Proceedings of the 2006 IEEE
International Conference on Control Applications
Munich, Germany, October 4-6, 2006

An automatic tuning method for cascade control systems

Antonio Visioli and Aurelio Piazzi

Abstract— **In this paper we present a new design method for cascade control systems. The proposed (open-loop) identification procedure allows to perform a one stage tuning of the PID controllers, whose parameters are selected in order to ensure a good load disturbance rejection performance. Then, the model of the overall control system is reduced in order to apply the closed-form solution of an analytical stable inversion procedure as a command function. Thus, a high (predefined) performance is achieved also in the set-point following task. Simulation results show the effectiveness of the methodology.**

I. INTRODUCTION

The achievement of satisfactory performances in addition to a fast (low-cost) setting-up are essential requirements for a control system to be adopted in the industrial context. Proportional-Integral-Derivative (PID) controllers are widely adopted in industry because acceptable performances can be obtained by adopting one of the many (auto-)tuning methods, based on a simple modelling of the plant, that have been devised in the last sixty years, or even by a trial and error procedure. When more demanding control specifications are required for a given application, PID controllers can be still adopted as a basis of more complex control schemes where couplings between simple control systems are exploited.

A typical example in this context is cascade control, which is available as a standard tool in almost all industrial process controller. It consists of employing an additional sensor in order to separate the fast and the slow dynamics of the process to obtain a fast load disturbance rejection. The resulting control scheme involves therefore two nested loops with two PID controllers, as shown in Figure 1. The controller of the inner loop is the so called secondary controller, while that of the outer loop is the primary controller. Thus, the design of the overall control system involves the tuning of two (PID) controllers. Usually, the secondary controller is tuned first by setting the primary controller in manual mode. Then, the primary controller is tuned while the previously tuned secondary controller is applied to the inner loop. It appears how the design procedure is longer and more complicated than that of a standard single-loop (PID) control.

In order to provide users with an effective aid in the design of the cascade control system, a relay feedback autotuning technique has been proposed in [1]. It basically consists of applying the typical relay feedback based autotuning [2] to the two loops, sequentially as in the conventional approach. Thus, the procedure is still time consuming. In order to avoid the sequential nature of the tuning procedure a simultaneous online automatic tuning technique has been proposed in [3]. However, it is assumed that the cascade controller had been already (roughly) tuned and the devised method aims mainly at (significantly) improving the tuning. Further, tuning rules have been proposed in [4] (and they have been subsequently improved in [5]) for the primary and secondary controller, but it has not been specified how the procedure can be automated.

This work was partially supported by MIUR scientific research funds

A. Visioli is with Dipartimento di Elettronica per l'Automazione, University of Brescia, Italy, antonio.visioli@ing.unibs.it

A. Piazzi is with Dipartimento di Ingegneria dell'Informazione, University of Parma, Italy aurelio@ce.unipr.it

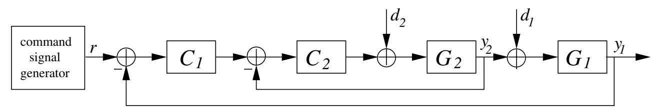

Fig. 1. Typical cascade control system.

In this paper we propose a new automatic tuning methodology that takes into account the load rejection performances, which are of main concerns in cascade control loops, and the set-point following ones. It consists of first estimating the two dynamics of the process by evaluating an open loop step response. Then, the secondary controller is tuned and, subsequently, the reduced model of the process seen by the primary controller is determined in order to tune the primary controller and at the same time to calculate the command reference input. This can be determined by exploiting a closed-form expression that results from the use of transition polynomials [6] and the application of a stable inversion procedure [7], [8]. The use of this command function instead of the typical step signal allows to recover the set-point following performance when the PID controllers are tuned by taking into account the load disturbance rejection task and, most of all, to achieve a predefined performance in the set-point response, almost independently from the PID controllers tuning [7].

II. CASCADE CONTROL

In a typical cascade control scheme the process has one input and two outputs. Indeed, in order to provide an effective disturbance rejection, an additional sensor is employed in order to separate as much as possible the fast dynamics of the process from the slow one (i.e. the one with the slowest poles and the nonminimum-phase part) [9]. The considered scheme is shown in Figure 1. The process transfer function is denoted by $G(s) = G_2(s)G_1(s)$ where $G_2(s)$ is the transfer function that models the fast dynamics of the process and $G_1(s)$ is the one of the slow part, $C_2(s)$ is the secondary controller and $C_1(s)$ is the primary controller. It is assumed that the controllers are of PID type, i.e.:

$$C_1(s) = K_{p1}\left(1+\frac{1}{T_{i1}s}+T_{d1}s\right)\frac{1}{T_{f1}s+1} \qquad (1)$$

and

$$C_2(s) = K_{p2}\left(1+\frac{1}{T_{i2}s}+T_{d2}s\right)\frac{1}{T_{f2}s+1} \qquad (2)$$

where K_{p1} and K_{p2} are the proportional gains, T_{i1} and T_{i2} are the integral time constants, T_{d1} and T_{d2} are the derivative time constants and T_{f1} and T_{f2} are the time constants of the filter that makes the controller proper. Finally, signals d_2 and d_1 represent disturbances affecting the fast and the slow part of the process respectively and r represents the command signal to be provided to the control scheme in order to achieve a predefined output transition (i.e. a set-point change).

It appears that the effectiveness of a cascade control scheme is due to the fact that disturbances affecting the (fast) secondary loop are effectively compensated before they affect the main process output y_1.

III. THE NEW DESIGN METHOD

A. Identification procedure

The open loop identification procedure can be performed by applying a step signal to the process $G(s)$. A first-order plus dead time (FOPDT) transfer function of the fast dynamics of the process can be estimated by evaluating the step response of $G_2(s)$, for example by applying the well-known area method [10], which is robust to the measurement noise. Thus, the following transfer function is estimated:

$$G_2(s) = \frac{K_2}{T_2 s+1}e^{-L_2 s} \qquad (3)$$

At the same time, a model for the slow dynamics of the process can be estimated by considering its input signal y_2 and its output signal y_1 and by applying a least squares procedure, such as the one proposed in [11] which is based on the integrated input and output signals and therefore it is inherently robust to measurement noise.

It has to be noted that, because of the different dynamics of G_2 and G_1, the step response of G_2 is indeed a sufficiently exciting signal to be adopted as an input signal for the least square based estimation of $G_1(s)$. Thus, the parameters of the following third-order transfer function are estimated:

$$G_1(s) = \frac{a_2 s + a_1 s + a_0}{s^3+b_2 s^2+b_1 s+b_0} \qquad (4)$$

Remark 1. It is worth stressing at this point that the choice of using a FOPDT model for G_2 is motivated by the fact that many industrial processes can be well described by this model and this allows the application of one of the many tuning rules that are based on this model (see subsection III-B). The choice of the transfer function (4) is motivated by the fact that the presence of a second order numerator polynomial allows to identify a nonminimum-phase dynamics (and/or a time-delay) with a good accuracy. On the other hand, there is no point in using a higher-order model, as it will be reduced for the purpose of the tuning of the primary controller (see subsections III-C and III-D).

B. Secondary controller tuning

The secondary controller C_2 can be tuned by using any standard tuning rule that is based on a FOPDT model (3) of the plant. However, taking into account the main purpose of the use of a cascade control structure, it is sensible to adopt a tuning rule devoted to the rejection of load disturbances. Here it is proposed for example to adopt the KT (Kappa-Tau) tuning rules proposed in [10]. They consider the relative dead time defined as:

$$\theta_2 = \frac{L_2}{T_2+L_2} \qquad (5)$$

and then, the PID parameters are calculated as:

$$K_{p2} = 3.8\exp(-8.4\theta_2+7.3\theta_2^2)\frac{T_2}{K_2 L_2} \qquad (6)$$

$$T_{i2} = 5.2\exp(-2.5\theta_2-1.4\theta_2^2)L_2 \qquad (7)$$

$$T_{d2} = 0.89\exp(-0.37\theta_2-4.1\theta_2^2)L_2 \qquad (8)$$

It has to be noted that two kinds of tuning rules are provided in [10], depending on the maximum sensitivity of the closed-loop system (namely, the trade-off between aggressiveness and robustness is handled). In this context, it is chosen to pursue the robustness of the control system, because the model reductions and approximations done during the whole methodology have to be taken into account.

Note also that the presence of the set-point weight, which is usually employed to recover the set-point following performances, is not exploited in this case, because a step reference signal is obviously never applied to the inner loop.

Finally, it is worth stressing that this tuning rules do not provide an explicit value for the filter time constant T_f. Its value can be however selected easily so that the filter dynamics does not influence the predefined performance (i.e. such as a high-frequency pole results).

C. Model reduction

Once the tuning of the secondary controller has been completed, the model of the process seen by the primary controller has to be determined. In particular, despite the possible presence of a high-order and nonminimum-phase dynamics in G_1, it is sensible to determine a FOPDT transfer function, denoted as

$$G_r(s) = \frac{K_r}{T_r s+1}e^{-L_r s}, \qquad (9)$$

because this model is useful in order to apply a tuning rule for the (PID) primary controller C_1 and to use a closed-form expression for the determination of the command signal r. For this purpose, the transfer function given by the series of the inner loop and $G_1(s)$, i.e.

$$G_m(s) := \frac{C_2(s)G_2(s)}{1+C_2(s)G_2(s)}G_1(s) \qquad (10)$$

is first calculated. In this context, the delay term $e^{-L_2 s}$ is approximated by a first order Padè approximation in order to have a rational transfer function. Then, the procedure presented in [11] is adopted to determine the reduced model (9), i.e. to determine the three parameters K_r, T_r and L_r. Specifically, the system gain is calculated as:

$$K_r = G_m(0), \tag{11}$$

while the time constant is subsequently calculated by solving, with a least squares method, the following equation:

$$T_r = \frac{\sqrt{K_r^2 - |G_m(j\omega_i)|^2}}{|G_m(j\omega_i)|\omega_i} \quad 0 < \omega < \omega_1 < \cdots \leq \omega_c \tag{12}$$

where ω_c is the critical frequency of the system (i.e. the frequency for which $|G_m(j\omega_c)| = 1$) and the ω_i's are a sufficient number of frequencies located with equal intervals between 0 and ω_c. In case the equation $|G_m(j\omega_c)| = 1$ has no solution (i.e. it is $|G_m(j\omega)| < 1$ for $\omega > 0$), then ω_c is chosen as the frequency for which $|G_m(j\omega_c)| = |G_m(0)| - 3\text{db}$. Finally, the dead time L_r is determined by imposing that the phase of $G_m(j\omega_c)$ is equal to that of $G_r(j\omega_c)$, i.e. by calculating

$$L_r = -\frac{\arg G_m(j\omega_c) + \arctan(\omega_c T_r)}{\omega_c}. \tag{13}$$

The rationale of using this model reduction method is that it provides a sufficiently good approximation in the whole bandwidth of the system, so that the subsequent tuning of the primary controller will be appropriate and the dynamic inversion procedure will be effective, because the determined command signal will have a frequency content in that range [13].

D. Primary controller tuning

Once a FOPDT transfer function $G_r(s)$ has been obtained, the primary controller can be tuned straightforwardly by adopting again one of the available tuning rules based on this model. However, it has to be taken into account that usually process G_1 contains the high-order dynamics of the whole process and its nonminimum-phase part and therefore it might be difficult to obtain a high performance with a PID controller (indeed, this is the reason for the application of a cascade control structure). Thus an aggressive tuning of the primary controller is not convenient in an autotuning framework, also taking into account that the dynamic inversion procedure is devised in order to recover the set-point following performances, which are obviously strongly affected by the choice of the parameters of C_1.

Based on these considerations, it is suggested to actually adopt a PI controller as a primary controller, where the parameters are selected again according to the KT method, i.e.:

$$K_{p1} = 0.41 \exp(-0.23\theta_r + 0.019\theta_r^2)\frac{T_r}{K_r L_r} \tag{14}$$

$$T_{i1} = 5.7 \exp(1.7\theta_r - 0.69\theta_r^2) L_r \tag{15}$$

where $\theta_r = L_r/(T_r + L_r)$. Note that, as for the secondary controller, the more robust tuning rule has been selected and the set-point weight has not been considered.

E. Alternative simultaneous tuning method

Instead of tuning the primary controller after having reduced the model (10), the two controllers can be tuned simultaneously by employing the tuning rules given in [5], which are based on an Internal Model Control (IMC) approach. In this case, a FOPDT model of G_1 and G_2 has to be available. The technique described in the previous subsections can be easily modified by first reducing model (4) to a FOPDT form and then by applying the proposed tuning rules. Then, a FOPDT model of the system (10) has to be in any case calculated, again with the technique described in subsection III-C, in order to exploit the closed-form expression of the inversion-based command function.

F. Command signal design

A noncausal approach for the improvement of the set-point following performances of PID controllers has been proposed in [7]. It is based on the application of an inversion-based command signal, instead of the traditional step signal, to the closed-loop system. The noncausal command signal is calculated by adopting a stable dynamic inversion analytical procedure [8] to the closed-loop system, once a polynomial function of suitable order is selected as a desired output function. Differently from the traditional (causal) set-point filtering, the noncausal approach allows to prevent the occurrence of a high value of the rise time (see subsection IV-C). A salient feature of the methodology is that predefined performances are obtained almost independently from the values of the PID parameters.

Formally, the desired output function that defines the transition from a set-point value y_1^s (in the following it will be assumed $y_1^s = 0$ without loss of generality) to another y_1^f, to be performed in the time interval $[0, \tau]$, is chosen as a third order (monotonic) transition polynomial parameterized by the user-chosen transition time τ [6], i.e.:

$$y_d(t;\tau) = y_1^f \left(-\frac{2}{\tau^3}t^3 + \frac{3}{\tau^2}t^2\right). \tag{16}$$

Outside the interval $[0, \tau]$ the function $y(t;\tau)$ is equal to 0 for $t < 0$ and equal to y_1^f for $t > \tau$. It has to be noted that the chosen function satisfies boundary conditions and its order implies that a continuous command reference function is eventually obtained.

Then, the closed-form expression of the command signal $r(t; K_r, T_r, L_r, K_{p1}, T_{i1}, T_{d1}, T_{f1}, \tau)$, defined over $(-\infty, +\infty)$, that provides the desired output function (16) is calculated by applying a stable input-output inversion procedure to the nominal closed-loop system

$$\frac{C_1(s)G_r(s)}{1 + C_1(s)G_r(s)} \tag{17}$$

where the delay term is approximated by a second order Padè approximation.

Actually, from a practical point of view, in order to use the synthesized function $r(t; K_r, T_r, L_r, K_{p1}, T_{i1}, T_{d1}, T_{f1}, \tau)$, it is necessary to truncate it, resulting therefore in an approximate generation of the desired output $y_d(t;\tau)$. However, this

can be done with arbitrarily precision, though in general it exhibits a pre- and a post-action time intervals [7].
It is worth stressing again that the overall stable dynamic inversion procedure can be performed by means of a symbolic computation, i.e. the actual command signal to be applied to the cascade control scheme is determined by substituting the actual value of the parameters K_r, T_r, L_r, K_{p1}, T_{i1}, T_{d1}, T_{f1} and τ into the resulting closed-form expression.
Remark 2. The design parameter τ represents a very desirable feature from a user point of view, as it allows to handle the trade-off between performance, robustness and control activity. Its value can be selected by solving an optimization problem where the transition time has to be minimized by taking into account the actuator constraints [8] or, more simply, by selecting a ratio between the bandwidth of the open-loop system and that of the closed-loop one, from which the value of τ can be calculated easily.
Remark 3. It has to be noted that the input-output inversion procedure presented in [8] can be applied to any stable rational transfer function. Thus, in case a more accurate (high-order) model of the plant is available, it can be exploited in order to achieve a higher performance in the set-point following task. However, a closed-form expression of the command input can be derived only for a FOPDT process and therefore the use of a high-order model is affective in the context of the design of a cascade controller more than in an automatic tuning framework.
Remark 4. Indeed, the devised methodology can be applied straightforwardly in the context of the design of a cascade control system, when the automatic tuning functionality is not exploited. In this case, the PID tuning can be still based on the models obtained by performing the identification procedure described in subsection III-A and, as the overall method does not need to be general, a more aggressive tuning could be possibly selected. In any case, the determination of the closed-loop command signal can be performed automatically once any choice of PID parameters has been done (note that, for this reason, the derivative time constant T_{d1} has been included, for generality, in the closed-form expression of the command input, despite in subsection III-D it is suggested to use a PI controller).

IV. SIMULATION RESULTS

A. Example 1

In order to verify the effectiveness of the proposed methodology, the following process model has been considered first:

$$G_1(s) = \frac{1}{100s+1}e^{-40s} \qquad G_2(s) = \frac{2}{20s+1}e^{-4s} \tag{18}$$

By applying the identification procedure and the suggested KT tuning method it results $K_{p2} = 2.85$, $T_{i2} = 13.21$, $T_{d2} = 2.99$, $K_r = 1$, $T_r = 99.65$, $L_r = 42.5$, $K_{p1} = 0.90$, $T_{i1} = 378.8$, $T_{d1} = 0$. In order to compare the results with other possible tuning methods, a (refined) Ziegler-Nichols (RZN) rule has also been considered [10], i.e.

$$K_{p2} = 1.2\frac{T_2}{K_2 L_2} \qquad T_{i2} = 2L_2 \qquad T_{d2} = 0.5L_2, \tag{19}$$

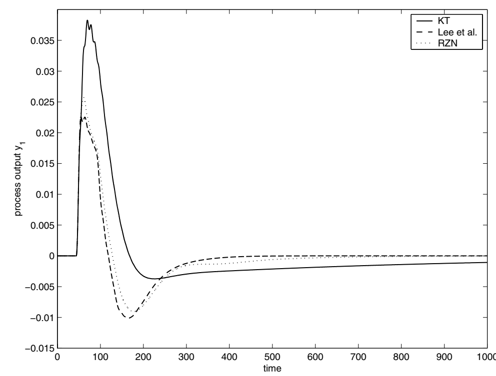

Fig. 2. Load disturbance d_2 responses for example 1.

$$K_{p1} = 0.63\frac{T_r}{K_r L_r} \qquad T_{i1} = 3.2L_r \qquad T_{d1} = 0, \tag{20}$$

In this case it results $K_{p2} = 2.0$, $T_{i2} = 8.03$, $T_{d2} = 2.0$, $K_r = 1$, $T_r = 99.68$, $L_r = 41.5$, $K_{p1} = 1.51$, $T_{i1} = 132.8$, $T_{d1} = 0$. The method by Lee et al. described in subsection III-E, has been considered as well. In this context the IMC filters time constants (see [5] for details) has been selected as

$$\lambda_1 = L_2 + L_r \qquad \lambda_2 = 0.5L_2. \tag{21}$$

It results $K_{p2} = 3.38$, $T_{i2} = 8.60$, $T_{d2} = 1.55$, $K_{p1} = 1.84$, $T_{i1} = 96.0$, $T_{d1} = 1.55$, $\lambda_1 = 44.0$, $\lambda_2 = 2.0$, $T_{f1} = 6.76$. It has to be noted that this method requires a (causal) set-point step filter that has been applied for the case of the set-point response, but it has been removed when the inversion-based command signal has been applied.
The load disturbance d_2 response, which is of main concern, for the different controller parameters are plotted in Figure 2. For completeness, the load disturbance d_1 responses are also shown in Figure 3. The process outputs y_1 that are obtained by applying a standard step signal to the set-point are reported in Figure 4, so that the differences in the tuning of the cascade controller can be appreciated.
The closed-form expression of the command function is then evaluated for the three cases,with $\tau = 100$ and the resulting command signals are plotted in Figure 5. The process outputs that are obtained by applying these signals to the closed-loop system are shown in Figure 6. It appears that, despite the different controller parameters, the achieved performance is practically the same for the three methods. Note that for the sake of clarity the time range of the plots has been changed in the different cases.

B. Example 2

As a second example, the following process model, which exhibits a significant high-order and nonminimum-phase dynamics has been considered [5]:

$$G_1(s) = \frac{10(-5s+1)}{(30s+1)^3(10s+1)^2}e^{-5s} \tag{22}$$

$$G_2(s) = \frac{3}{13.3s+1}e^{-3s} \tag{23}$$

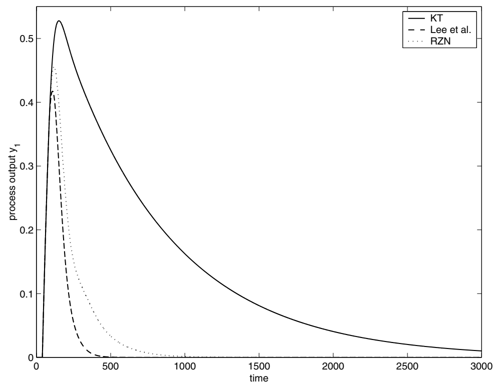

Fig. 3. Load disturbance d_1 responses for example 1.

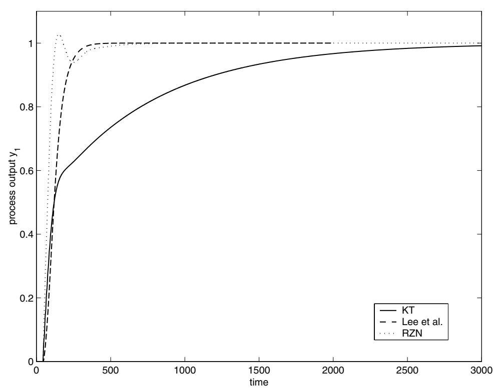

Fig. 4. Step set-point responses for example 1.

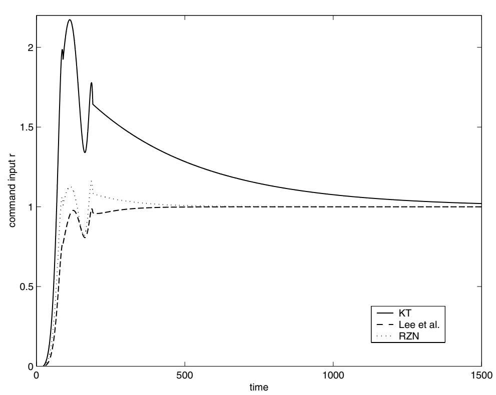

Fig. 5. Inversion-based command signals for example 1.

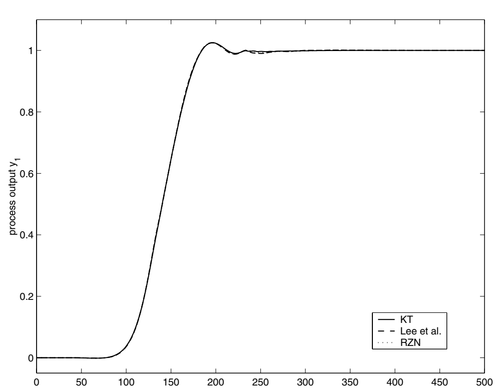

Fig. 6. Inversion-based set-point responses for example 1.

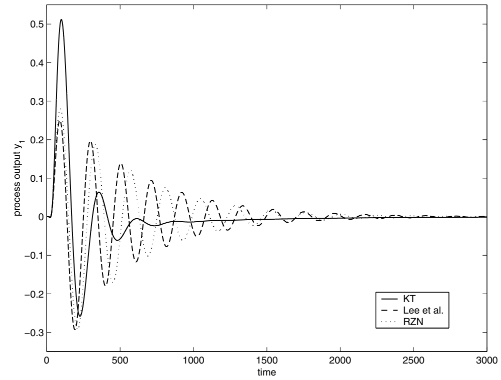

Fig. 7. Load disturbance d_2 responses for example 2.

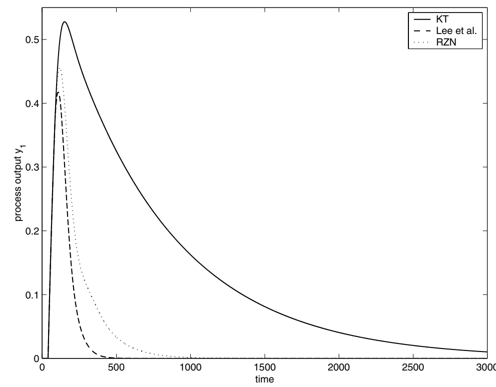

Fig. 8. Load disturbance d_1 responses for example 2.

With the suggested KT tuning method (5)-(8) and (14)-(15) it results $K_{p2} = 1.52$, $T_{i2} = 9.41$, $T_{d2} = 2.18$, $K_r = 10.0$, $T_r = 145.5$, $L_r = 52.2$, $K_{p1} = 0.11$, $T_{i1} = 162.8$, $T_{d1} = 0$. Conversely, if the refined Ziegler-Nichols rules are applied, we obtain $K_{p2} = 1.76$, $T_{i2} = 6.03$, $T_{d2} = 1.51$, $K_r = 10$, $T_r = 145.8$, $L_r = 50.9$, $K_{p1} = 0.18$, $T_{i1} = 162.8$, $T_{d1} = 0$. Finally, by following the method of Lee et al. it results $K_{p2} = 1.99$, $T_{i2} = 6.34$, $T_{d2} = 1.15$, $K_{p1} = 0.23$, $T_{i1} = 124.5$, $T_{d1} = 16.4$, $T_{f1} = 4.96$. Load disturbances responses are plotted in Figures 7 and 8 for d_2 and d_1 respectively. Responses to the standard step change in the set-point signal are reported in Figure 9. By fixing $\tau = 400$, the command signals plotted in Figure 10 are determined. When applied, the process outputs shown in Figure 11 are obtained. Note again that different time ranges are adopted in the different plots. It appears that the inversion-based approach allows to improve the set-point following performance and it provides almost the same performance despite the different closed-loop dynamics.

Remark 5. The presented results related to the method of Lee et al. are different from those shown in [5]. This is due to the different estimated model and the consequent different calculated PID parameters.

C. Discussion

The effectiveness of the approach has been shown in the presented example. It is worth noting that the suggested

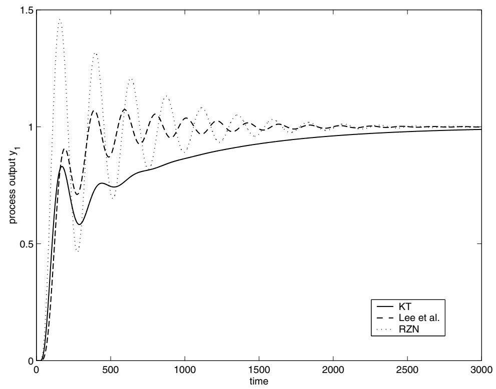

Fig. 9. Step set-point responses for example 2.

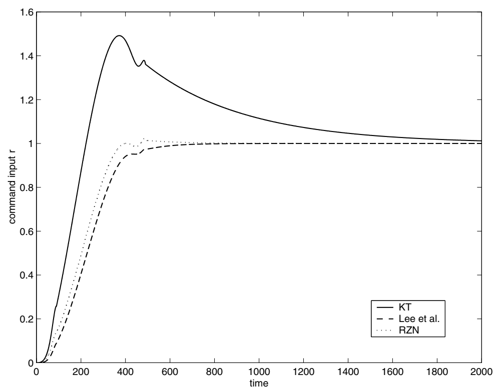

Fig. 10. Inversion-based command signals for example 2.

use of the KT tuning rules, especially with respect to the primary loop, aims at providing a robust performance, which is often of main concern for users, and therefore to cope with the possible presence of a high-order and nonminimum-phase dynamics. This is evident from the results shown. Actually, from example 2 it appears that the other tuning rules yield to somewhat oscillatory responses, although the performances they provide in example 1, where both G_1 and G_2 have a FOPDT dynamics, are generally better. Indeed, a very sluggish set-point response is obtained in example 1 with the KT tuning rules in case a step signal is applied

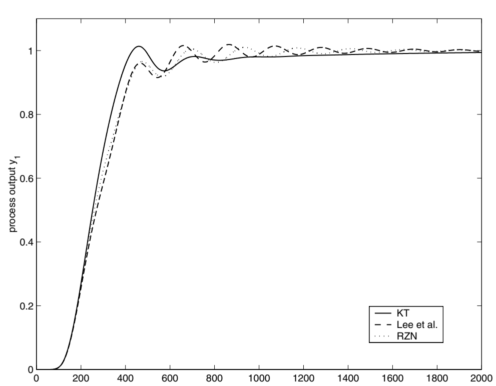

Fig. 11. Inversion-based set-point responses for example 2.

to the set-point. However, by substituting the classic step with the inversion-based command signal, the performance is recovered (note that this is not possible in general by filtering the set-point with a causal filter).

From another point of view, being general, the proposed autotuning procedure provides results that might be improvable in the single cases. In this context the devised approach is in any case useful in providing the appropriate command function even if the controllers are re-tuned (see Remark 4). Finally, it has to be stressed again that from the presented results it appears that the main merit of the method is the achievement of predefined (high) performances in the set-point response, as it allows to obtain very similar process responses despite significantly different PID parameters.

V. CONCLUSIONS

A new method for the automatic tuning of a cascade control system has been presented. Differently from those proposed in the literature, due to the devised open-loop identification procedure, it allows the simultaneous tuning of the PID controllers and the determination of a command signal, via a closed-form expression, that guarantees that a predefined (high) performance is obtained in the set-point following task (a high load disturbance rejection performance is naturally achieved through the choice of the cascade control structure). The presented results have demonstrated that the (noncausal) approach is effective also in the presence of a significant high-order and nonminimum-phase process dynamics.

REFERENCES

[1] C. C. Hang, A. P. Loh, V. U. Vasnani, "Relay feedback autotuning of cascade controllers" *IEEE Transactions on Control Systems Technology*, Vol. 2, pp. 42-45, 1994.

[2] C. C. Yu, *Autotuning of PID controllers*, Springer-Verlag, London, UK, 1999.

[3] K. K. Tan, T. H. Lee, R. Ferdous "Simultaneous online automatic tuning of cascade control for open loop stable processes", *ISA Transactions*, Vol. 39, pp. 233-242, 2000.

[4] Y. Lee, S. Park, M. Lee, "PID controller tuning to obtain desired closed loop responses for cascade control systems", *Industrial Engineering and Chemistry Research*, Vol. 37, pp. 1859-1865, 1998.

[5] Y. Lee, S. Oh, S. Park, "Enhanced control with a general cascade control structure", *Industrial Engineering and Chemistry Research*, Vol. 41, pp. 2679-2688, 2002.

[6] A. Piazzi, A. Visioli, "Optimal noncausal set-point regulation of scalar systems", *Automatica*, Vol. 37, pp. 121-127, 2005.

[7] A. Piazzi, A. Visioli, "A noncausal approach for the improvement of PID control", *Proc. American Cont. Conf.*, Boston (USA), 2004.

[8] A. Piazzi, A. Visioli, "Using stable input-output inversion for minimum-time feedforward constrained regulation of scalar systems", *Automatica*, Vol. 41, pp. 305-313, 2005.

[9] P. R. Krishnaswami, G. P. Rangaiah, R. K. Jha, P. D. Deshpande, "When to use cascade control", *Industrial Engineering and Chemistry Research*, Vol. 29, pp. 2163-2166, 1990.

[10] K. J. Aström, T. Hägglund, *PID controllers - theory, design and tuning*, ISA Press, Research Triangle Park, USA, 1995.

[11] S. W. Sung, I.-B. Lee, B.-K. Lee, "On-line process identifi cation and automatic tuning method for PID controllers", *Chemical Engineering Science*, Vol. 53, pp. 1847-1859, 1998.

[12] M. Zhuang, D. P. Atherton, "Automatic tuning of optimum PID controllers", *IEE Proceedings - Control Theory and Applications*, Vol. 140, pp. 216-224, 1993.

[13] A. Piazzi, A. Visioli, "Optimal inversion-based control for the set-point regulation of nonminimum-phase uncertain scalar systems", *IEEE Transactions on Automatic Control*, Vol. 46, pp. 1654-1659, 2001.