A Method for PID Controller Tuning Using Nonlinear Control Techniques

Prashant Mhaskar, Nael H. El-Farra, and Panagiotis D. Christofides
Dept. of Chemical Engineering, University of California, Los Angeles, CA 90095

DOI 10.1002/aic.10566
Published online August 19, 2005 in Wiley InterScience (www.interscience.wiley.com).

Keywords: PID controller, tuning methods, nonlinear control, process control

Introduction

The majority (>90%) of the regulatory loops in the process industries use conventional proportional–integral–derivative (PID) controllers. Because of the abundance of PID controllers in practice and the varied nature of processes that the PID controllers regulate, extensive research studies have been dedicated to the analysis of the closed-loop properties under PID controllers and to devising new and improved tuning guidelines for them, focusing on closed-loop stability, performance, and robustness[1-7] (also see the survey papers[8,9]). Most of the tuning rules are based on obtaining linear models of the system, either through running step tests or by linearizing a nonlinear model around the operating steady state, and then computing values of the controller parameters that incorporate stability, performance, and robustness objectives in the closed-loop system.

Although the use of linear models for the PID controller tuning makes the tuning process easy, the underlying dynamics of many processes are often highly complex because of such phenomena as the inherent nonlinearity of the underlying chemical reaction or operating issues such as actuator constraints, time delays, and disturbances. Ignoring the inherent nonlinearity of the process when setting the values of the controller parameters may result in the controller's inability to stabilize the closed-loop system and may call for extensive retuning of the controller parameters.

The shortcomings of classical controllers in dealing with complex process dynamics, together with the abundance of such complexities in modern-day processes, have been an important driving force behind the significant and growing body of research work within the area of nonlinear process control over the past two decades, leading to the development of several practically implementable nonlinear control strategies that can deal effectively with a wide range of process control problems such as nonlinearities, constraints, uncertainties, and time-delays[10-13] (also see the books by Sepulchre et al.,[14] Khalil,[15] and Isidori[16]). Whereas process control practice has the potential to benefit from these advances through the direct implementation of the developed nonlinear controllers, an equally important direction in which process control practice stands to gain from these developments lies in investigating how nonlinear control techniques can be used for the improved tuning of classical PID controllers. This is an appealing goal because it allows control engineers to potentially take advantage of the improved stability and performance properties provided by nonlinear control without actually forsaking the ubiquitous conventional PID controllers or redesigning the control system hardware.

There has been some research effort toward incorporating nonlinear control tools in the design of PID controllers. For example, in Wright et al.[17] it is shown that controllers resulting from nonlinear model–based control theory can be put in a form that looks like the PI or PID controllers for first and second-order systems. Other examples include Benaskeur and Desbiens,[18] in whose work adaptive PID controllers are designed using a backstepping procedure, and Chang et al.,[19] where a self-tuning PID controller is derived using Lyapunov techniques. In these works, however, even though the resulting controller has the same structure as that of a PID controller, the controller parameters (gain, K_c; integral time constant, τ_I; and derivative time constant, τ_D) are not constant but functions of the error or process states. Even though such analysis provides useful analogies between nonlinear controllers and PID controllers, implementation of these control designs would require changing the control hardware in a way that allows the tuning parameter values to be continuously changed while the process is in operation.

Motivated by the above considerations, we propose in this work a two-level, optimization-based method for the derivation of tuning guidelines for PID controllers that take nonlinear

Correspondence concerning this article should be addressed to P. D. Christofides at pdc@seas.ucla.edu.

process behavior explicitly into account. The central idea behind the proposed method is the selection of the tuning parameters in a way that has the PID controller emulate, as closely as possible, the control action and closed-loop response obtained under a given nonlinear controller, for a broad set of initial conditions and set-point changes. To this end, classical tuning guidelines (typically derived on the basis of linear approximations, running open or closed-loop tests) are initially used in the first level to obtain reasonable bounds on the range of stabilizing tuning parameters over which the search for the parameters best matching the PID and nonlinear controllers is to be conducted. In addition to stability, performance and robustness considerations for the linearized closed-loop system can be introduced in the first level to further narrow down the parameter search range. The bounds obtained from the first level are then incorporated as constraints on the optimization problem solved at the second level to yield a set of tuning parameter values that enforce closed-loop behavior under the PID controller that closely matches the closed-loop behavior under the nonlinear controller. Implications of the proposed method, as a transparent and meaningful link between the classical and nonlinear control domains, as well as possible extensions of the tuning guidelines and other implementation issues, are discussed. Finally, the proposed tuning method is demonstrated through a chemical reactor example.

Two-Level PID Tuning Method

In this work, we consider continuous-time single-input/single-output (SISO) nonlinear systems, with the following state–space description

$$\dot{x}(t) = f[x(t)] + g[x(t)]u(t)$$

$$y = h(x) \tag{1}$$

where $x = [x_1 \cdots x_n]' \in \mathbb{R}^n$ denotes the vector of state variables and x' denotes the transpose of x; $y \in \mathbb{R}$ is the controlled output; $u \in \mathbb{R}$ is the manipulated input; $f(\cdot)$ is a sufficiently smooth nonlinear vector function, with $f(0) = 0$; $g(\cdot)$ is a sufficiently smooth nonlinear vector function; and $h(\cdot)$ is a sufficiently smooth scalar nonlinear function, with $h(0) = 0$. Throughout the paper, the notation $L_f h$ denotes the standard Lie derivative of a scalar function $h(\cdot)$ with respect to the vector function $f(\cdot)$, that is, $L_f h(x) = (\partial h/\partial x) f(x)$.

The basic idea behind the proposed approach is the design (but not implementation) of a nonlinear controller that achieves the desired closed-loop response, and then the tuning of the PID controller parameters so as to best "emulate" the control action and the closed-loop process response under the nonlinear controller, subject to constraints derived from classical PID controller tuning rules. These ideas are described algorithmically below (see also Figure 1):

(1) Construct a nonlinear process model and derive a linear model around the operating steady state (either through linearization or by running step tests).

(2) On the basis of the linear model, use classical tuning guidelines to determine bounds on the stabilizing range of the tuning parameters, K_c, τ_I, and τ_D.

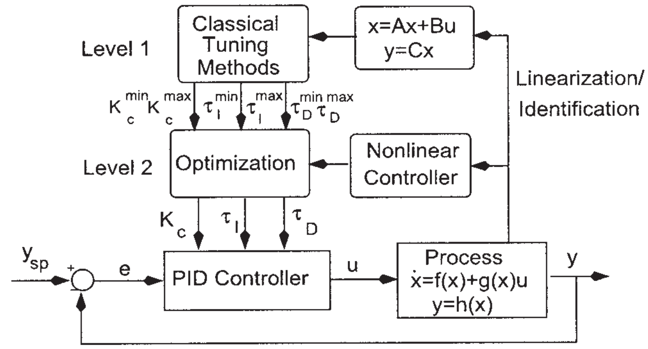

Figure 1. Implementation of the proposed two-level, optimization-based tuning method.

(3) Using the nonlinear process model and the desired closed-loop response, design a nonlinear controller.

(4) For a set-point change, compute off-line, through simulations, the input trajectory $[u_{nl}(t)]$ "prescribed" by the nonlinear controller over the time interval, $[0, t_{final}]$, required to achieve the set-point change. Compute the corresponding closed-loop output profile under the nonlinear controller, $y_{nl}(t)$.

(5) Compute the PID controller tuning parameters, K_c, τ_I, and τ_D, as the solution to the following optimization problem

$$\min_{K_c,\tau_I,\tau_D} J = \int_0^{t_{final}} \{[y_{nl}(t) - y_{PID}(t)]^2 + [u_{nl}(t) - u_{PID}(t)]^2\}dt \tag{2}$$

$$\text{s.t.} \qquad u_{PID}(t) = K_c\left[e + \frac{\int_0^t e(t')dt'}{\tau_I} + \tau_D \frac{de}{dt}\right]$$

$$e(t) = y_{sp} - y_{PID}$$

$$\dot{x}(t) = f[x(t)] + g[x(t)]u_{PID}(t)$$

$$y_{PID} = h(x)$$

$$\alpha_1 K_c^c \leq K_c \leq \alpha_4 K_c^c$$

$$\alpha_2 \tau_I^c \leq \tau_I \leq \alpha_5 \tau_I^c$$

$$\alpha_3 \tau_D^c \leq \tau_D \leq \alpha_6 \tau_D^c$$

$$(K_c,\ \tau_I,\ \tau_D) = \text{argmin}(J) \tag{3}$$

where y_{sp} is the desired set point; y_{nl} and u_{nl} are the closed-loop process response and control action, respectively, under the nonlinear controller; K_c^c, τ_I^c, and τ_D^c are parameter values obtained using (first-level) tuning rules based on linear models; and $0 \leq \alpha_i \leq 1$, $i = 1, 2, 3$ and $1 \leq \alpha_i < \infty$, $i = 4, 5, 6$ are design parameters.

Remark 1. The optimization problem of Eqs. 2 and 3 computes values for K_c, τ_I, and τ_D such that the closed-loop control action and process response under the PID controller are similar to those under the nonlinear controller, while being within acceptable ranges of the values obtained from the classical tuning methods. The method thus allows fine-tuning the closed-loop performance under the PID controller to mimic that of the nonlinear controller. Note that, in principle, the PID controller parameters could have been tuned to mimic any desired arbitrarily chosen closed-loop behavior. Mimicking the behavior of the nonlinear controller, however, is a meaningful objective because the nonlinear controller uses the nonlinear system dynamics in generating the prescribed input and output response, and therefore provides a "target" closed-loop system behavior that is realizable. Note also that the performance index of Eq. 2 can be generalized to include a weight factor (in front of the input penalty term) to quantify the relative importance of the two main terms: the first one that aims at minimizing the closed-loop output mismatch associated with the nonlinear vs. the PID controller, and the second input penalty term. Furthermore, the inclusion of a weight factor would ensure consistency of units for the two terms in the performance functional.

Remark 2. It should be noted that the controller actually implemented in the closed-loop system is the PID controller with fixed parameter values and that it may not always be possible for the PID controller to exactly match the closed-loop behavior under the nonlinear controller. The purpose behind introducing the first-level tuning is twofold: (1) to ensure that important objectives (such as closed-loop stability) are not sacrificed in the (possibly unsuccessful) quest for a nonlinear controller–like behavior (which is accomplished through the optimization), and (2) to provide a rational way of constructing a range of the tuning parameter values over which the optimization is performed. With respect to the first objective, we note that the essence of the second-level optimization is to try to find the best tuning parameter values that make the PID controller emulate the behavior of the nonlinear controller. However, this objective should not come at the expense of more overriding objectives such as closed-loop stability. In particular, if the optimization problem were to be carried out without imposing any constraints on the parameter values, the solution may indeed lead to a closer match, but no longer guarantee that the PID controller enforces closed-loop stability when implemented. This is one reason why the proposed method includes the first level whose purpose, in part, is to make use of existing methods for PID controller tuning to first determine the range of tuning parameters for which closed-loop stability of the linearized process model under the PID controller is guaranteed. For such a range of values, the PID controller will enforce local closed-loop stability when implemented on the nonlinear process.

Remark 3. Having obtained the stabilizing parameter range and incorporated it as a constraint on the optimization, the search for the "optimal" gain then can take place only over this range. However, the stabilizing range may be too large to search over and the designer may wish to limit this range further by incorporating additional performance and robustness considerations. The use of first-level methods (which are based on the use of linear models) provides a rational—although not necessarily unique—way of constructing an appropriate subrange to work with. For example, if certain robustness margins can be obtained (and quantified explicitly as ranges on the tuning parameters) through the use of existing methods based on the linearized model, these margins can be incorporated as constraints that further limit the search range. This of course does not (nor is intended to) guarantee that the PID controller will exhibit such robustness or performance when implemented on the nonlinear process because the margins are based on the linearized model. However, at a minimum, this is a meaningful way to go about constructing or, more precisely, narrowing down the range over which the optimization is done (by requesting that the performance of the linearized model be consistent with what existing methods based on linear models yield). Ultimately, it is the "closeness" of the PID controller to the nonlinear controller resulting from the second-level optimization (not the first-level methods) that is essentially responsible for the performance properties exhibited by the PID controller when implemented on the nonlinear process. Given that different (first-level) tuning methods lead to different performance properties and yield different parameter values, the designer can examine the values obtained from different methods to form a reasonable idea about what an acceptable range might be around these nominal values and then construct such a range (through choosing appropriate α_i values) and implement it as constraints on the optimization (see the simulation study for an example). If the parameter values obtained (after performing the optimization) do not yield satisfactory performance (tested through simulations), then the parameter range could be expanded further (but still within the stabilizing range determined initially) in an iterative procedure.

Remark 4. The α_i values in Eq. 3 are introduced into the optimization as design parameters that allow the designer flexibility in tightening or relaxing the range of parameter values over which the optimization is carried out. If a given method yields satisfactory performance, and it is desired that the tuning parameters not be changed appreciably, this can be enforced by using values of the design parameters (α_i) close to 1. The tuning parameters resulting from the solution to the optimization problem in this case, although changed to mimic the nonlinear control action, will be close to those obtained from the classical tuning method considered in the first level. If, on the other hand, it is decided that some further improvement is warranted, then, at a minimum, the α_i values should be chosen to reflect the range (or a subset of the range) within which the parameter values can be changed by the optimization without losing stability. If the designer seeks to constrain the search over a smaller range (using certain performance or robustness margins such as those obtained from the first-level methods for the linearized closed-loop system), then the α_i values can be modified to reflect the new range. In general, the choice of α_i varies depending on the particular process under consideration and on the particular tuning methods that are being considered in the first level.

Remark 5. Regarding the performance properties of the proposed method with respect to those of the first-level methods, we first note that the first-level tuning guidelines are derived on the basis of the linearized process model, and therefore the robustness and performance properties obtained when using these methods to tune the PID controller are not guaranteed to carry over when the PID controller is implemented on the nonlinear system. So even if retuning of the first-level methods may bring about further performance improvement in the linear case (by possibly sacrificing stability and robustness margins), this does not imply that a similar improvement should be expected in the nonlinear setting. In

Table 1. Process Parameters and Steady-State Values

Parameter	Value	Unit
V	0.1	m^3
E/R	8000	K
C_{A0}	1.0	$kmol/m^3$
T_{A0}	400.0	K
ΔH	2.0×10^5	kJ/kmol
k_0	7.85×10^6	s^{-1}
c_p	1.0	$kJ\ kg^{-1}\ K^{-1}$
ρ	1000.0	kg/m^3
UA	1.667×10^3	$kJ\ s^{-1}\ K^{-1}$
F	0.001667	m^3/s
C_A^s	0.52	$kmol/m^3$
T_R^s	398.97	K
T_j^{nom}	493.87	K

general, there is no systematic way in which such methods can be retuned (if at all possible) to improve the performance in the nonlinear case. By contrast, the proposed method aims to improve the performance of the PID controller in the nonlinear setting by explicitly accounting for process nonlinearity through the optimization (an objective not shared by the first-level approaches). Whether this necessarily means that the resulting parameters will always yield performance that is "better" than what a given tuning method might yield is difficult to judge, but, more important, is not a point that the method is intended to address. The point is that the proposed approach is a meaningful way of tuning PID controllers that can yield good performance when implemented on the nonlinear system. From this perspective, the first-level methods serve as a rational starting point for the construction of the search ranges as discussed in *Remark 2*.

Remark 6. The optimization problem of Eqs. 2 and 3 is solved off-line as part of the design procedure to compute the optimal values of the tuning parameters. Also, the above optimization problem can be carried out over a range of initial conditions and set-point changes that are locally representative of the process operation to obtain PID tuning parameters that allow the PID controller to approximate, in an average (with respect to initial conditions and set-point changes) sense, the closed-loop response under the nonlinear controller. If the process is required to operate at an equilibrium point that is very far from the operating point for which the parameters are tuned, then it is best to perform the optimization again around the new, desired operating point to yield new tuning parameter values (as is also done in classical tuning). Regarding the optimization complexity issue, we note that possible complexity of the proposed optimization is mainly a function of the model complexity (such as nonlinearity, model order, and so on). The increase in computational demand expected in the case of higher-order and highly nonlinear systems is primarily attributed to the need to solve a higher-order system of nonlinear differential equations. However, with current computational capabilities, this does not pose unduly significant limitations on the practical implementation prospects of the proposed method, especially when compared with the computational complexity encountered in typical nonlinear optimization problems [such as nonlinear model predictive control (MPC)]. Furthermore, the approximations discussed in *Remark 9* below provide possible means that can help manage potential complexities even further. Finally, we note that, because the method involves a form of nonlinear optimization, it is expected that, in general, multiple optimal solutions may exist.

Remark 7. The basic idea behind the proposed PID controller tuning methodology, that is, that of tuning the PID controller to emulate some other well-designed controller that effectively handles complex dynamics, can be used to develop conceptually similar tuning methods for PID control of processes with other sources of complexities (besides nonlinearity) such as uncertainty, time delays, and manipulated input constraints. The logic behind such extensions is based on the following intuitive parallel: just as a nonlinear controller is a meaningful guide to be emulated by a PID controller being implemented on a nonlinear process, a controller that effectively handles constraints, uncertainty, and/or time delays can also be a meaningful guide to be emulated by a PID controller that is being implemented on a process with these characteristics. In principle, the extensions can be realized by adequately accounting for the complex characteristics of these processes within both levels of the tuning method. For example, for systems with uncertainty, classical tuning methods that provide sufficient robustness margins can be used to come up with the first-level parameter values. Then a robust nonlinear controller (see, for example, El-Farra and Christofides[10]) can be designed and the closed-loop profiles, obtained under the robust nonlinear controller for a sufficient number of realizations of the uncertainty (which may be simulated, for instance, using random number generators), may be computed. Finally, the parameter values obtained from the first-level tuning method may be improved upon by solving an optimization problem that minimizes the error over the profiles in the representative set. In a conceptually similar fashion, for systems with constraints, an anti-windup scheme could be used initially to obtain the first-level parameter values. A nonlinear controller design that handles input constraints can then be chosen for the second-level optimization. The PID controller tuning guidelines can then serve to "carry over" the constraint handling properties of this nonlinear controller and improve upon the first-level tuning methods in two ways: (1) through the objective function, by requiring the control action and closed-loop process response under PID control to mimic that under the constrained nonlinear controller; and (2) through the incorporation of the input constraints directly into the optimization problem. However, it should be noted that, although such extensions are intuitively appealing, a detailed assessment and characterization of their potential requires further investigation.

Remark 8. Note that the derivative part of the PID controller is often implemented using a filter. This feature can be easily incorporated in the optimization problem by explicitly accounting for the filter dynamics. Constraints on the filter time constant τ_f, obtained empirically through knowledge of the nature of noise in the process, can be imposed to ensure that the filtering action restricts the process noise from being transmitted to the control action.

Remark 9. To allow for simple computations, approxima-

Table 2. PI Tuning Parameters

Tuning Method	K_c	τ_I
IMC	1.81	0.403
Ziegler–Nichols	2.86	4.16
Proposed method	5.56	0.297

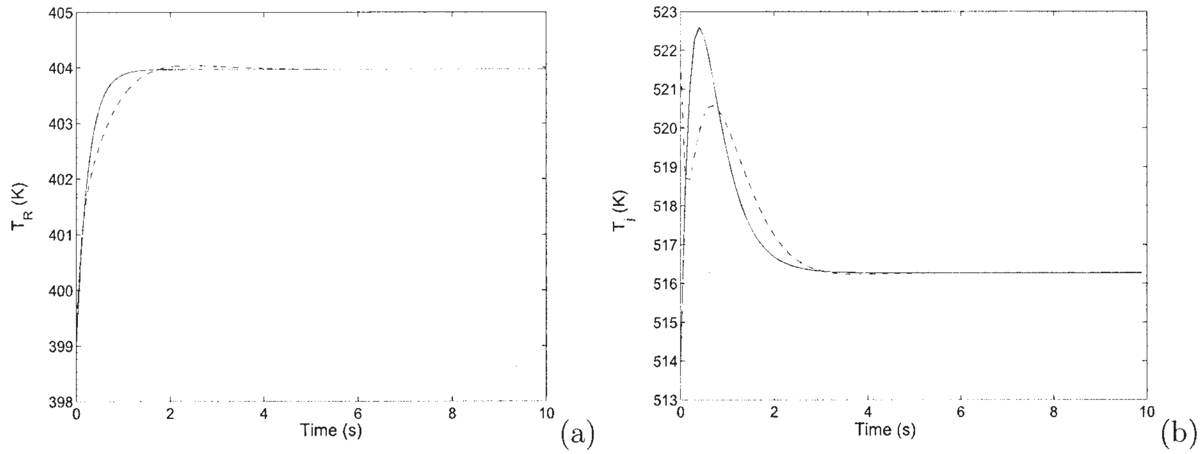

Figure 2. Closed-loop output (a) and manipulated input (b) profile under a linearizing controller (solid line) and a PI controller tuned using the proposed method (dashed line).

tions can be introduced in solving the optimization problem of Eqs. 2 and 3. For instance, in the computation of the control action, the error $e(t)$ may be approximated by simply taking the difference between the set point y_{sp} and the process output under the nonlinear controller $y_{nl}(t)$, leading to a simpler optimization problem that can be easily solved using numerical solvers such as Microsoft Excel (for a given choice of the decision variables, the objective function can be computed algebraically and does not involve integrating the process dynamics). The justification behind this is that, if the resulting value of $u_{PID}(t)$ is "close enough" to $u_{nl}(t)$, then this approximation holds (see the simulation example for a demonstration). If the solution of the optimization problem does not yield a sufficiently small value for the objective function (indicating that u_{PID}, and thus y_{PID}, is significantly different from u_{nl}, and y_{nl}), this approximation may no longer be valid. In this case, one could revert to using $e(t) = y_{sp} - y_{PID}(t)$ in the optimization problem, where y_{PID} is the closed-loop process response under the PID controller. Note also that, in some cases (particularly for low-dimensional systems with real analytic vector fields), the value of the performance index may be calculated explicitly as an algebraic function of the controller parameters (leading to a static finite-dimensional optimization problem) by solving Zubov's partial differential equation using techniques similar to those presented in Kazantzis et al.[20] Those techniques can also be used in designing an optimally tuned nonlinear controller that serves as a meaningful target to be emulated by the PID controller.

Remark 10. Finally, we note that the proposed method does not turn the PID controller into a nonlinear controller. The tuning method can only serve to improve upon the process response of the PID controller for operating conditions for which PID control action can be used to stabilize the process. If the process is highly nonlinear, or a complex process response is desired, it may be possible that the PID controller structure is not adequate and, in this case, the appropriate nonlinear controller should be implemented in the closed loop to achieve the desired closed-loop properties.

Application to a Chemical Reactor Example

We consider a continuous stirred-tank reactor where an irreversible, first-order reaction of the form $A \xrightarrow{k} B$ takes place. The inlet stream consists of pure species A at flow rate F, concentration C_{A0}, and temperature T_{A0}. Under standard modeling assumptions, the mathematical model for the process takes the form

$$\dot{C}_A = \frac{F}{V}(C_{A0} - C_A) - k_0 e^{-E/RT_R} C_A$$

$$\dot{T}_R = \frac{F}{V}(T_{A0} - T_R) + \frac{(-\Delta H)}{\rho c_p} k_0 e^{-E/RT_R} C_A + \frac{UA}{\rho c_p V}(T_j - T_R) \tag{4}$$

where C_A denotes the concentration of the species A; T_R denotes the temperature of the reactor; T_j is the temperature of the fluid in the surrounding jacket; U is the heat-transfer coefficient; A is the jacket area; V is the volume of the reactor; k_0, E, and ΔH are the preexponential constant, the activation energy, and the enthalpy of the reaction, respectively; and c_p and ρ are the heat capacity and fluid density in the reactor, respectively. The values of all process parameters are given in Table 1. At the nominal operating condition of $T_j^{nom} = 493.87$ K, the reactor is operating at the unique, stable steady state (C_A^s, T_R^s) = (0.52 kmol/m^3, 398.97 K). The control objective is to implement set-point changes in the reactor temperature using the jacket fluid temperature T_j as the manipulated input, using a P, PI, or PID controller.

To proceed with our controller tuning method, we initially design an input/output linearizing nonlinear controller. Note that the linearizing controller design is used in the simulation example only for the purpose of illustration, and any other nonlinear controller design deemed fit for the problem at hand can be used as part of the proposed controller tuning method.

Defining $x = [C_A - C_A^s, T_R - T_R^s]'$ and $u = T_j - T_j^{nom}$, the

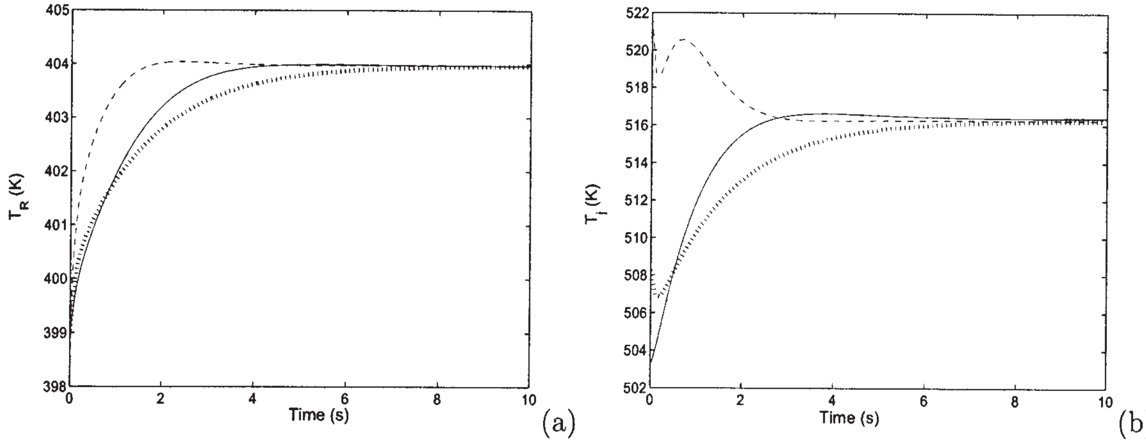

Figure 3. Closed-loop output (a) and manipulated input (b) profile using IMC tuning rules for PI controller (solid line), using Ziegler–Nichols tuning rules (dotted line), and the proposed method (dashed line).

process of Eq. 4 can be recast in the form of Eq. 1, where the explicit form of $f(\cdot)$ and $g(\cdot)$ are omitted for brevity. Consider the control law

$$u = \frac{\nu - y(t) - \gamma L_f h(x)}{\gamma L_g h(x)} \tag{5}$$

where $L_f h(x)$ and $L_g h(x)$ are the Lie derivatives of the function $h(x)$ with respect to the vector functions $f(x)$ and $g(x)$, respectively; γ, a positive real number, is a design parameter; and ν is the set point. Taking the derivative of the output in Eq. 1 with respect to time, we get $\dot{y} = L_f h(x) + L_g h(x)u$. Substituting the linearizing control law of Eq. 5, we get $\dot{y} = (\nu - y)/\gamma$.

Under the control law of Eq. 5, the controlled output y evolves linearly, to achieve the prescribed value of ν, and the design parameter γ is the time constant of the closed-loop response.

It is well known that when a first-order closed-loop response, with a given time constant, is requested for a linear first-order process, the method of direct synthesis yields a PI controller. Note that the relative order of the controller output T_R, with respect to the manipulated input T_j, in the example of Eq. 4 is also one. Even though the nonlinear controller is a static controller, and the PI controller is dynamic, both controllers are capable of generating closed-loop behaviors that are of the same kind (a linear first-order response with a prescribed closed-loop time constant). This encourages the use of a PI controller and tuning its parameters to achieve the prescribed first-order response.

For the purpose of tuning the PI controller, the nonlinear process response generated under the nonlinear controller, using a value of $\gamma = 0.25$, was used in the optimization problem. An appropriate range for the tuning parameters was derived from the K_c and τ_I suggested by the IMC-based and Ziegler–Nichols tuning rules (where the parameters $K_{cu} = 6.36$ and $P_u = 5.0$ are obtained using the method of relay auto tuning).[21] In particular, the constraints on the values of the parameters were chosen as follows: for a given parameter, the largest and the smallest values prescribed by the available tuning methods (in this case the IMC-based and Ziegler–Nichols) were chosen and the upper bound on the parameters was chosen as twice the maximum value, and the lower bound was chosen as half the minimum value: $0.9 \le K_c \le 5.7$ and $0.2 \le \tau_I \le 8.2$. The values of the parameters, computed using the IMC method, Ziegler–Nichols, and the two-level PI tuning method are reported in Table 2.

The solid lines in Figures 2a and 2b show the closed-loop response of the output and the manipulated input under the nonlinear control of Eq. 5. Note that the value of γ was chosen as 0.25 to yield a smooth, fast transition to the desired set point. The optimization problem was solved approximately, using the closed-loop process response under the nonlinear controller to compute $e(t)$, and the objective function included only penalties on the difference between the control actions under the PI controller and the nonlinear controller (see *Remark 9*). The dashed line shows the response of the PI controller tuned using the proposed optimization-based method. The result shows that the response under the PI controller is close to that under the nonlinear controller and demonstrates the feasibility of using a PI controller to generate a closed-loop response that mimics the response of the nonlinear controller.

In Figure 3a, we present the closed-loop responses when the controller parameters computed using the IMC-based tuning rules and Ziegler–Nichols are implemented. As can be seen, the transition to the new set point under the PID controller tuned using the proposed method (dashed lines) is fastest when compared to a classical PI controller tuned using IMC tuning rules (solid line) and Ziegler–Nichols tuning rules (dotted line). The corresponding manipulated input profiles are shown in Figure 3b.

We now demonstrate the application of the proposed method to the same system, but with C_A as the controlled variable and T_j as the manipulated variable. As in the previous case, we initially design an input/output linearizing nonlinear controller

Table 3. PID Tuning Parameters

Tuning Method	K_c	τ_I	τ_D
IMC-I	−678.8	0.95	0.0524
IMC-II	−1208.9	1.00	0.114
Ziegler–Nichols	−2072.0	0.149	0.028
Proposed method	−951.21	0.978	0.114

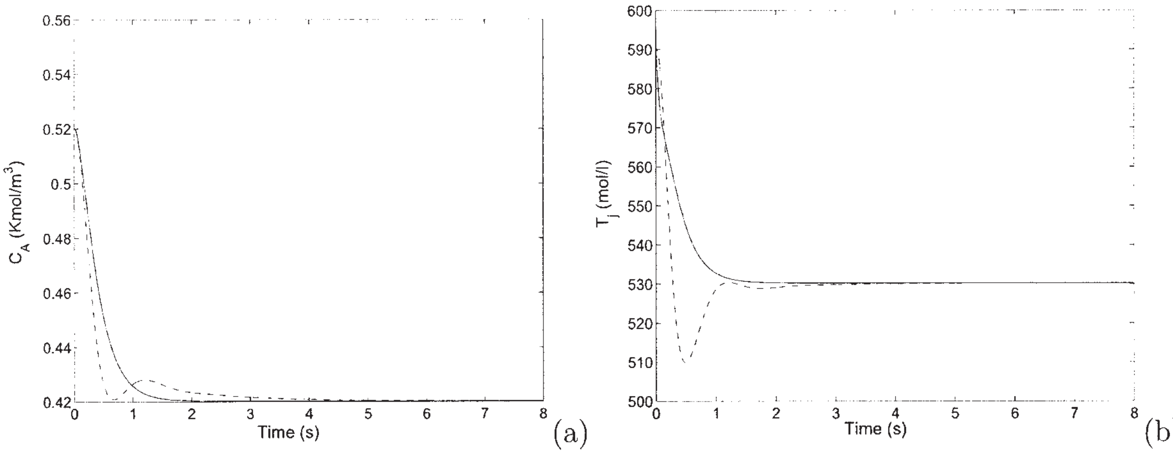

Figure 4. Closed-loop output (a) and manipulated input (b) profile under a linearizing controller (solid line) and a PID controller tuned using the proposed method (dashed line).

to yield a second-order linear input–output response in the closed-loop system of the form $\tau_{cl}^2\ddot{y} + (2\xi/\tau_{cl})\dot{y} + y = \nu$, where τ_{cl} and ξ are design parameters and were chosen as $\tau_{cl} = 0.2$ and $\xi = 1.05$ (implying that the closed-loop system is a slightly overdamped second-order system). The following tuning methods were used for the first level: (1) IMC-based tuning rule, where a step test is run to approximate the system by a first-order + time-delay process, hereafter referred to as IMC-I; (2) IMC-based tuning rule, where the process is linearized around the operating steady state to obtain a second-order linear model, hereafter referred to as IMC-II; and (3) Ziegler–Nichols tuning rules, where the parameters $K_{cu} = -34{,}543$ and $P_u = 0.223$ are obtained using the method of relay autotuning[21] (the tuning parameter values are reported in Table 3). Based on the parameter ranges suggested by the first-level tuning methods, the following constraints were used in the optimization problem set up to compute K_c, τ_I, and τ_D: $-2072.0 \leq K_c \leq -678$, $0.149 \leq \tau_I \leq 1$, and $0.0114 \leq \tau_D \leq 0.052$. The derivative part of the controller was implemented using a first-order filter with time constant $\tau_f = 0.1$.

The solid lines in Figures 4a and 4b show the closed-loop response of the output and the manipulated input under the linearizing control design. The dashed-line shows the response of the PID controller tuned using the proposed method, which is close to the response of the nonlinear controller. As is clear from Figure 4, the resulting PID controller yields a response that is sufficiently close to that of the nonlinear controller. In Figure 5a, we present the closed-loop responses when the controller parameters computed using the classical tuning rules are implemented. The values suggested by Ziegler–Nichols tuning lead to closed-loop instability. In the simulation, a smaller value for $K_c = -518.4$ and a larger $\tau_I = 0.14$ were used. As can be seen, the transition to the new set point using the proposed tuning method (dashed lines in Figure 5) compares favorably to that obtained when using the IMC-based tuning rules I and II (dotted and solid lines, respectively) and the Ziegler–Nichols tuning rules (dash–dotted line). The corresponding manipulated input profiles are shown in Figure 5b.

In summary, the proposed tuning method leads to PID controller parameters that achieve a closed-loop response that is sufficiently close to the one obtained under the nonlinear controller, and compares favorably with the responses obtained using classical tuning techniques.

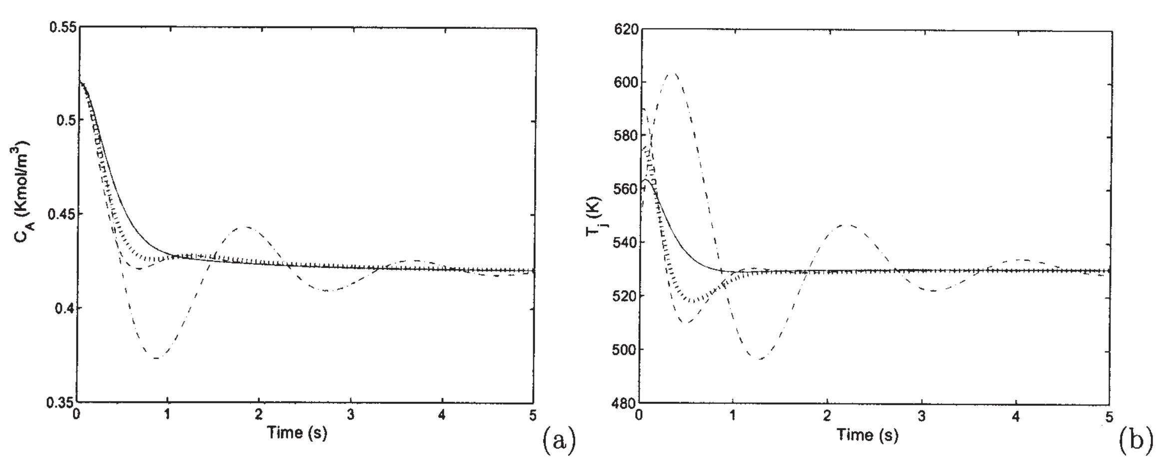

Figure 5. Closed-loop output (a) and manipulated input (b) profile using IMC tuning rules I (dotted line), IMC tuning rules II (solid line), Ziegler–Nichols tuning rules (dash–dotted line), and the proposed method (dashed line).

Acknowledgments

Financial support from the National Science Foundation (Grant CTS-0129571) is gratefully acknowledged.

Literature Cited

1. Ziegler JG, Nichols NB. Optimum settings for automatic controllers. *Trans ASME.* 1942;64:759-768.
2. Rivera DE, Morari M, Skogestad S. Internal model control. 4. PID controller design. *Ind Eng Chem Process Des Dev.* 1986;25:252-265.
3. Tyreus BD, Luyben WL. Tuning PI Controllers For Integrator Dead Time Processes. *Ind Eng Chem Res.* 1992;31:2625-2628.
4. Lee JT, Cho W, Edgar TF. Control system design based on a first order plus time delay model. *J Process Contr.* 1997;7:65-73.
5. Panagopoulos H, Astrom KJ, Hagglund T. Design of PID controllers based on contrained optimization. *IEE Proc.-Control Theory Appl.* 2002;149:32-40.
6. Skogestad S. Simple analytic rules for model reduction and PID controller tuning. *J Process Contr.* 2003;13:291-309.
7. Astrom KJ, Hagglund T. Revisiting the Ziegler–Nichols step response method for PID control. *J Process Contr.* 2004;14:635-650.
8. Astrom KJ, Hagglund T, Hang CC, Ho WK. Automatic tuning and adaption for PID controllers—A survey. *Contr Eng Pract.* 1993;1:699-714.
9. Cominos P, Munro N. PID controllers: Recent tuning methods and design to specification. *IEE Proc.-Control Theory Appl.* 2002;149:46-53.
10. El-Farra NH, Christofides PD. Integrating robustness, optimality, and constraints in control of nonlinear processes. *Chem Eng Sci.* 2001;56:1841-1868.
11. El-Farra NH, Christofides PD. Bounded robust control of constrained multivariable nonlinear processes. *Chem Eng Sci.* 2003;58:3025-3047.
12. Lin Y, Sontag ED. A universal formula for stabilization with bounded controls. *Syst Contr Lett.* 1991;16:393-397.
13. Antoniades C, Christofides PD. Feedback control of nonlinear differential difference equation systems. *Chem Eng Sci.* 1999;54:5677-5709.
14. Sepulchre R, Jankovic M, Kokotovic P. *Constructive Nonlinear Control*. Berlin/Heidelberg, Germany: Springer-Verlag; 1997.
15. Khalil HK. *Nonlinear Systems*. 2nd Edition. New York, NY: Macmillan; 1996.
16. Isidori A. *Nonlinear Control Systems: An Introduction*. 2nd Edition. Berlin/Heidelberg, Germany: Springer-Verlag; 1989.
17. Wright RA, Kravaris C, Kazantzis N. Model-based synthesis of nonlinear PI and PID controllers. *AIChE J.* 2001;47:1805-1818.
18. Benaskeur AR, Desbiens A. Backstepping-based adaptive PID control. *IEE Proc.-Control Theory Appl.* 2002;149:54-59.
19. Chang W, Hwang R, Hsieh J. A self-tuning PID control for a class of nonlinear systems based on the Lyapunov approach. *J Process Contr.* 2002;12:233-242.
20. Kazantzis N, Kravaris C, Tseronis C, Wright RA. Optimal controller tuning for nonlinear processes. *Automatica.* 2005;41:79-86.
21. Astrom KJ, Hagglund T. Automatic tuning of simple regulators with specification on the gain and phase margins. *Automatica.* 1984;20:645-651.

Manuscript received Apr. 26, 2004, and revision received Mar. 30, 2005.