0.1024	0.0532	Identity
6	0.1030	0.0540	Hessian
7	0.1054	0.0541	Hessian
8	0.1104	0.0545	Hessian
9	0.1180	0.0550	Hessian
10	0.1281	0.0557	Hessian
20	0.4507	0.0826	Hessian
30	0.9282	0.1694	Hessian
40	0.9950	0.1992	Hessian

TABLE II

OUTER LOOP IFT RESULTS.

int	J	$C_2(q)$ k_p	k_d	R_i
1	0.1063	0.0500	0.0040	Identity
2	0.0629	0.0500	0.0050	Identity
3	0.0356	0.0500	0.0060	Identity
4	0.0189	0.0501	0.0070	Identity
5	0.0098	0.0501	0.0080	Identity
6	0.0052	0.0501	0.0090	Hessian
7	0.0053	0.0502	0.0090	Hessian
8	0.0051	0.0504	0.0090	Hessian
9	0.0051	0.0507	0.0090	Hessian
10	0.0050	0.0511	0.0091	Hessian
20	0.0038	0.0561	0.0096	Hessian
30	0.0036	0.0579	0.0098	Hessian
40	0.0036	0.0578	0.0099	Hessian

C. Proposed Methodology

We tune both inner and outer loop controllers at same time using the proposed *Cascade Iterative Feedback Tuning*. The controller parameters convergence of $C_1(q,\rho_1)$ and $C_2(q,\rho_2)$, with their cost can be shown in Table III.

After 40 iterations the Cascade IFT achieves the controllers

$$C_1(q,\rho_1) = \begin{bmatrix} 0.1986 \\ 0.0691 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{1}{q-1} \end{bmatrix}$$

$$C_2(q,\rho_2) = \begin{bmatrix} 0.0797 \\ 0.0090 \end{bmatrix}^T \begin{bmatrix} 1 \\ \frac{1}{q-1} \end{bmatrix}.$$

Comparing Tables II and III we can observe that the proposed approach achieved a much better performance that can be measured by $J(\rho)$. The proposed Cascade Iterative Feedback Tuning cost is 9 times smaller than the conventional method that tune the inner and outer loops separately. Figure 3 shows the convergence of the parameters which were initialised with controller parameters $\rho_1 = \begin{bmatrix} 0.1 & 0.05 \end{bmatrix}$ and $\rho_2 = \begin{bmatrix} 0.05 & 0.004 \end{bmatrix}^T$. After 40 iterations the algorithm is very close to the minimum of the criterion, but after 20 iterations the method had already obtained a very good performance and could be stopped there, without the need of the final iterations. After 8 iterations the proposed method already found a smaller cost than the conventional method.

TABLE III

CASCADE IFT RESULTS.

int	J	$C_1(q)$		$C_2(q)$		R_i
		k_p	k_d	k_p	k_d	
1	0.1071	0.1000	0.0500	0.0500	0.0040	Identity
2	0.0651	0.1000	0.0500	0.0500	0.0050	Identity
3	0.0401	0.1000	0.0500	0.0501	0.0060	Identity
4	0.0284	0.1000	0.0501	0.0502	0.0070	Identity
5	0.0246	0.1000	0.0502	0.0503	0.0080	Identity
6	0.0240	0.1000	0.0509	0.0511	0.0078	Hessian
7	0.0042	0.1046	0.0533	0.0777	0.0081	Hessian
8	0.0023	0.1123	0.0574	0.0826	0.0084	Hessian
9	0.0016	0.1206	0.0657	0.0773	0.0087	Hessian
10	0.0013	0.1257	0.0698	0.0766	0.0087	Hessian
20	0.0004	0.1912	0.0714	0.0789	0.0090	Hessian
30	0.0004	0.1971	0.0693	0.0792	0.0090	Hessian
40	0.0004	0.1986	0.0691	0.0797	0.0090	Hessian

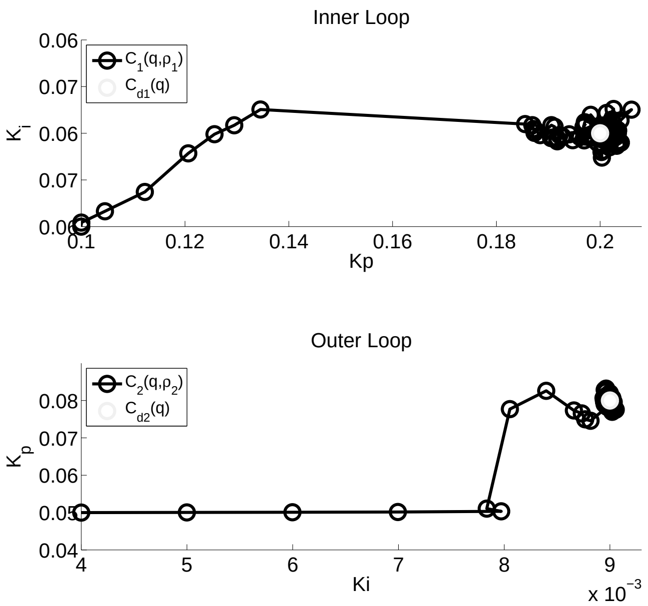

Fig. 3. Parameter convergence of controllers $C_1(q, \rho_1)$ and $C_2(q, \rho_2)$ with *Cascade IFT algorithm*.

The closed loop response $y_2(t)$ for the two methodologies are shown in Figure 4.

VI. CONCLUSION

We have presented an extension of IFT for the tuning of cascaded controllers. The resulting procedure requires three experiments at each iteration and is based only on the performance specified for the outer loop. It is thus very advantageous when compared to the separate application of IFT to each one of the two controllers. Indeed, separate application of IFT to each controller in the cascade would require more experiments per iteration (two experiments for each controller) and additional work for the designer (the choice of an extra reference model for the inner loop). Moreover, and perhaps most importantly, our cascade IFT converges to the optimal performance of the outer loop, whereas in the standard IFT this would require the designer

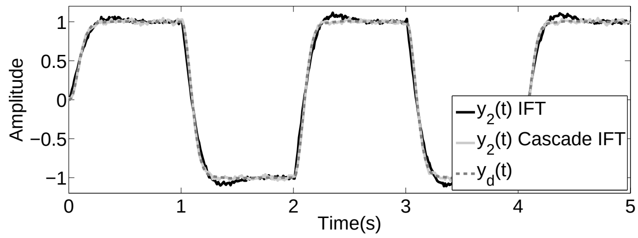

Fig. 4. Output $y_2(t)$ obtained with differents IFTs methodologies.

to chose the best reference model for the inner loop - an unlikely occurrence, as it is hard to think of a guideline for that.

Our immediate need for an automatic tuning and adaptation procedure for cascade controllers came from the design of controllers for quadrotors, so current research is focused on the application of the above methodology to the attitude control of a quadrotor.

REFERENCES

[1] H. Hjalmarsson, S. Gunnarsson, and M. Gevers, "A convergent iterative restricted complexity control design scheme," in *Proc. 33rd Conference on Decision and Control*. Lake Buena Vista, Fl, USA: New York: IEEE, December 1994, pp. 1735–1740.

[2] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin, "Iterative feedback tuning: theory and applications," *IEEE control systems magazine*, vol. 18, no. 4, pp. 26–41, 1998.

[3] H. Hjalmarsson, "Iterative feedback tuning - an overview," *International journal of adaptive control and signal processing*, vol. 16, no. 5, pp. 373–395, 2002.

[4] A. S. Bazanella, L. Campestrini, and D. Eckhard, *Data-driven controller design: the H2 approach*. Springer Science & Business Media, 2011.

[5] M. C. Campi, A. Lecchini, and S. M. Savaresi, "Virtual reference feedback tuning: a direct method for the design of feedback controllers," *Automatica*, vol. 38, no. 8, pp. 1337–1346, 2002.

[6] A. Karimi, L. Miskovic, and D. Bonvin, "Iterative correlation-based controller tuning," *International journal of adaptive control and signal processing*, vol. 18, no. 8, pp. 645–664, 2004.

[7] L. C. Kammer, R. R. Bitmead, and P. L. Bartlett, "Direct iterative tuning via spectral anaysis," *Automatica*, vol. 36, no. 9, pp. 1301–1307, 2000.

[8] L. Campestrini, D. Eckhard, A. S. Bazanella, and M. Gevers, "Model reference control design by prediction error identification," in *16th IFAC Symposium on System Identification*. Brussels: IFAC, 2012, pp. 1478–1483.

[9] G. C. Goodwin and K. S. Sin, *Adaptive filtering, prediction and control*. Englewood Cliffs, NJ, EUA: Prentice Hall, 1984.

[10] D. Eckhard and A. S. Bazanella, "Robust convergence of the steepest descent method for data-based control," *International Journal of Systems Science*, vol. 43, no. 10, pp. 1969–1975, 2012.

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

[12] G. Solari and M. Gevers, "Unbiased estimation of the hessian for iterative feedback tuning (IFT)," in *Proc. 43rd IEEE Conference on Decision and Control*, vol. 2, 2004, pp. 1759–1760 Vol.2.

[13] A. B. Corripio, *Tuning of Industrial Control Systems*, 2nd ed. ISA, 2000.

[14] Y. Lee, S. Park, and M. Lee, "Pid controller tuning to obtain desired closed loop responses for cascade control systems," *Industrial & engineering chemistry research*, vol. 37, no. 5, pp. 1859–1865, 1998.

[15] A. Ralston and P. Rabinowitz, *A first course in numerical analysis*. Courier Corporation, 2012.