ARTICLE IN PRESS
ISA Transactions xxx (xxxx) xxx
Contents lists available at ScienceDirect
ISA Transactions
journal homepage: www.elsevier.com/locate/isatrans

Research article

A new method for experimental tuning of PI controllers based on the step response

Roberto Sanchis *, Ignacio Peñarrocha-Alós
Department of Industrial Systems Engineering and Design, Universitat Jaume I, Spain

ARTICLE INFO

Article history:
Received 12 June 2020
Received in revised form 13 September 2021
Accepted 14 September 2021
Available online xxxx

Keywords:
PID
PI
Auto-tuning
Step response

ABSTRACT

In this paper we present a new method for tuning Proportional Integral (PI) controllers from experimental data obtained through an open loop step test over the process to be controlled. The tuning procedure requires first the measurement of the process gain, and the times taken to reach the 5%, 35.3% and 85.3% of the final output and then applying a set of tuning equations. The tuning equations approximate the controller that minimizes the Integral of Absolute Error (IAE) of the disturbance response for a model with three real poles and time delay and are very accurate for a wide range of non oscillatory stable systems. The user can select the desired robustness (through the required maximum of the Sensitivity function (M_s)), as a difference with usual methods that allow only to choose among two or three predefined robustness. The PI controller that minimizes the disturbance IAE is defined by default, but the user can also select a detuning factor to define slower controllers with the same robustness, allowing to find the desired compromise between performance and actuator activity due to sensor measurement noise. An application for Android, that can be downloaded for free, and a web based application, have been developed to implement the tuning procedure.

1. Introduction

Proportional, Integral and Derivative (PID) controllers are the most widely used control algorithms in industry [1]. There are two main types of approaches for tuning the PID parameters. The first type is the model based approach, where a model of the plant is needed to perform accurate calculations of the controller parameters, but difficult to apply in real plants, due to the need of an accurate process model identification or the need of computationally cost and complex optimization procedures. The second type is the experimental tuning approach that is less accurate, but much easier to apply, since the user only needs to perform a simple experiment in the plant, take some measurements in the response, and apply some tuning equations with a very low computing cost. Experimental tuning methods can be classified in open loop (OL) and closed loop (CL) ones. The OL methods are based on the use of simple models of the plant, generally obtained from the OL response to a step input. On the other hand, the CL methods are based on applying a relay feedback to obtain one point of the frequency response (commonly where the phase angle is $-\pi$). Thanks to the use of a PID controller one finally has a CL behaviour with a given time response performance, actuator activity due to sensor measurement noise, and a given robustness. Usually, the performance is quantified in terms of the response under step disturbances, with several metrics as IAE (integral of the absolute error), ITAE (integral of the time multiplied by the absolute error) or ISE (integral of the squared error). The robustness can be quantified through the maximum of the frequency response of the CL sensitivity function (M_s) or the OL phase and gain margin. For both model-based and experimental PID design methods, one can encounter different design procedures depending on the guarantee of fulfilment of robustness, performance or noise amplification requirements.

The design of PI controllers to minimize the effect of load disturbances, using a model of the process, has been widely addressed in the literature. Some works try to maximize the integral gain, like [2,3], where a direct numerical optimization is proposed, constrained to a given value of the maximum of the sensitivity function (M_s). In [4], the authors presented a procedure for tuning PI and PID controllers that maximize the integral gain, while fulfilling an exact phase margin, and a lower bound in the gain margin. In the case of experimental PID tuning, most of the works propose to fulfil some robustness conditions and some requirements in the closed-loop bandwidth (as for example [5–7]). There are less papers that deal with the optimization of disturbance response, with limited results. Maybe this is because the calculation of PID parameters involves a computationally complex multi-parametric optimization problem. In recent times, the development of metaheuristic optimization methods has led to the proposal of several PI and PID model based tuning strategies

* Corresponding author.
E-mail address: rsanchis@uji.es (R. Sanchis).

https://doi.org/10.1016/j.isatra.2021.09.008

Please cite this article as: R. Sanchis and I. Peñarrocha-Alós, A new method for experimental tuning of PI controllers based on the step response. ISA Transactions (2021), https://doi.org/10.1016/j.isatra.2021.09.008.

that are based on the solution of different constrained optimization problems. Those optimization problems are non linear and non convex, and methods based on evolutive algorithms, particle swarm optimization, artificial bee colony, grey wolf and other, have been applied successfully to the PID model based tuning problem. In [8], a survey of the application of evolutive algorithms to PID tuning is presented. In several recent works, as [9–11] or [12], some metaheuristic methods, as particle swarm or bee colony optimization are used to find the optimal parameters of PID controllers.

In [13] and [14] Internal Model Control techniques to tune PID controllers are revisited, proposing new PID tuning strategies. In both cases, a tuning parameter that defines the desired reference model can be selected by the user, leading to designs that range from slow and robust to fast and less robust. The tuning equations are based on a simple model of the system, First Order plus Time Delay (FOTD) in the case of PI controller, but the robustness cannot be selected explicitly and independently of the speed response. The resulting robustness depends on the tuning parameter, but cannot be predicted in advance. Furthermore, the disturbance response can be indirectly addressed through the selection of the tuning parameter, but is not optimized (using any index as IAE, ITAE or ISE for example).

In the development of an experimental tuning procedure the final goal is to obtain simple equations that allow the user to calculate the controller parameters as a function of some measurements taken from the experimental response data, as a difference with model based tuning methods. One approach to obtain those tuning equations consists of using a simple intermediate fixed structure model, whose parameters are calculated from the measurements using simple functions, and performing some model based tuning procedure to obtain the controller parameters. Finally, some equations must be found that approximate the obtained controller parameters as a function of the original measurements taken from the experiment. Then, the use of the proposed controller parameters on a real plant will lead to a given robustness or performance depending on the ability of the proposed fixed model structure to capture the real dynamics of the controlled plant. Some advanced experimental procedures allow the user to select also some simple tuning parameters to achieve a given robustness. The most widely used plant model for experimental tuning procedures is the FOTD model.

The work [15] summarizes a compendium of more than 600 PI and PID tuning rules including the ones based in OL step response experiments, but they do not allow to freely choose the robustness, or to select the desired compromise between performance and high frequency noise amplification. Astrom and Hagglund presented in [16] an algorithm for tuning PI and PID controllers based on relay experiments that maximizes the integral gain, subject to a robustness constraint in the phase and gain margins. In [17] a relay feedback auto-tuning algorithm for PID is presented. It is based on finding two or three points of the frequency response, and using the resulting straight line approximation of the Bode diagram to design a PID that maximizes the integral gain subject to a robustness constraint in the phase and gain margins. The use of the straight line approximation leads to a much more accurate result than the methods based on using one single point. In [18], an integrator with relay feedback autotuning algorithm is presented based on the use of sampling filters. The idea is to obtain two points of the frequency response of the process, and then to estimate the dominant time constant and the static gain to define the desired closed loop transfer function. The procedure allows the user to select the desired closed loop time constant, but it does not take into account the robustness nor the disturbance response explicitly.

The work [19] describes a method based on the approximation of the process behaviour by a fractional order plus time delay model. Closed loop experiments are proposed to first obtain the frequency response of the system. Then a reduced model of fractional order plus time delay is obtained from the frequency response. For that type of simple process model, tuning formulas are derived that approximate the parameters of the PID controllers that minimize the ITAE (either for reference or for disturbance response). The results are good, but the calculations needed to obtain the parameters of the fractional order plus time delay model are very complex, including the solution of an optimization problem. Furthermore, the tuning procedure does not take into account the robustness, or the compromise between performance and noise amplification.

With respect the experimental OL methods, in [20] a tuning method is proposed for PI controllers using a FOTD approximated model and by maximizing the integral gain while fulfilling a robustness constraint defined by a value $M_s = 1.4$. The author demonstrates that it is not possible to find an accurate tuning rule for a wide range of systems, based only on the FOTD approximated parameters, and it proposes a conservative tuning rule. However, no rules are provided to detune the PI controller if a slower response with a lower noise amplification is desired. In [21], the previous work for PI controllers is extended to PID controllers, leading to a set of conservative tuning rules derived for different robustness constraints.

In this work an experimental OL tuning rule is proposed that uses a slightly more complex underlying model with the addition of fastest dynamics that can be easily deduced from the measurements of the experimental response data with just one more measurement than the traditional methods found in the literature (based on FOTD model). The controller for that model structure is also designed as a function of the desired robustness, disturbance rejection performance and actuator activity limitations due to measurement noise. With that design, the expressions that translate from experimental data and desired behaviour, to the controller parameters, have been obtained. These tuning rules have been applied to different process models of different orders and lead/lag dominance, showing a better performance than the aforementioned methods based on an underlying FOTD model due to a wider ability to capture dynamics for higher-order systems. The main novelties of the proposed approach, that cannot be found in previous works about experimental PI tuning, can be summarized as

- The tuning equations are based on simple measurements taken from the open loop step response, that are the dc gain and the times taken to reach the 5%, 35.3% and 85.3% of the final value. The addition of the measurement of the time to reach the 5% of the output (w.r.t. FOTD methods) and the use of a third order model leads to very accurate tuning equations for a wide range of processes, from lag dominant to delay dominant.
- The tuning procedure allows to first select the robustness in terms of M_s in a continuous range and a default controller that minimizes the IAE is proposed.
- The required compromise between performance and noise amplification can be selected to obtain a detuned PI controller keeping the robustness, defining slower controllers with a lower noise amplification.
- The tuning rules that translate from experimental data and desired behaviour to controller parameters have been implemented in a free available software (in mobile app and web based format) to ease obtaining the controller parameters. The software shows also a simulation of the controlled plant for a plant defined by the process model structure used in the controller design just to show an approximation of the response that the controller will drive on the real controlled process.

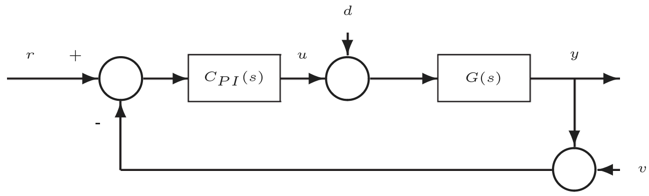

Fig. 1. SISO PI control loop.

The structure of the paper is as follows. In Section 2, the problem is stated. Section 3 describes in detail the PI tuning procedure, including the tuning equations. In Section 4, the android application that implements the tuning equations is presented. In Section 5, the procedure is tested on a batch of processes to illustrate the validity of the approach for different process dynamics, while several simulation examples are developed in Section 6 to illustrate how to find the desired compromise between performance and noise amplification. Finally, in Section 7, a real experimental case is developed to show the applicability and accuracy of the approach in a real system with measurement noise.

2. Problem statement

In this paper, the SISO PI control loop shown in Fig. 1 is considered. The signal r represents the set point to be tracked, the signal u is the input of the process, y is the output to be controlled, v is the measurement noise and d is a disturbance.

The structure of the PI controller is assumed to be

$$C_{PI}(s) = K_p \left(1 + \frac{1}{T_i s}\right) \tag{1}$$

the model of the process $G(s)$ is assumed to be unknown.

The objective of the experimental PI tuning procedure is to define the parameters of the controller to fulfil the following objectives:

- Achieve a required robustness defined as an approximate value of the maximum of the sensitivity function, M_s.
- Find the desired compromise between performance (defined as the IAE in disturbance rejection) and measurement noise amplification.

The usual PI experimental tuning methods, either based on an open loop step response or on a closed loop relay, propose a table with two PI controllers to choose from, with different robustness (high and low). Those tables offer very limited possibilities for tuning the PI, especially if the user wants to find a compromise between performance and noise amplification, or if an intermediate robustness is desired.

The main control parameter to be selected by the user with the proposed approach is the robustness in terms of the required M_s. The lower the value of M_s, the more robust and less oscillatory behaviour, meaning that the controller can cope better with modelling errors or process changes. But as a counterpart, the performance will be worse (higher values of disturbance IAE). On the contrary, high values of M_s lead to a better performance (lower IAE), but the behaviour is more oscillatory, and the controller cannot cope as well with modelling errors or process changes. Hence, the selection of the desired value of M_s depends on the process uncertainty and future process changes, and also on the needed performance of the loop. Once the robustness (M_s) has been selected, the tuning method gives by default the PI that minimizes the disturbance IAE, but allows to detune this controller to define a slower control that reduces the measurement noise amplification, with the same robustness. This is performed with the second tuning parameter to be selected by the user (the detuning factor). A value of 1 in the detuning factor leads to the minimum IAE controller. The lower the detuning factor, the slower performance and the lower noise amplification. The detuning factor allows to find the required compromise between performance (IAE) and noise amplification. The required compromise depends mainly on the noise level of the sensor, and the nature of the actuator, since the noise amplification results in actuator fluctuation. For example, if the actuator is a valve, the fluctuation due to noise amplification is very harmful, because produces wear in the moving parts. In that case, a low value of the detuning factor should be used to reduce the noise amplification, at the expense of a lower performance (higher IAE).

The proposed tuning procedure is based on the measurement of simple points in the step response of the process, and uses tuning equations that lead to controllers of any required intermediate robustness, and with the desired compromise between performance and noise amplification.

3. Controller tuning procedure

The proposed tuning method is based on the experimental data obtained from an open loop step change in the input signal, after a steady state has been reached. The experiment is assumed to be long enough to reach the steady state after the step change. Then, the static gain can be directly computed as $K = \frac{\Delta y(\infty)}{\Delta u}$.

The controller tuning procedure is based on taking three measurements: the times taken to reach the 5%, the 35.3% and the 85.3% of the final output value. With those measured times and the process gain, the user can apply some tuning equations to obtain the controller parameters. Those tuning equations, (2) and (3), are also a function of M_s, so, the user can freely select the desired robustness in terms of M_s. Choosing values around $M_s = 1.4$ lead to a robust control loop, that is not oscillatory, and can cope with modelling errors or process changes, but as a counterpart, the performance will be low (high values of disturbance IAE). On the contrary, values around $M_s = 2$ lead to a less robust control loop, that is more oscillatory, and cannot cope as well with modelling errors or process changes, but achieve higher performances (lower values of disturbance IAE). Hence, the selection of the desired value of M_s depends on the process uncertainty and future process changes, and also on the needed performance of the loop.

The tuning equations give the controller that minimize the Integral of Absolute Error (IAE) in the input disturbance rejection.

$$K_{p,opt} = \frac{1}{K} f_1(t_{5\%}, t_{35\%}, t_{85\%}, M_s) \tag{2}$$

$$T_{i,opt} = f_2(t_{5\%}, t_{35\%}, t_{85\%}, M_s) \tag{3}$$

The previous equations define the controller that minimizes the disturbance IAE for the required robustness. However, if the user wants to define a slower controller in order to reduce the high frequency noise amplification, a detuning equation is needed to define a slower controller while keeping the required robustness. If a detuning parameter $\gamma \in (0, 1]$ is defined, the detuning equations should be in the form:

$$K_p = \gamma K_{p,opt} \tag{4}$$

$$T_i = f_3(t_{5\%}, t_{35\%}, t_{85\%}, M_s, \gamma) T_{i,opt} \tag{5}$$

where $\gamma = 1$ results in the PI controller that minimizes the IAE, and $\gamma \to 0$ results in arbitrarily slow controllers with an arbitrarily low noise amplification.

Table 1
PI tuning table. FOTD method.

Controller	K_p	T_i
Low robustness PI	$\frac{0.859}{K}\left(\frac{L}{\tau}\right)^{-0.997}$	$1.484\tau\left(\frac{L}{\tau}\right)^{0.468}$
High robustness PI	$\frac{0.15}{K}+\left(0.35-\frac{\tau L}{(\tau+L)^2}\right)\frac{\tau}{KL}$	$0.35L+\frac{13\tau^2 L}{\tau^2+12\tau L+7L^2}$

3.1. Tuning equations fundamentals

In order to derive expressions for Eqs. (2), (3) and (5) that give accurate results for a wide range of plants, first, a model of the plant is proposed to be estimated from the measurements. The proposed model has three real poles and a time delay. This model has only one more parameter than the most commonly used in literature (the first order plus time delay, FOTD, (6)), and can be easily derived from it. The FOTD model is defined as

$$G(s)=\frac{K}{1+\tau s}e^{-Ls} \tag{6}$$

where L is the time delay, τ is the time constant and K is the static gain. For the simple FOTD model, several PI tuning formulas can be found in literature. Table 1 shows as an example a compendium of equations by Murrill (low robustness PI, [22]) and Astrom and Hagglund (high robustness PI, [20]) that try to define two controllers of different robustness.

No matter which tuning formulas are used, there are two drawbacks in these well known and widely used tuning methods. First, usually there are only two PI controllers of high and low robustness to choose from, and second, there is no way of finding a compromise between performance and noise amplification from the tuning table. Furthermore, despite the FOTD model is reasonable to approximate many non oscillating processes, one could use other types of models that are better approximators of the process behaviour. The fact is that with only 3 parameters it is not possible to find a tuning equation that is accurate for a wide range of plants. In this sense, there are two options: to find very conservative tuning equations that can be applied to a wide range of plants, but with a low performance (this is the approach of [20]), or to find more accurate tuning equations using a fourth parameter that can be measured easily (this is the approach of this paper).

Our proposed model is shown in (7). This model is a more general approximator of the behaviour of general over damped systems than (6) . In fact, model (6) is a particular case of model (7) (model (7) reduces to model (6) when $\alpha = 1$). This model can be obtained by first computing a FOTD model and then substituting the delay L by two real time constants of $\frac{1-\alpha}{2}L$ and a time delay of αL, where $\alpha \in [0, 1]$.

$$G(s)=\frac{K}{(1+\tau s)(1+\frac{(1-\alpha)L}{2}s)^2}e^{-\alpha Ls} \tag{7}$$

The gain can be directly computed as $K=\frac{\Delta y(\infty)}{\Delta u}$. To calculate L and τ, we propose the well known method based on the times taken to reach the 35.3% and the 85.3% of the final value:

$$L=1.3t_{35\%}-0.29t_{85\%} \tag{8}$$

$$\tau=0.67(t_{85\%}-t_{35\%}) \tag{9}$$

Once L and τ have been obtained, Eq. (10) shows our proposal to obtain α as a function of $t_{5\%}$, L and τ. The function has been obtained from a batch of plants with different values of $\frac{L}{\tau}$, and different lag–delay ratio, by approximating the step response of model (7) to those plants.

$$\alpha=0.598+0.4799\frac{t_{5\%}}{L}-\frac{0.41}{\left(\frac{t_{5\%}}{\tau}\right)^{0.6}} \tag{10}$$

For lag dominant processes, a low value of α results, while for delay dominant processes, a high value of α is obtained. Valid values of α range between 0 and 1. In fact, the tuning equations have been developed for $\alpha \in [0, 1]$. However, for very lag dominant plants, Eq. (10) can lead to $\alpha < 0$. In those cases, model (7) is nonsense, but the developed tuning equations can still be applied for $\alpha < 0$ through extrapolation.

Once the parameters L, τ and α of model (7) have been defined as a function of $t_{5\%}$, $t_{35\%}$ and $t_{85\%}$, the tuning equations can be expressed as:

$$K_{p,opt}=\frac{1}{K}f_1'(\frac{L}{\tau},M_s,\alpha) \tag{11}$$

$$T_{i,opt}=\tau f_2'(\frac{L}{\tau},M_s,\alpha) \tag{12}$$

$$T_i=f_3'(\frac{L}{\tau},M_s,\alpha,\gamma)T_{i,opt} \tag{13}$$

The following section describes in detail the development of the tuning equations.

3.2. Tuning equations

In this section the tuning equations are developed. The idea is to find the PI that minimizes the IAE for system (7), while fulfilling the required robustness, defined in terms of M_s. If this tuning problem is solved for a grid of values of M_s, $\frac{L}{\tau}$ and α, then the resulting values of K_p and T_i can be fitted to a function of M_s, $\frac{L}{\tau}$ and α.

As the fitting of a very nonlinear function of 3 parameters is very difficult, the proposal is to develop simpler tuning equations, as functions of M_s and $\frac{L}{\tau}$, for fixed values of α in the set $\{0, 1/3, 2/3, 1\}$ and then use a cubic polynomial interpolation, (14), to find the parameters for a given value of α not included in the set. The equation is identical for T_i. Eq. (10) can lead to negative α values in some cases (especially for lag dominant processes with very low values of $\frac{L}{\tau}$). In those cases, the controller parameters can still be obtained through extrapolation, using Eq. (14).

$$\begin{aligned}K_{p,opt}&=K_{p0}+(-5.5K_{p0}+9K_{p13}-4.5K_{p23}+K_{p1})\alpha\\&+(9K_{p0}-22.5K_{p13}+18K_{p23}-4.5K_{p1})\alpha^2\\&+(-4.5K_{p0}+13.5K_{p13}-13.5K_{p23}+4.5K_{p1})\alpha^3\end{aligned} \tag{14}$$

The tuning equations shown in the sequel are the result of fitting functions to a grid of exactly computed PI parameters that minimize the disturbance IAE of process (7), constrained to robustness M_s. The grid covers several values of $\frac{L}{\tau}$ and several values of M_s.

The resulting functions for $\alpha = 0$ are shown in Eqs. (15) and (16).

$$K_pK=0.27+1.35(M_s-1.2)^{1.1}+\frac{0.05957+0.4746(M_s-1.12)^{0.75}}{(\frac{L}{\tau})^{1.12}} \tag{15}$$

$$\frac{T_i}{\tau}=A(M_s)\left(1-e^{-B(M_s)\frac{L}{\tau}}\right)+C(M_s)\frac{L}{\tau}+D(M_s)\frac{L}{\tau}e^{-\frac{L}{\tau}} \tag{16}$$

where

$$A(M_s) = 0.72 - 0.12(M_s - 1.2)$$

$$B(M_s) = 4.552 + 9.984e^{-5.028(M_s-1.2)}$$

$$C(M_s) = 7.382 - 12.29M_s + 7.218M_s^2 - 1.342M_s^3$$

$$D(M_s) = -0.05 + 0.35(M_s - 1.2)$$

The resulting functions for $\alpha = 1/3$ are shown in Eqs. (17) and (18).

$$K_pK = 0.165+0.63(M_s-1.2)^{1.1}+\frac{0.01121 + 0.497(M_s - 1.12)^{0.55}}{(\frac{L}{\tau})^{1.08}} \quad (17)$$

$$\frac{T_i}{\tau} = A(M_s)\left(1 - e^{-B(M_s)\frac{L}{\tau}}\right) + C(M_s)\frac{L}{\tau} + D(M_s)\frac{L}{\tau}e^{-\frac{L}{\tau}} \quad (18)$$

where

$$A(M_s) = 0.71 - 0.1(M_s - 1.2)$$

$$B(M_s) = 4.16 + 10.34e^{-3.206(M_s-1.2)}$$

$$C(M_s) = 6.939 - 11.71M_s + 6.758M_s^2 - 1.2276M_s^3$$

$$D(M_s) = -0.05 + 0.35(M_s - 1.2)$$

The resulting functions for $\alpha = 2/3$ are shown in Eqs. (19) and (20).

$$K_pK = 0.097 + 0.35(M_s - 1.2) + \frac{0.06 + 0.47(M_s - 1.13)^{0.65}}{(\frac{L}{\tau})^{1.05}} \quad (19)$$

$$\frac{T_i}{\tau} = A(M_s)\left(1 - e^{-B(M_s)\frac{L}{\tau}}\right) + C(M_s)\frac{L}{\tau} + D(M_s)\frac{L}{\tau}e^{-\frac{L}{\tau}} \quad (20)$$

where

$$A(M_s) = 0.75 - 0.15(M_s - 1.2)$$

$$B(M_s) = 4.902 + 8.092e^{-3.408(M_s-1.2)}$$

$$C(M_s) = 4.876 - 8.113M_s + 4.594M_s^2 - 0.8076M_s^3$$

$$D(M_s) = -0.05 + 0.35(M_s - 1.2)$$

The resulting functions for $\alpha = 1$ are shown in Eqs. (21) and (22).

$$K_pK = 0.082+0.3(M_s-1.2)^{1.3}+\frac{0.02674 + 0.4916(M_s - 1.12)^{0.55}}{(\frac{L}{\tau})^{1.04}} \quad (21)$$

$$\frac{T_i}{\tau} = A(M_s)\left(1 - e^{-B(M_s)\frac{L}{\tau}}\right) + C(M_s)\frac{L}{\tau} + D(M_s)\frac{L}{\tau}e^{-\frac{L}{\tau}} \quad (22)$$

where

$$A(M_s) = 0.73 - 0.13(M_s - 1.2)$$

$$B(M_s) = 4.932 + 8.244e^{-3.164(M_s-1.2)}$$

$$C(M_s) = 4.81 - 8.038M_s + 4.548M_s^2 - 0.8037M_s^3$$

$$D(M_s) = 0.37(M_s - 1.2)$$

To use the tuning equations, the user must follow the procedure:

- Select the desired robustness in terms of the required M_s.
- Measure the gain K and the times $t_{5\%}$, $t_{35\%}$ and $t_{85\%}$ from the step response obtained in the experiment.
- Apply Eqs. (8), (9) and (10) to obtain L, τ and α.
- Apply Eqs. (15), (16), (17), (18), (19), (20), (21), (22) to obtain K_p and T_i for the four values of α in the set $\{0, 1/3, 2/3, 1\}$.
- Obtain the final values of K_p and T_i through the cubic polynomial interpolation (14).

The value of M_s is an approximation, since the final exact M_s depends on the real process that is unknown.

The previous tuning equations give an approximation of the optimal PI controller (the one that minimizes the IAE of disturbance rejection). If a slower PI controller is desired, to reduce the high frequency noise amplification, one possibility is to increase robustness. However, this will worsen the compromise between performance and noise amplification (i.e. for a given noise amplification, the IAE will be higher). A second possibility could be to detune the PI controller maintaining the desired robustness. This will lead to a better performance (lower IAE) for the same noise amplification. In order to detune the controller, the gain K_p can simply be reduced, but the integral time T_i must be changed in a specific way to maintain the robustness. Therefore, a detuning equation is needed for T_i as a function of K_p. If K_p is detuned as

$$K_p = \gamma K_{p,opt} \quad (23)$$

with $\gamma < 1$, then the equation of T_i is

$$T_i = f_3'(\frac{L}{\tau}, M_s, \alpha, \gamma)T_{i,opt} \quad (24)$$

For selecting a slower PI controller, the user simply needs to select a detuning factor $\gamma \in (0, 1]1$, and calculate K_p and T_i using Eqs. (23), (24).

The detuning factor must be selected taking into account the desired compromise between performance and high frequency noise amplification. A value of $\gamma = 1$ in the detuning factor implies the fastest response (lowest IAE), but the highest noise amplification. A value $\gamma \to 0$ leads to a response as slow as desired, and a noise amplification as low as required. The user must decide depending on the sensor noise, and on the actuator sensitivity to fast changes. For example, if the actuator has moving parts that can wear out, the noise amplification should be kept low, while if the actuator has no moving parts, then the noise amplification can be higher to improve performance. Another aspect to be taken into account in order to choose the detuning factor is the actuator saturation: if the actuator range is small, then the response will not be as fast as expected, even if the detuning factor is high, and hence, a lower detuning factor should be chosen to reduce the noise amplification.

The function $f_3'(\frac{L}{\tau}, M_s, \alpha, \gamma)$ is the result of fitting a grid of T_i values obtained by computing the PI controllers that keep the desired M_s with lower values of K_p. As in the case of the main tuning equations, the proposal is to first obtain functions of M_s, $\frac{L}{\tau}$ and γ, for fixed values of α in the set $\{0, 1/3, 2/3, 1\}$ and then use a cubic polynomial interpolation, (14), to find the parameters for a given value of α not included in the set.

The grid in this case covers a range of values of M_s, $\frac{L}{\tau}$ and γ. Obviously, the resulting functions are different for each value of α.

The proposed general form of the function (obtained through extensive trial and error) is

$$f_3'(\frac{L}{\tau}, M_s, \gamma) = 1 + A(M_s, \frac{L}{\tau})(\gamma - 1) + B(M_s, \frac{L}{\tau})(\gamma - 1)^2 + C(M_s, \frac{L}{\tau})(\gamma - 1)^3 \quad (25)$$

where the expression for $\alpha = 0$ is

$$A(M_s, \frac{L}{\tau}) = 1.2 + 0.56(M_s - 1.2)^{1.3} + \frac{0.0009184}{(\frac{L}{\tau})^{1.75+0.25(M_s-1.2)^{2.8}}}$$

$$B(M_s, \frac{L}{\tau}) = 0.4 + 0.6(M_s - 1.2)^{1.3} + \frac{1}{(\frac{L}{\tau})^{0.3213}}$$

$$C(\frac{L}{\tau}) = 1.055 + 0.9914e^{-2.192\frac{L}{\tau}}$$

for $\alpha = 1/3$, B and C are the same, and

$$A(M_s, \frac{L}{\tau}) = 1.223 + 0.56(M_s - 1.2)^{1.3} + \frac{0.0008405}{(\frac{L}{\tau})^{1.625+0.35(M_s-1.2)}}$$

for $\alpha = 2/3$, B and C are the same, and

$$A(M_s, \frac{L}{\tau}) = 1.219 + 0.58(M_s - 1.2)^{1.25} + \frac{0.001925}{(\frac{L}{\tau})^{1.37+0.2226\arctan(3.962(M_s-1.5))}}$$

while for $\alpha = 1$, B and C are the same, and

$$A(M_s, \frac{L}{\tau}) = 1.232 + 0.58(M_s - 1.2)^{1.3} + \frac{0.002}{(\frac{L}{\tau})^{0.872+0.7755(M_s-1.2)^{0.28}}}$$

In order to compute the controller for a required M_s, and a given value of γ, the previous equations are used to compute four controllers for the values of $\alpha \in \{0, 1/3, 2/3, 1\}$. The final PI controller parameters are obtained through cubic interpolation from those four controllers (Eq. (14)), using the value of α obtained from Eq. (10).

The computational effort needed to compute the controller parameters using the developed equations is very small, since there are no iterations at all. Even though the tuning equations are relatively complex, they are computed in few milliseconds in a simple computer or smartphone, using the application described in the next section.

4. PI tuning application for Android

The tuning equations are relatively complex to be computed by hand. However, with a computer, the calculation of the controller parameters is immediate. In order to facilitate the use of the proposed tuning method, an application for Android has been developed that implements the tuning equations. The tool has been developed in Javascript and can be run or downloaded from https://sites.google.com/a/uji.es/freepidtools/pituningapp. The Fig. 2 shows the two tabs of the application. In the first tab (Data input), the user must introduce the measurements taken in the step response experiment: Δu, $\Delta y(\infty)$, $t_{5\%}$, $t_{35\%}$ and $t_{85\%}$. The application computes internally the model parameters (K, τ, L and α), that are hidden by default, but can be shown if desired. Alternatively, the user could enter manually the values of model parameters, but this is not the main purpose of the app. In the second tab, the desired robustness is then defined introducing the value of M_s (through a slider or a numeric input). Finally, the user can select the desired detuning factor ($\gamma \leq 1$). For $\gamma = 1$, the optimum PI controller is obtained. For lower values of γ one obtains slower PI controllers with the same robustness. The application shows the performance indicator $IE = \frac{1}{K_i}$, and the high frequency noise amplification, K_p, to guide in the selection of the detuning parameter. The Integral of Error of disturbance rejection (IE) is an approximation of the IAE if the response is not too oscillatory. The expected response to a step change in the reference and to a step input disturbance is also shown in a graph. The weighting factor of the proportional part of the reference can be selected with a slider, to show its effect in the

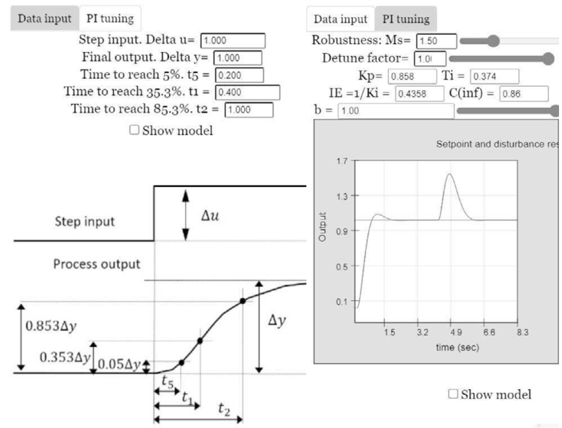

Fig. 2. PI tuning application for Android.

overshoot reduction of the reference response. This weighting factor, b, implies that the proportional part of the controller is computed as $K_p(br - y)$ instead of $K_p(r - y)$. The simulation is computed using model (7) when $\alpha \geq 0$. When $\alpha < 0$, model (7) has a negative time delay, and cannot be used for simulation. In that case, model (26) is used instead, forcing a zero time delay, and fixing two different time constants. The tuning equations, however, are valid for $\alpha < 0$.

$$G(s) = \frac{K}{(1+\tau s)(1+\frac{(1-\alpha)L}{2}s)(1+\frac{(1+\alpha)L}{2}s)} \quad (26)$$

5. Validation of the tuning equations

For the validation of the tuning procedure, PI controllers have been tuned for the following test batch of models obtained from those proposed in [2] and [3]:

$$G_{1,2,3,4}(s) = \frac{1}{(s+1)^p} \quad p = 3, 4, 5, 6 \quad (27)$$

$$G_5(s) = \frac{e^{-s}}{(s+1)^3} \quad (28)$$

$$G_6(s) = \frac{1}{(s+1)(1+0.1s)^2} \quad (29)$$

$$G_7(s) = \frac{1}{(s+1)(1+0.2s)(1+0.2^2 s)(1+0.2^3 s)} \quad (30)$$

$$G_8(s) = \frac{1}{(s+1)(1+0.05s)^2} \quad (31)$$

$$G_9(s) = \frac{1-2s}{(s+1)^3} \quad (32)$$

$$G_{10}(s) = \frac{e^{-5s}}{(s+1)^3} \quad (33)$$

$$G_{11}(s) = \frac{e^{-0.1s}}{(s+1)(1+0.1s)^2} \quad (34)$$

These models capture the typical dynamics encountered in control applications: multiple poles, different poles, time delay, non minimum phase, and cover a wide range of values of $\frac{L}{\tau}$, therefore the conclusions of the next study may be assumed to apply for most of the industrial self regulating non oscillatory process models.

For those processes, the step response is obtained to apply the proposed tuning procedure. The results are compared (in terms of performance and robustness) with the optimal controllers designed with the full models. PI controllers with robustness $M_s = 1.4$ and $M_s = 2$ are tuned. The obtained results will also be compared to the results of the tuning methods proposed by Murrill (low robustness, [22]) and Astrom and Hagglund (high robustness, [20]), shown in Table 1. The results have also been compared with the modified IMC method proposed by Lee in [14], with tuning equations defined in (35). In this case, for each plant, the values of tuning parameter λ have been obtained by trial and error to reach the same robustness, in terms of M_s with the exact plant model, as the proposed approach.

$$K_p = \frac{\tau}{K(\lambda + \tau)} \quad ; \quad T_i = min(\tau, 5\lambda) \tag{35}$$

The results have also been compared with the improved iSIMC method proposed by Grimholt and Skogestad in [23], with tuning equations defined in (36). In this case, for each plant, the values of tuning parameter λ have also been obtained by trial and error to reach the same robustness, in terms of M_s, as the proposed approach.

$$K_p = \frac{\tau + L/3}{K(\lambda + L)} \quad ; \quad T_i = min(\tau + L/3, 4(\lambda + L)\lambda) \tag{36}$$

The Table 2 shows the results for the plants with medium to low values of $\frac{L}{\tau}$. The proposed tuning procedure gives really accurate results for most systems with any value of $\frac{L}{\tau}$. In the case of system G_7, the achieved IAE is about a 30% higher than the optimum, with a higher robustness than expected. This is because this is the more lag dominant plant in the batch. However, even in this case, the results are better than the other methods. The Table 3 shows the results for large values of $\frac{L}{\tau}$. The obtained controllers are also really close to the optimum ones in all cases except the non minimum phase system (G_9). Comparing our approach with the method of Murrill [22], for low values of $\frac{L}{\tau}$, the controllers are very similar. However, for higher values, the result of Murrill tends to be much more robust than expected, and the performance much worse. In the case of the method of Astrom and Hagglund, their controllers tend to be also much more robust than expected, and the performance much worse. The only exceptions in this case are models with high values of $\frac{L}{\tau}$, as G_9 and G_{10}.

On the other hand, the comparison with the methods in [14, 23] is also favourable to the new proposed approach. First, the required values of λ to achieve a given robustness is very dependent on the process, and cannot be known in advance (in this case they have been obtained by trial and error). And second, for the same robustness, the performance (in terms of IAE) is always worse with the Lee and Grimholt tunings, compared with the proposed one.

The Tables 2 and 3 also show the control effort for each controller, computed as the H2 norm from disturbance to control action. This norm is equal to the integral of the square of the derivative of the control action in the response to a step disturbance. One can see that the increase of performance (reduction of IAE) is achieved by means of increasing the control effort, as expected. However, the control effort is not excessive in any case. Furthermore, the proposed tuning approach allows to select a detuned slower controller in case the user wants to reduce the control effort (obviously at the expense of a larger IAE). The high frequency measurement noise amplification is also shown in the tables. This amplification equals the controller proportional gain K_p, that is the high frequency gain from measurement noise to control action. Again, if the user wants to reduce the noise amplification, the proposed tuning method allows to select a slower (detuned) controller with a lower noise amplification, but with the same robustness. Again, the result would be a larger IAE.

Table 2
Comparison of the PI controller results reached by the proposed tuning method, the optimum exact one and the one proposed by Murrill (low robustness) and Astrom and Hagglund (high robustness), shown in Table 1. The results of Lee method [14] and Grimholt method [23] are also shown, where the value of λ has been selected to achieve the required robustness. Opt: Optimum exact method. App: proposed method. M: Murrill. A&H Astrom and Hagglund. Lee: Lee method. Grimholt: Grimholt method.

G_n, ($\frac{L}{\tau}$)	Design	IAE	M_s	Control effort	Noise amplif.
G_8, (0.115)	Opt ($M_s = 2$)	0.043	2	2.76	8.03
	Opt ($M_s = 1.4$)	0.122	1.4	1.71	3.85
	App ($M_s = 2$)	0.049	1.926	2.65	8.19
	App ($M_s = 1.4$)	0.142	1.359	1.58	3.28
	M ($M_s = 2$)	0.048	1.9	2.56	7
	A&H ($M_s = 1.4$)	0.268	1.22	1.23	2.34
	Lee ($\lambda = 0.0549$)	0.059	1.926	2.533	5.84
	Lee ($\lambda = 0.128$)	0.157	1.359	1.63	4.08
	Grimholt ($\lambda = 0.0013$)	0.0525	1.926	2.7	8.86
	Grimholt ($\lambda = 0.112$)	0.2	1.359	1.622	4.54
G_6, (0.221)	Opt ($M_s = 2$)	0.132	2	1.96	4.51
	Opt ($M_s = 1.4$)	0.33	1.4	1.2	2.06
	App ($M_s = 2$)	0.155	1.863	1.83	4.28
	App ($M_s = 1.4$)	0.395	1.343	1.1	1.85
	M ($M_s = 2$)	0.151	1.9	1.82	3.72
	A&H ($M_s = 1.4$)	0.753	1.19	0.8	1.05
	Lee ($\lambda = 0.0933$)	0.18	1.863	1.74	3.15
	Lee ($\lambda = 0.228$)	0.45	1.343	1.12	2.21
	Grimholt ($\lambda = 0$)	0.185	1.843	1.82	4.8
	Grimholt ($\lambda = 0.248$)	0.47	1.343	1.11	2.27
G_7, (0.263)	Opt ($M_s = 2$)	0.1457	2	1.89	4.67
	Opt ($M_s = 1.4$)	0.3868	1.4	1.12	1.93
	App ($M_s = 2$)	0.197	1.753	1.63	3.87
	App ($M_s = 1.4$)	0.495	1.323	1	1.65
	M ($M_s = 2$)	0.203	1.761	1.57	3.14
	A&H ($M_s = 1.4$)	0.983	1.179	0.71	0.84
	Lee ($\lambda = 0.1072$)	0.238	1.753	1.52	2.7
	Lee ($\lambda = 0.258$)	0.523	1.323	1.02	1.92
	Grimholt ($\lambda = 0$)	0.257	1.67	1.555	4.12
	Grimholt ($\lambda = 0.28$)	0.544	1.323	1.02	2
G_{11}, (0.323)	Opt ($M_s = 2$)	0.298	2	1.55	2.48
	Opt ($M_s = 1.4$)	0.642	1.4	0.95	1.22
	App ($M_s = 2$)	0.32	1.947	1.51	2.53
	App ($M_s = 1.4$)	0.666	1.385	0.93	1.23
	M ($M_s = 2$)	0.285	2.131	1.65	2.58
	A&H ($M_s = 1.4$)	1.296	1.196	0.64	0.66
	Lee ($\lambda = 0.1291$)	0.325	1.947	1.5	2.2
	Lee ($\lambda = 0.4$)	0.722	1.385	0.93	1.38
	Grimholt ($\lambda = 0.066$)	0.39	1.947	1.508	2.84
	Grimholt ($\lambda = 0.446$)	0.768	1.385	0.926	1.43
G_1, (0.77)	Opt ($M_s = 2$)	1.6	2	0.74	1.63
	Opt ($M_s = 1.4$)	3.071	1.4	0.44	0.63
	App ($M_s = 2$)	1.69	1.930	0.71	1.45
	App ($M_s = 1.4$)	2.8	1.445	0.47	0.67
	M ($M_s = 2$)	2	1.685	0.6	1.11
	A&H ($M_s = 1.4$)	6	1.2	0.3	0.29
	Lee ($\lambda = 0.324$)	2.08	1.93	0.68	1.04
	Lee ($\lambda = 1.44$)	2.84	1.445	0.465	0.63
	Grimholt ($\lambda = 0.215$)	1.71	1.93	0.703	1.4
	Grimholt ($\lambda = 1.47$)	2.84	1.445	0.475	0.782

(continued on next page)

Table 2 (*continued*).

$G_n, (\frac{L}{\tau})$	Design	*IAE*	M_s	Control effort	Noise amplif.
G_2, (1.05)	Opt ($M_s = 2$)	3.04	2	0.57	1.09
	Opt ($M_s = 1.4$)	5.2	1.4	0.35	0.43
	App ($M_s = 2$)	3.05	2.023	0.58	1.06
	App ($M_s = 1.4$)	4.61	1.464	0.38	0.5
	M ($M_s = 2$)	3.834	1.656	0.45	0.82
	A&H ($M_s = 1.4$)	8.39	1.234	0.26	0.25
	Lee ($\lambda = 0.457$)	3.61	2.023	0.56	0.78
	Lee ($\lambda = 2.485$)	4.67	1.464	0.37	0.44
	Grimholt ($\lambda = 0.5$)	3.114	2.023	0.574	1
	Grimholt ($\lambda = 2.6$)	4.75	1.464	0.38	0.58

In summary, as expected, the use of parameter α allows the tuning procedure to reach very accurate results for a wide range of $\frac{L}{\tau}$ and for different lag–delay ratios. However, for non minimum phase systems, the result is not correct, as the obtained controller is much less robust than expected. One could still use the approach with those systems, but selecting a higher initial robustness than the one required.

6. Examples

In this section, some examples are developed to illustrate the tuning procedure. All computations of the tuning equations have been done with the Android (or web) application that has been developed. The examples include systems with very different values of $\frac{L}{\tau}$, and also with very different lag–delay ratio, to demonstrate the wide range of systems covered.

6.1. Example 1

Consider the system,

$$G_{11}(s) = \frac{e^{-0.1s}}{(s+1)(1+0.1s)^2} \tag{37}$$

The step response measurements are $t_{35\%} = 0.743$, $t_{85\%} = 2.221$ and $t_{5\%} = 0.298$ that lead to parameters $K = 1$, $\tau = 0.991$, $L = 0.321$ and $\alpha = 0.189$. Assume that a robustness defined by $M_s = 1.4$ is desired, and the fastest PI controller is required, no matter which is the noise amplification. The approximate "optimum" PI controller obtained from the tuning equations is defined by $K_p = 1.232$, $T_i = 0.812$. Fig. 3 shows the results of the application, with the expected response. If this controller is applied to the exact process, the resulting robustness is $M_s = 1.385$, and the response is very similar to the one predicted by the application, with $IAE = 0.666$ (see Fig. 4). As a comparison, the optimal PI controller computed with the exact model (the one that minimizes IAE with $M_s = 1.4$) is very similar ($K_p = 1.218$, $T_i = 0.77$, $M_s = 1.4$, $IAE = 0.642$). Fig. 4 shows the response to a step change in the reference and to a step change in the disturbance with a measurement noise of amplitude 0.1 of the proposed controller and the PI controller tuned by Astrom and Hagglund method, and the one by Lee method (with $\lambda = 0.377$). The proposed controller is the fastest one, with lower IAE, with a higher but reasonable control effort and noise amplification.

Consider now that a noise amplification lower than 0.5 is desired (because the actuator is very sensitive to fast changes). In this case, the optimal PI controller has a noise amplification of 1.213, i.e., too high. The PI controller must be detuned while maintaining the robustness until the noise amplification is 0.5. This can be done by simply moving the PI detuning slider in the application. With $\gamma = 0.411$ the result is $K_p = 0.499$, $T_i = 0.477$. The result when applied to the original plant is a value $M_s = 1.36$ and $IAE = 1.148$. Fig. 5 shows the result of the tuning app, and Fig. 6 the response of the true system with the detuned controller, compared to the optimal one (with $\gamma = 1$). The control action fluctuation due to noise is much lower in the detuned controller, at the price of a slower response (higher IAE). As a comparison, the optimal PI controller (the one with $K_p < 0.5$ and $M_s = 1.4$),

Table 3
Comparison of the PI controller results reached by the proposed tuning method, the optimum exact one and the one proposed by Murrill (low robustness) and Astrom and Hagglund (high robustness), shown in Table 1. The results of Lee method [14] and Grimholt method [23] are also shown, where the value of λ has been selected to achieve the required robustness. Opt: Optimum exact method. App: proposed method. M: Murrill. A&H Astrom and Hagglund. Lee: Lee method. Grimholt: Grimholt method.

$G_n, (\frac{L}{\tau})$	Design	*IAE*	M_s	Control effort	Noise amplif.
G_3, (1.3)	Opt ($M_s = 2$)	4.47	2	0.48	0.87
	Opt ($M_s = 1.4$)	7.5	1.4	0.3	0.34
	App ($M_s = 2$)	4.46	2.055	0.5	0.88
	App ($M_s = 1.4$)	6.45	1.473	0.33	0.42
	M ($M_s = 2$)	6.08	1.608	0.37	0.66
	A&H ($M_s = 1.4$)	10.38	1.267	0.24	0.23
	Lee ($\lambda = 0.745$)	5.36	2.055	0.49	0.61
	Lee ($\lambda = 3.51$)	6.51	1.473	0.32	0.35
	Grimholt ($\lambda = 0.975$)	4.55	2.055	0.498	0.828
	Grimholt ($\lambda = 3.72$)	6.69	1.473	0.327	0.488
G_4, (1.53)	Opt ($M_s = 2$)	5.88	2	0.43	0.76
	Opt ($M_s = 1.4$)	9.47	1.4	0.27	0.31
	App ($M_s = 2$)	5.86	2.061	0.44	0.77
	App ($M_s = 1.4$)	8.42	1.467	0.29	0.37
	M ($M_s = 2$)	8.65	1.562	0.31	0.57
	A&H ($M_s = 1.4$)	12.18	1.296	0.23	0.22
	Lee ($\lambda = 1.04$)	7.15	2.061	0.44	0.51
	Lee ($\lambda = 4.68$)	8.51	1.467	0.284	0.29
	Grimholt ($\lambda = 1.362$)	5.98	2.061	0.445	0.72
	Grimholt ($\lambda = 4.956$)	8.74	1.467	0.289	0.426
G_5, (1.34)	Opt ($M_s = 2$)	2.4	2	0.55	0.8
	Opt ($M_s = 1.4$)	4.22	1.4	0.33	0.32
	App ($M_s = 2$)	3.65	2.022	0.56	0.8
	App ($M_s = 1.4$)	5.36	1.463	0.36	0.38
	M ($M_s = 2$)	4.97	1.632	0.42	0.64
	A&H ($M_s = 1.4$)	8.19	1.277	0.27	0.23
	Lee ($\lambda = 0.737$)	4.33	2.022	0.54	0.57
	Lee ($\lambda = 3$)	5.38	1.463	0.36	0.33
	Grimholt ($\lambda = 0.964$)	3.69	2.022	0.553	0.767
	Grimholt ($\lambda = 3.23$)	5.6	1.463	0.363	0.457
G_9, (1.84)	Opt ($M_s = 2$)	6.94	2	0.53	0.39
	Opt ($M_s = 1.4$)	11.03	1.4	0.3	0.18
	App ($M_s = 2$)	7.79	3.239	0.88	0.61
	App ($M_s = 1.4$)	8.16	1.672	0.42	0.29
	M ($M_s = 2$)	8.83	1.978	0.52	0.47
	A&H ($M_s = 1.4$)	9.73	1.486	0.34	0.22
	Lee ($\lambda = 0.821$)	9.78	3.239	0.82	0.426
	Lee ($\lambda = 3.93$)	8.13	1.672	0.4	0.234
	Grimholt ($\lambda = 1.4$)	7.9	3.239	0.873	0.596
	Grimholt ($\lambda = 4.7$)	8.81	1.672	0.42	0.34
G_{10}, (3.62)	Opt ($M_s = 2$)	8.24	2	0.37	0.44
	Opt ($M_s = 1.4$)	13.3	1.4	0.22	0.19
	App ($M_s = 2$)	10.06	1.973	0.37	0.46
	App ($M_s = 1.4$)	14.57	1.442	0.23	0.21
	M ($M_s = 2$)	25.81	1.302	0.18	0.24
	A&H ($M_s = 1.4$)	14.29	1.454	0.24	0.2
	Lee ($\lambda = 2.65$)	13.2	1.973	0.33	0.195
	Lee ($\lambda = 9.1$)	15.54	1.442	0.22	0.114
	Grimholt ($\lambda = 2.85$)	10.13	1.973	0.365	0.422
	Grimholt ($\lambda = 8.75$)	15.16	1.442	0.237	0.258

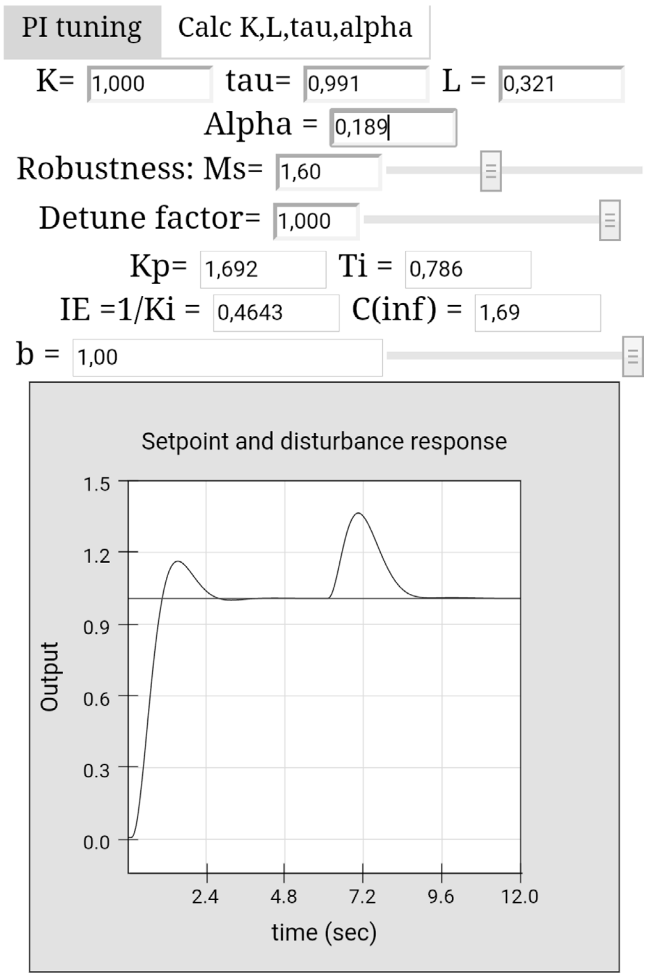

Fig. 3. Result of the application in example 1.

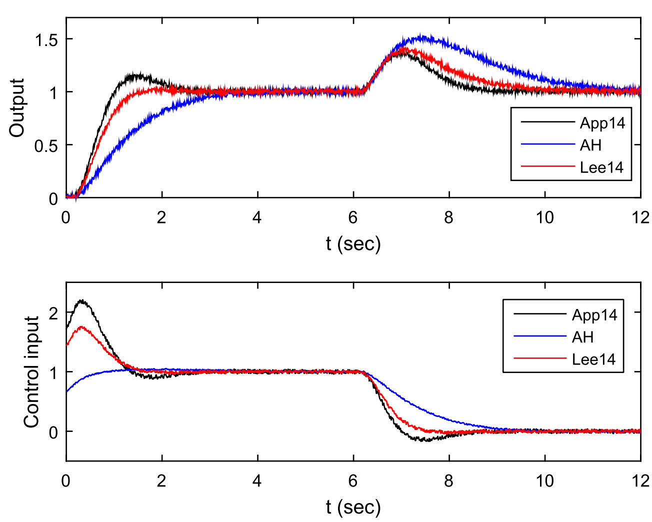

Fig. 4. Response of the true system in example 1. Proposed approach with $M_s = 1.4$ (black), Astrom and Hagglund (blue) and Lee with $\lambda = 0.377$ (red).

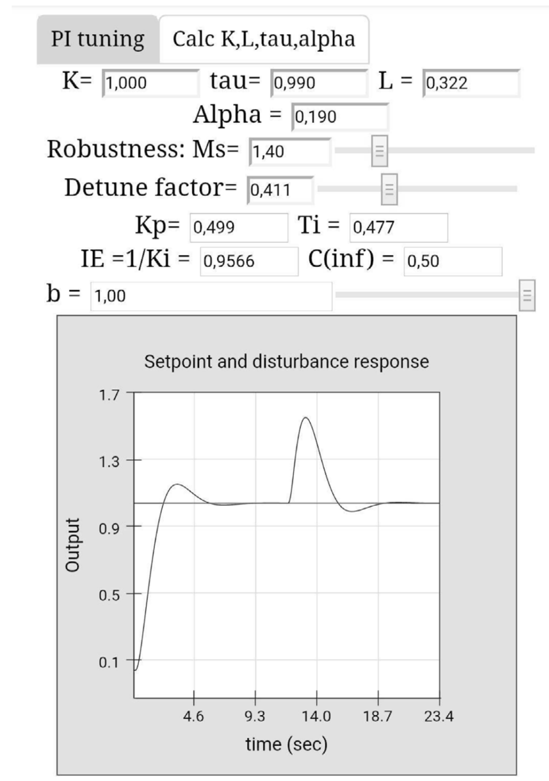

Fig. 5. Result of the application in example 1 after detuning.

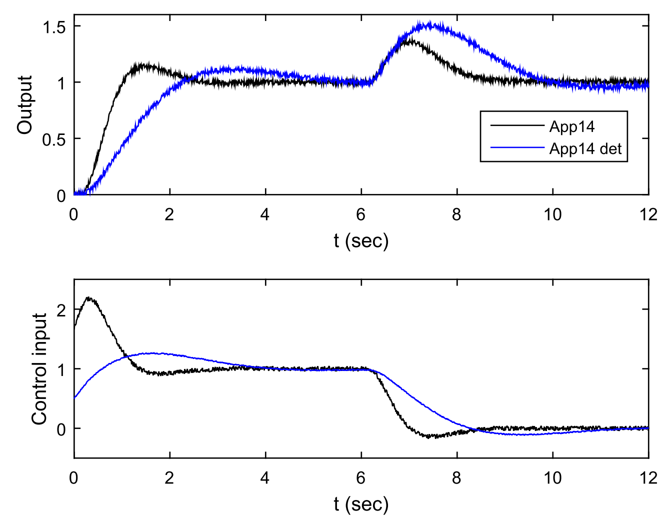

Fig. 6. Response of the true system in example 1 after detuning, compared with the minimum IAE controller.

computed with the exact model, is very similar ($K_p = 0.499$, $T_i = 0.447$, $M_s = 1.4$, $IAE = 1.116$), and the response is almost indistinguishable.

If instead of detuning the PI controller maintaining the robustness, the controller is made slower by increasing robustness (i.e. decreasing M_s), for the highest robustness permitted by the application ($M_s = 1.2$), the result is $K_p = 0.664$, $T_i = 0.87$, with $IAE = 1.31 > 1.148$. Therefore, if the system is made slower by increasing the robustness, the ratio performance-noise amplification is clearly worse.

6.2. Example 2

Consider the system,

$$G_4(s) = \frac{1}{(s+1)^6} \tag{38}$$

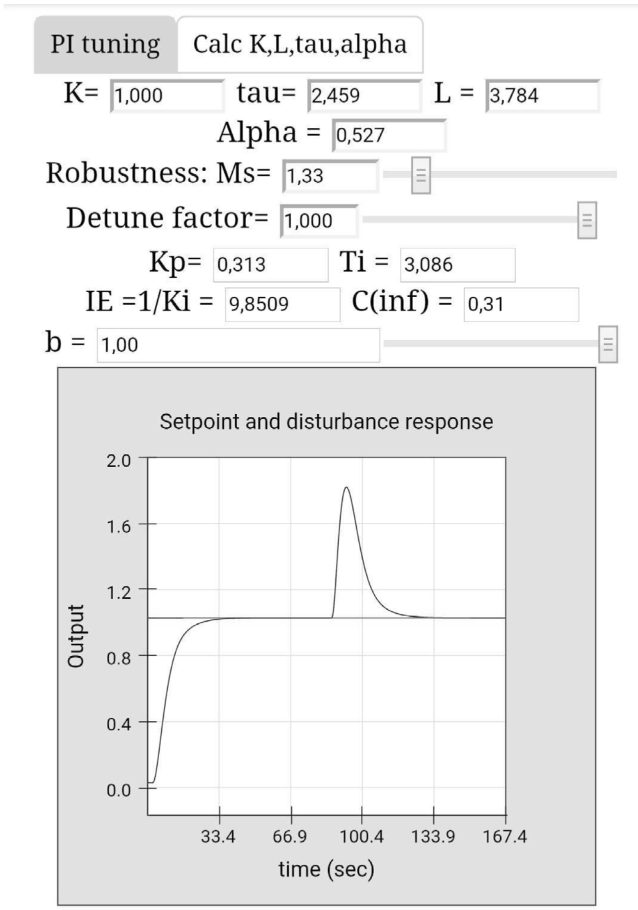

Fig. 7. Result of the application in example 2.

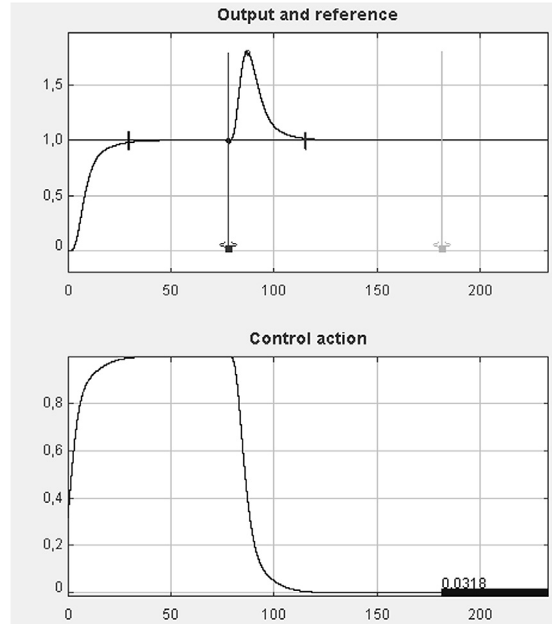

Fig. 8. Response of the true system in example 2, with the proposed controller, and with the one by Astrom and Hagglund method.

The step response measurements are $t_{35\%} = 4.8$, $t_{85\%} = 8.47$, $t_{5\%} = 2.61$ that lead to parameters $K = 1$, $\tau = 2.459$, $L = 3.784$, $\alpha = 0.527$. Assume that the most robust PI controller is to be found, such that $K_i \geq 0.1$. If the proposed tuning approach is applied to obtain the PI controller for $M_s = 1.6$, the result is a controller with $K_i = 0.16 > 0.1$, therefore the controller can be made more robust. Increasing robustness until a value of $K_i = 0.1$ is reached ($1/K_i = 10$), one obtains, for $M_s = 1.33$, the controller $K_p = 0.313$, $T_i = 3.086$. If this controller is applied to the real system, $M_s = 1.38$ and $IAE = 9.9$ are obtained. The response is very similar to the one predicted by the application, as shown in Figs. 7 and 8. Fig. 8 compares the response of the proposed PI with the one obtained with Astrom and Hagglund method, and with Lee method ($\lambda = 5.7$), that has a similar robustness. Our controller is very similar to the Lee method in this case, and slightly faster than the Astrom and Hagglund one.

6.3. Example 3

Consider the system,

$$G_{10}(s) = \frac{e^{-5s}}{(s+1)^3} \tag{39}$$

Assume that the PI controller of maximum robustness is to be found, such that the performance is $K_i \geq 0.09$ ($IE \leq 11.1$). The step response measurements are $t_{35\%} = 7.11$, $t_{85\%} = 9.75$, $t_{5\%} = 5.81$ that lead to parameters $K = 1$, $\tau = 1.771$, $L = 6.414$, $\alpha = 0.823$. If the proposed tuning approach is applied to obtain the PI controller, changing the robustness until $1/K_i \leq 11.1$, a controller is obtained, for $M_s = 1.61$, with $K_p = 0.298$, $T_i = 3.294$. If this controller is applied to the real system, $M_s = 1.669$ and $IAE = 11.05$ are obtained. The response is very similar to the one predicted by the application, as shown in Figs. 9 and 10. The Fig. 10 shows the response of the proposed controller compared to the Astrom and Hagglund and the Lee method ($\lambda = 10.5$). The proposed controller is clearly faster, in part due to the use of an intermediate robustness, that the other methods do not permit. The Fig. 11 shows the response of the proposed controller compared to the Merrill and the Lee method ($\lambda = 2.5$). The controller by Merrill is very slow in this case, as that method does not work well with very delay dominant processes. The controller by Lee method has a similar speed response, but is much more oscillatory.

6.4. Example 4

Consider the system,

$$G_7(s) = \frac{1}{(s+1)(1+0.2s)(1+0.2^2 s)(1+0.2^3 s)} \tag{40}$$

Assume an intermediate robustness PI controller ($M_s = 1.7$) is desired, but a performance defined by $K_i \geq 2$ is required, with the minimum noise amplification. The step response measurements are $t_{35\%} = 0.692$, $t_{85\%} = 2.185$, $t_{5\%} = 0.21$ that lead to parameters $K = 1$, $\tau = 1.003$, $L = 0.264$, $\alpha = -0.076$. In this case, the value of α is negative. The tuning equations are valid for negative values of α. However, the response simulation cannot be computed with a model with negative time delay (as would result with $\alpha < 0$). In that case, the developed application computes the simulation with a model with zero time delay and with two different time constants. If the optimal PI controller is tuned for $M_s = 1.7$, the result is $K_i = 3.75 > 2$, and a noise amplification of $K_p = 2.75$. To minimize this amplification the controller can be detuned until the K_i is slightly over 2, obtaining the controller $K_p = 0.713$, $T_i = 0.351$, with a noise amplification of 0.713, much lower than 2.75. When applied to the real system, $M_s = 1.6$, $IAE = 0.76$ are obtained, and a response that is very similar to the one predicted by the application, as shown in Figs. 12 and 13. In this case,

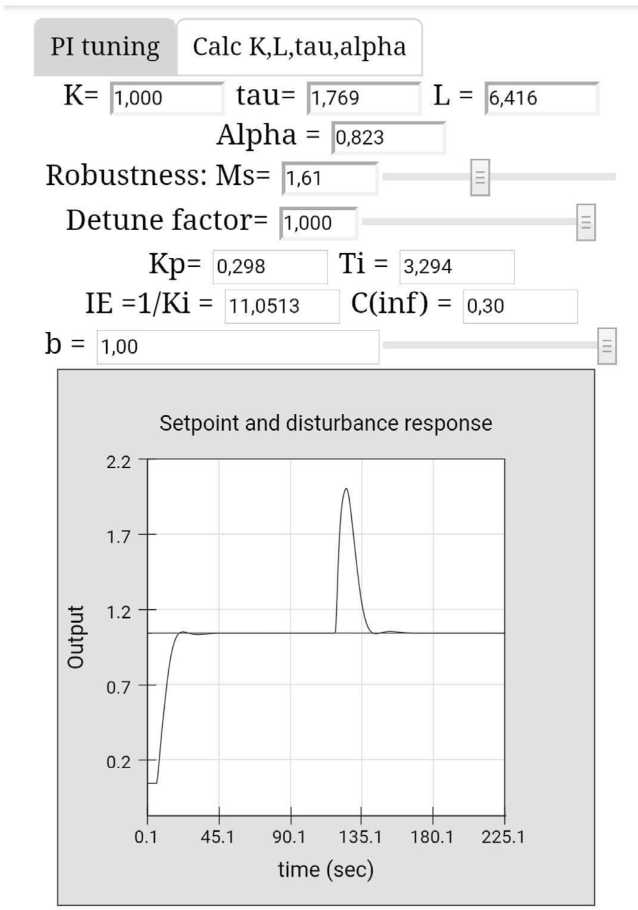

Fig. 9. Result of the application in example 3.

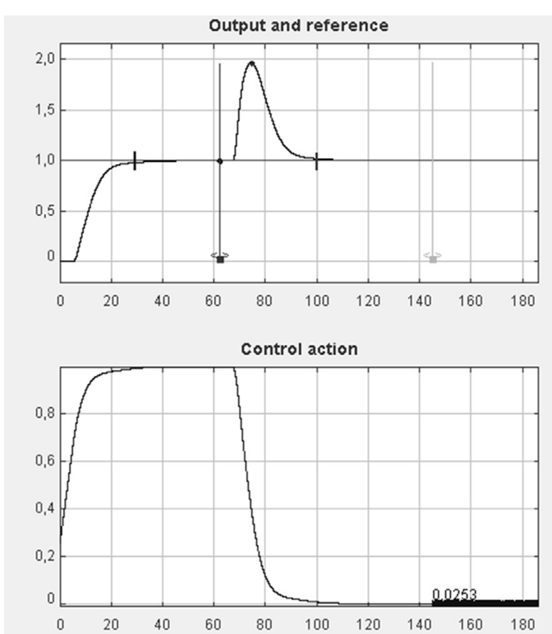

Fig. 10. Response of the true system in example 3, compared to Astrom and Hagglund, and Lee ($\lambda = 10.5$) methods.

the comparison with other tuning methods is not possible, since they do not allow to select an arbitrary intermediate robustness,

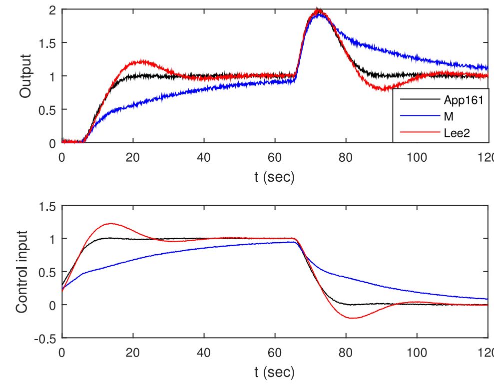

Fig. 11. Response of the true system in example 3, compared to Merrill and Lee ($\lambda = 2.5$) methods.

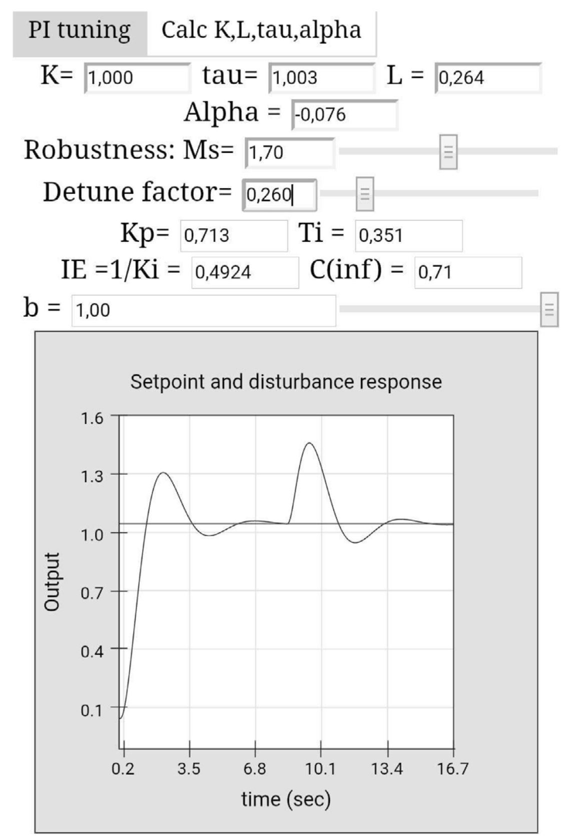

Fig. 12. Result of the application in example 4.

and neither to detune the controller maintaining the selected robustness.

7. Real plant case study

In this section, the PI tuning procedure is applied to a real laboratory plant, that consists of a water tank with an emptying valve, that can be filled up with a pump. The output of the process is the level of the tank (in cm), measured with a capacitive sensor. The control action is the duty cycle, in percentage, applied to an amplifier that drives the dc pump. The position of the emptying valve is used to introduce a disturbance.

For this process, an open loop experiment is performed, consisting of fixing the control action at 35%, and then (after the output has settled) changing this value from 35 to 45%. Fig. 14

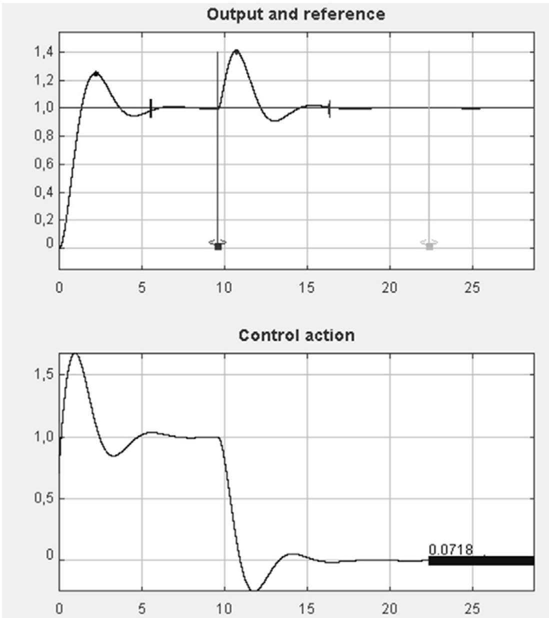

Fig. 13. Response of the true system in example 4.

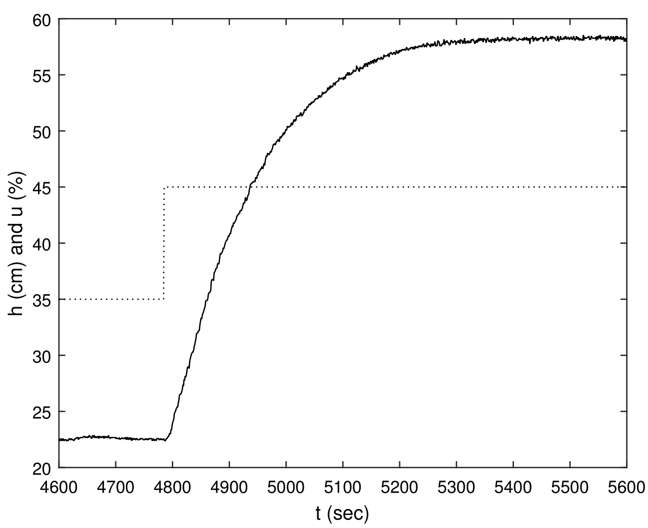

Fig. 14. Experiment performed in the tank. (...) Input. (-) Output.

shows the performed experiment. The resulting response to this step change in the input is used to measure the final value (to obtain the gain) and the times taken to reach the 5%, the 35.3% and the 85.3% of the final value. The results are $K = \frac{\Delta y}{\Delta u} = 3.572$, $t_{5\%} = 17$, $t_{35\%} = 77.1$, $t_{85\%} = 271.1$.

First, the optimum PI controller with robustness defined by $M_s = 1.4$ is tuned using the proposed approach. The resulting parameters are $K_p = 0.711$, $T_i = 78.1$. The Fig. 15 shows the response of the real system to a step change in the reference, and to a step change in the disturbance (a partial closing of the discharge valve at instant 1250). The response of the controlled system is correct, as expected. The amplitude of measurement noise is about 0.6 cm. The amplitude of the fluctuation of the control action due to noise is about $0.6K_p = 0.43\%$. The figure also shows the response of the PI controller tuned by Astrom and Hagglund method, and by Lee method with $\lambda = 6.07$. Both methods have an initial robustness objective of $M_s = 1.4$, as our approach. In the three cases, a weighting factor in the proportional part of the reference has been used in the PI controller to reduce the overshoot. Table 4 shows the resulting IAE, the control effort and the high frequency noise amplification in the three cases. It is clear that our approach leads to a much faster response (lower IAE), at the expense of a higher (but very reasonable) control effort and noise amplification.

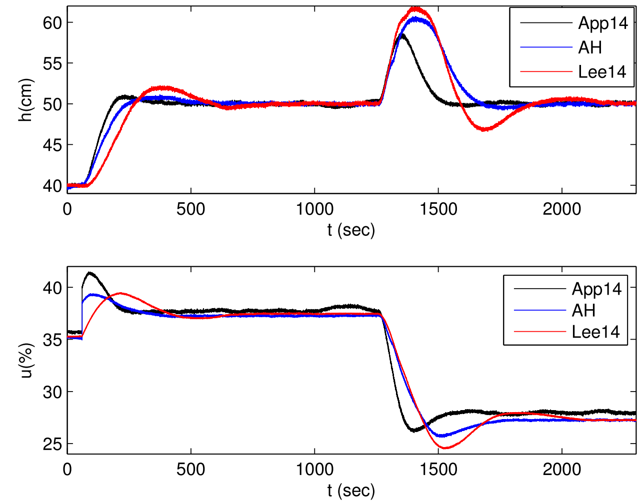

Fig. 15. Closed loop behaviour for robust PI ($M_s \approx 1.4$). Proposed approach (App14), Astrom and Hagglund approach (AH) and Lee approach (Lee).

If a less robust controller is selected now, with $M_s = 2$, the optimal controller obtained with our approach is defined by $K_p = 2.245$, $T_i = 75.95$. Fig. 16 shows the response of the real system to a step change in the reference, and to a step change in the disturbance. The response of the controlled system is also correct, but it is more oscillatory and faster (lower IAE) than the controller with $M_s = 1.4$, as expected. The amplitude of the fluctuation of the control action due to noise is about $0.6K_p = 1.35\%$, around three times the fluctuation with the slower and more robust controller. The figure also shows the response of the PI controller tuned by Merrill method, and by Lee method with $\lambda = 10.19$. Both methods have an initial robustness objective of $M_s = 2$, as our approach. In the three cases, a weighting factor in the proportional part of the reference has been used in the PI controller to reduce the overshoot (the proportional part of the controller is computed as $K_p(br - y)$ instead of $K_p(r - y)$). The Table 4 shows the resulting IAE, the control effort and the high frequency noise amplification in the three cases. It is clear that our approach leads to a much faster response (lower IAE), at the expense of a higher (but still very reasonable) control effort and noise amplification.

The proposed controller with $M_s = 2$ has the highest fluctuation of the control action due to noise. In order to reduce this fluctuation, the controller can be detuned by reducing the detune factor γ. This leads to a new controller of the same robustness ($M_s = 2$), but slower and with a lower noise amplification. Doing that until the same noise amplification of the controller with $M_s = 1.4$ is obtained, the result is (with $\gamma = 0.317$) a controller defined by $K_p = 0.711$, $T_i = 41.24$. The Fig. 17 shows the disturbance response of the controlled system, where the three controllers tuned with our approach are compared. The amplitude of control action fluctuation of the detuned controller with $M_s = 2$ is the same as in the controller with $M_s = 1.4$, but the response is faster (lower IAE), obviously at the expense of a more oscillating response. The controller with $M_s = 2$ without detuning has a higher control action fluctuation due to noise, but the response is faster (lower IAE).

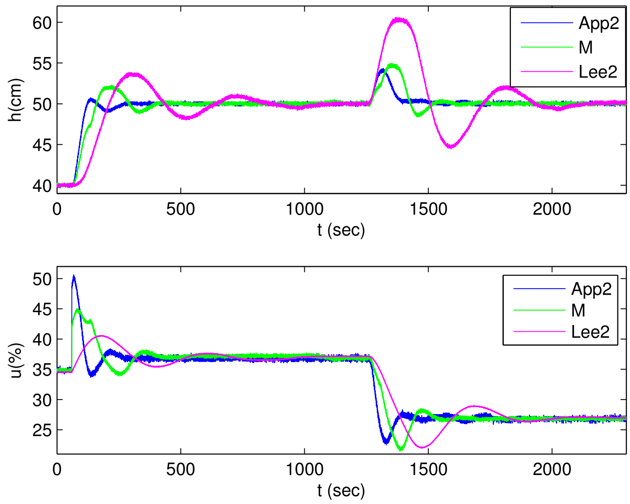

Fig. 16. Closed loop behaviour for less robust PI ($M_s \approx 2$). Proposed approach (App2), Merrill approach (M) and Lee approach (Lee).

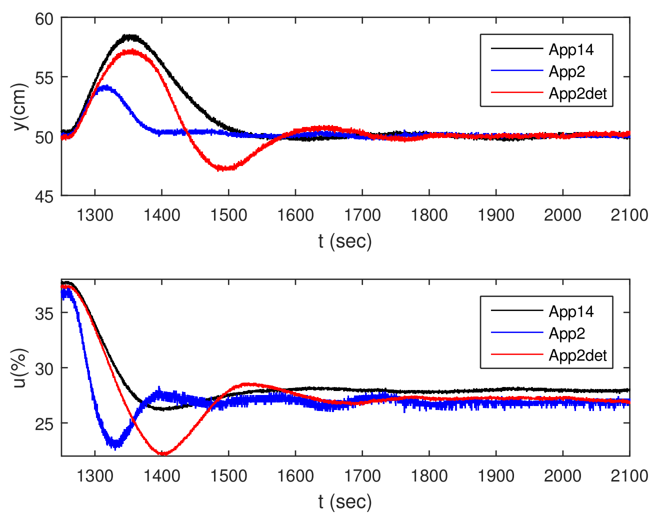

Fig. 17. Closed loop disturbance response. Detuned PI with $M_s = 2$ and $\gamma = 0.317$, compared to PI with $M_s = 1.4$ and PI with $M_s = 2$ without detuning.

Table 4
Comparison of the PI controller results reached in the experimental plant by the proposed tuning method and the one proposed by Murrill (low robustness) and Astrom and Hagglund (high robustness), shown in Table 1. The results of Lee method [14] are also shown, where the value of λ has been selected to achieve the required robustness. App: proposed method. M: Murrill. A&H Astrom and Hagglund. Lee: Lee method.

Design	*IAE*	Control effort	Noise amplif.
App ($M_s = 2$)	383	2.56	2.28
Merrill ($M_s = 2$)	617	2.15	1.39
Lee ($\lambda = 6.07$)	2677	1.45	0.27
App ($M_s = 1.4$)	1198	1.06	0.72
A&H ($M_s = 1.4$)	2448	0.61	0.43
Lee ($\lambda = 10.19$)	3053	0.75	0.26
App ($M_s = 2$, detuned)	1121	1.97	0.72

8. Conclusions

In this paper, an experimental PI tuning procedure has been presented. It is based on simple measurements taken from the open loop step response experiment, and the use of some tuning equations based on those measurements. The tuning equations include the desired value of M_s as a parameter to be freely selected by the user.

The main novelty of the PI tuning procedure is the possibility of selecting the desired robustness in a continuous range (in terms of M_s), and also the free selection of a detuning parameter to obtain slower controllers maintaining the robustness. In overall, the tuning method allows the user to select the desired robustness, and to tune the controller to reach the desired compromise between performance (IAE) and noise amplification. This can be done by selecting the adequate value of detuning parameter γ. If a value $\gamma = 1$ is selected, the minimum IAE and maximum noise amplification is obtained. For lower values of γ, higher values of IAE but lower values of noise amplification are obtained.

The tuning equations have been obtained by approximating the PI controllers that minimize the disturbance IAE with a given constraint in M_s, for a third order plus time delay model defined by the same three parameters that determine a FOTD model, plus a fourth parameter that is obtained using the time to reach the 5% of the output. This parameter is used to apportion the initial time delay of the FOTD model between a smaller time delay and two real poles.

In the case of over damped systems, the resulting tuning method is very accurate for a wide range of $\frac{L}{\tau}$ values, and for systems that are lag dominant or delay dominant. For non minimum phase systems, the tuning method is not really accurate, leading to controllers much less robust than expected.

The tuning equations have been implemented in an easy to use Android and web application, freely available at https://sites.google.com/a/uji.es/freepidtools/pituningapp (to download or run online).

The tuning equations have been tested using a batch of models, and several examples have been developed to show the applicability of the approach, including a real laboratory plant case.

Future research work will try to extend the proposed tuning method to PID controllers, to be able to find the required compromise between performance and noise amplification taking into account the derivative filter.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work has been supported by MICINN, Spain project number TEC2015-69155-R from the Spanish government.

References

[1] Borase Rakesh P, Maghade DK, Sondkar SY, Pawar SN. A review of PID control, tuning methods and applications. Int J Dyn Control 2020;1–10.
[2] Astrom KJ, Panagopoulos H, Hagglund T. Design of PI controllers based on non-convex optimization. Automatica 1998;34(5):585–601.
[3] Astrom KJ, Panagopoulos H, Hagglund T. Design of PID controllers based on constrained optimization. IEE Proc Control Theory Appl 2002;149(1):32–40.
[4] Sanchis R, Romero JA, Balaguer P. Tuning of PID controllers based on simplified single parameter optimisation. Internat J Control 2010;83(9):1785–98.
[5] Leva A. PID autotuning algorithm based on relay feedback. In: Control Theory and Applications, IEE Proceedings D [See Also IEE Proceedings-Control Theory and Applications]. 140, (5):1993, p. 328–38, ID: 1.
[6] da Ma Ming, jian Zhu Xin. A simple auto-tuner in frequency domain. Comput Chem Eng 2006;30(4):581–6.

[7] Ho WK, Hong Y, Hansson A, Hjalmarsson H, Deng JW. Relay auto-tuning of PID controllers using iterative feedback tuning. Automatica 2003;39(1):149–57.
[8] Reynoso-Meza G, Sanchis J, Blasco X, Martinez M. Algoritmos evolutivos y su emnpleo en el ajuste de controladores del tipo PID: estado actual y perspectivas. Rev Iberoam Autom E Inform Indust 2013;(10):251–68.
[9] Yan D, Zhou M, Chen S, Levy D, Lu Y. Handbook of Real-Time Computing. Springer; 2019.
[10] Zhao S-Z, Iruthayarajan MW, Baskar S, Suganthanr PN. Multi-objective robust PID controller tuning using two lbests multi-objective particle swarm optimization. Inform Sci 2011;181(16):3323–35.
[11] Amirinejad Moslem, Eslami M, Noori Ali Ali. Automatic PID controller parameter tuning using bees algorithm. Int J Sci Eng Res 2014;5.
[12] Romasevych Yuriy, Loveikin Viatcheslav, Makarets Valeriy. Optimal constrained tuning of PI-controllers via a new PSO-based technique. Int J Swarm Intell Res 2020;11(4):87–105.
[13] Wang GQ, Lu C, Pan W. IMC PID controller tuning for stable and unstable processes with time delay. Chem Eng Res Des 2016;(105):120–9.
[14] Lee J, Cho W, Edgar TF. Simple analytic PID controller tuning rules revisited. Ind Eng Chem Res 2013;5038–47.
[15] O'Dwyers Aidan. Handbook of PI and PID Controller Tuning Rules. Imperial College Press; 2006.
[16] Åström KJ, Hägglund T. Automatic tuning of simple regulators with specifications on phase and amplitude margins. Automatica 1984;20(5):645–51.
[17] Romero JA, Sanchis R, Balaguer P. PI and PID auto-tuning procedure based on simplified single parameter optimization. J Process Control 2011;21(6):840–51.
[18] Wang Liuping. Automatic tuning of PID controllers using frequency sampling filters. IET Control Theory Appl 2017;11(7):985–95.
[19] Jin Cheon Yu, Ryub Kyung Hwan, Sungb Su Whan, Leeb Jietae, Lee In-Beum. PID auto-tuning using new model reduction method and explicit PID tuning rule for a fractional order plus time delay model. J Process Control 2014;24:113–28.
[20] Hagglund T, Astrom KL. Revisiting the Ziegler-Nichols tuning rules for PI control. Asian J Control 2002;4(4):364–80.
[21] Astrom KL, Hagglund T. Revisiting the Ziegler-Nichols step response method for PID control. J Process Control 2004;(14):635–50.
[22] Murrill PW. Automatic Control of Processes. International Textbook Co.; 1967.
[23] Grimholt Chriss, Skogestad Sigurd. Optimal PI and PID control of first-order plus delay processes and evaluation of the original and improved SIMC rules. J Process Control 2018;70:36–46.