2012 IEEE International Conference on Control Applications (CCA)
Part of 2012 IEEE Multi-Conference on Systems and Control
October 3-5, 2012. Dubrovnik, Croatia

A Specifications based PID Autotuner

Robin De Keyser, Abhishek Dutta*, Andrés Hernandez and Clara M. Ionescu**, *Member IEEE*

Abstract— **This paper presents the development and validation of a specifications based PID autotuner. The performance of the autotuned PID is compared to that of a PID controller which is designed using a computer-assisted design tool assuming the availability of a full process model. An illustrative example in simulation of a high-order system, for which a simple PID controller may not be an optimal choice, proves that the proposed algorithm works well. Additionally, the algorithm has been tested on three real-life setups: a coupled water tank process, a poorly damped electromechanical system and a variable time delay thermal process. Again, the autotuned PID performed equally well as the model based PID.**

I. Introduction

Along the many decades in the history of control, the inventions based on feedback control had a crucial impact in the mechanical, scientific, electrical, aerospace, and information revolutions [1]. Experience has shown that progress in automatic control is reported more by solving concrete process-oriented control problems, rather than developing firstly the new control principles. It is hard to imagine the 21st century without the reverse parking aid system based on inner control loops of the steering wheel and gear-box in our zero emission cars. Despite the glorious and pioneering landmarks from the past, controller design is nowadays still an art, as much as a science. Tuning controllers for optimal closed loop performance depends heavily on the process to be controlled and identification is still a burden for the control engineer and remains a significant time-consuming task.

To simplify this task, PID controllers can incorporate *autotuning* capabilities, which reduce the start-up period [2]. The autotuners are equipped with a mechanism capable of automatically computing a reasonable set of parameters when the regulator is connected to the process. Autotuning is a very desirable feature and almost every industrial PID controller provides it nowadays. These features provide easy-to-use controller tuning and have proven to be well accepted among process engineers [3].

For the automatic tuning of the PID controllers, several methods have been proposed. Some of these methods are based on identification of one point of the process frequency response, while others are based on the knowledge of some characteristic parameters of the open-loop process step response. The identification of a point of the process frequency response can be performed either using a proportional regulator, which brings the closed-loop system to the stability boundary, or by using a relay feedback which forces the process output to oscillate [4],[5],[6],[7]. Usually these preliminary tests are necessary to determine a (partial) model for the process, along with the tuning of the controller parameters [8], [9]. Some other employed autotuning methods vary broadly from fuzzy inference [10], to neural networks [11], iterative feedback tuning [12], and robust control tuning [13].

This paper presents the development and validation of a novel specifications based PID autotuner. The original contribution of this paper is given in the second section and stands in the development of the algorithm. The third section presents briefly a computer aided design tool for calculating PID parameters based on the knowledge of the full process model. Next, the validation of the novel algorithm is performed in simulation on a high-order system (where a PID controller is not the obvious choice). In the fifth section we present the testing of the proposed autotuner versus the model based PID on 3 real-life setups: a coupled water tank process, a poorly damped electromechanical system and a variable time delay thermal process. Finally, a conclusion section summarizes the main outcome of this work.

II. Principle of a Specifications based PID Autotuner

The approximation of a closed-loop response by a *dominant 2nd order* transfer function with gain 1 gives the relationship between the closed-loop percent overshoot (%*OS*) and the peak magnitude M_p in frequency domain [14]:

$$\%OS = 100 e^{-\zeta\pi/\sqrt{1-\zeta^2}} \;,\quad M_p = \frac{1}{2\zeta\sqrt{1-\zeta^2}} \tag{1}$$

By *specifying* the allowed *overshoot* in the closed-loop, it follows that the closed-loop transfer function must fulfill the condition:

$$\max_{\omega}\left|T(j\omega)\right| = \max_{\omega}\left|\frac{G(j\omega)}{1+G(j\omega)}\right| = M_p \tag{2}$$

with $G(j\omega)$ the open loop transfer function (both the process and controller). Rewriting (2) as:

$$T(j\omega) = \frac{R(\omega) + jI(\omega)}{\left[1+R(\omega)\right] + jI(\omega)} \tag{3}$$

with R the real part and I the imaginary part of $G(j\omega)$, and taking $\left|T(j\omega)\right|^2$, it results that:

$$\left(R+c\right)^2 + I^2 = r^2 \tag{4}$$

Research supported by *IWT Vlaanderen and **FWO Vlaanderen.

All authors are with Ghent University, Department of Electrical energy, Systems and Automation, Technologiepark 913, B9052 Gent, Belgium (corresponding author R. De Keyser: 32-9-2645585; fax: 32-9-2645839; e-mail: Robain.DeKeyser@UGent.be).

where $c = \frac{M_p^2}{M_p^2 - 1}$, $r = \frac{M_p}{M_p^2 - 1}$. The equation (4) represents a (Hall-)circle with radius r and center in $\{-c, 0\}$. At this point the reader should remind that in the Nyquist plane, the specification (2) means that the Nyquist curve of the Controller*Process $G(j\omega)$ is tangent to the M-circle. The equivalent of the M-circle in the Nichols chart is a curve (the red curves in Figs 1 & 2), to which the Controller*Process curve should thus be tangent.

Our long-standing experience with a frequency response CAD tool (ref. section III) has learned us that a good closed-loop time response is achieved if the frequency response curve of Controller*Process is going smoothly around the M-curve at the intersection with the (horizontal) 0dB line in the Nichols chart. The 0dB line represents the unit circle in the Nyquist plot, hence the phase margin (*PM*) can be calculated. Intersection with the unit circle is achieved by adding the condition:

$$R^2 + I^2 = 1 \tag{5}$$

Solving for R and I in (4) and (5) yields:

$$R = 0.5\frac{1 - 2M_p^2}{M_p^2} \text{ and } I = -\frac{\sqrt{M_p^2 - 0.25}}{M_p^2} \tag{6}$$

and the phase margin can be calculated as:

$$PM = \tan^{-1}\frac{\sqrt{M_p^2 - 0.25}}{M_p^2 - 0.5} \tag{7}$$

It was earlier stated in [15] that specifying the *PM* is not sufficient to guarantee a good closed-loop performance in all cases. Therefore, the next step is to determine the cross-over frequency ω_c, which is the frequency where the Controller*Process frequency response should cross 0dB.

If the *settling time* T_s of the (dominant 2nd order) closed-loop is also *specified*, then similar derivations as for (1) lead to (ref. [14]):

$$\omega_{BW} = \omega_n\sqrt{\left(1 - \zeta^2\right) + \sqrt{4\zeta^4 - 4\zeta^2 + 2}} \tag{8}$$

with $$\omega_n = \frac{4}{\zeta T_s} \tag{9}$$

which allows us to calculate the bandwidth frequency ω_{BW}. This is the frequency where Controller*Process curve intersects the -3dB *closed loop* magnitude line (the green line in Figs 1 & 2), which occurs at about 7 to 8dB below the 0dB line. Root-locus insights show that a 2nd order system with gain 1 is the result of closing the loop around an integrator (with -90° phase lag) and a 1st order system (with -45° phase lag at its time constant). In order to have a *PM* of at least 45°, the cross-over frequency ω_c must be smaller than the frequency corresponding to this time-constant. The magnitude around ω_c then decreases between -20dB/dec (integrator) and -40dB/dec. To cover the -7.5dB between ω_c and ω_{BW} it follows that $\omega_{BW} \approx 2\omega_c$ and generalization to higher order systems (with steeper magnitude decrease) gives the rule $\omega_c \le \omega_{BW} \le 2\omega_c$.

From the knowledge of ω_c a sinusoid with period $T_c = \frac{\omega_c}{2\pi}$ is applied to the process and the output:

$$G(j\omega_c) = Me^{j\varphi} = M\left(\cos\varphi + j\sin\varphi\right) \tag{10}$$

can easily be found from the corresponding system response.

The task is now to find the controller parameters such that the specification for *PM* is fulfilled at ω_c, given the *specified* [$\%OS$, T_s] and the *measured* M and φ. Based on the relation:

$$R(j\omega_c)G(j\omega_c) = -[\ \cos PM + j\sin PM\ ] \tag{11}$$

and the PID controller given by:

$$R(s) = K_p(1 + \frac{1}{T_i s} + T_d s)$$
$$\text{or } R(j\omega_c) = K_p\left[1 + j\left(T_d\omega_c - \frac{1}{T_i\omega_c}\right)\right] = K_p(1 + j\alpha) \tag{12}$$

we obtain:

$$\alpha = \frac{\tan PM - \tan\varphi}{1 + \tan PM\tan\varphi} = \tan(PM - \varphi) = T_d\omega_c - \frac{1}{T_i\omega_c} \tag{13}$$

Specifying the usual PID relationship T_i=4T_d, (13) becomes:

$$T_d\omega_c - \frac{1}{4T_d\omega_c} = \tan(PM - \varphi) \tag{14}$$

from where

$$T_i = T_c\frac{\sin(PM - \varphi) \pm 1}{\pi\cos(PM - \varphi)} \tag{15}$$

which gives only one positive result. For the magnitude, from (11) and (10) we have that:

$$K_p M(\cos\varphi - \alpha\sin\varphi) = -\cos PM$$
$$K_p M(\sin\varphi + \alpha\cos\varphi) = -\sin PM \tag{16}$$

which gives the K_p controller parameter:

$$K_p = \pm\frac{\cos(PM - \varphi)}{M} \tag{17}$$

with only one positive result.

Summarizing algorithm:

1) With the overshoot specification %*OS*, calculate *PM* using (1)(7)

2) With the settling time specification T_s, calculate ω_c using (9)(8) and the relationship $\omega_c \le \omega_{BW} \le 2\omega_c$

3) Enter in the system a sine signal with period $Tc = \frac{2\pi}{\omega_c}$ and obtain (10)

4) Calculate [K_p T_i T_d] using (17)(15) and T_i=4T_d

III. A Model Based PID Tuning Method as a Performance Reference

It is always interesting to compare the performance of an autotuned PID controller to the performance of a PID controller which is designed *based on the full knowledge of the process model*. When a process model is available, the controller design can be done using computer aided design (CAD) tools.

In this paper, the CAD design uses the Frequency Response toolbox (FRtool) for Matlab® as described in [16]. Its user-friendly graphical interface is depicted in Figure 1. There is a separate window (not shown) allowing the design of the compensator by dragging compensator's poles and zeros with the mouse.

The interface offers the possibility to introduce design specifications (such as $\%OS$, T_s, and more …) as graphical restrictions on the Nichols plot – including a real-time update while dragging controller's poles and zeros.

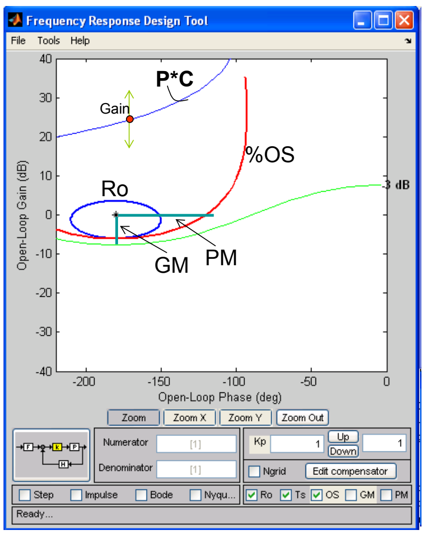

Fig. 1: Illustrative example of the graphical interface of FRtool. After the system has been imported from Matlab workspace, it appears as a curve in the Nichols chart corresponding to the loop frequency response of the process and controller (P*C).

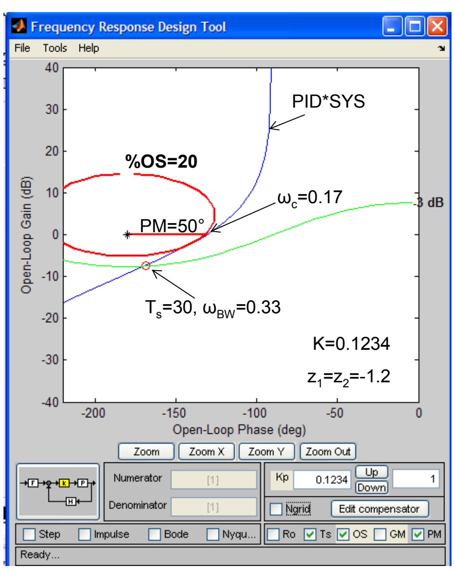

Fig. 2: The CAD- FRtool control design, based on the transfer function of the process $P_1(s)$

IV. An Illustrative Example

In order to validate the proposed autotuner, the process transfer function

$$P_1(s) = \frac{1}{(s+1)^6} \tag{18}$$

is considered, with specifications: $\%OS$=20 and T_s=30 seconds. For $\omega_c = 0.2$ rad/s it follows M=0.87, $\varphi = -71°$ and PM=45.6°. The controller parameters are then calculated using (15) and (17), with K_p=0.5, T_i=2.25 and T_d=0.56.

Figure 2 shows the design of the PID controller when using the full knowledge of the transfer function of the process, by means of FRtool. The corresponding PID parameters are: K_p=0.3; T_i=1.66 and T_d=0.41.

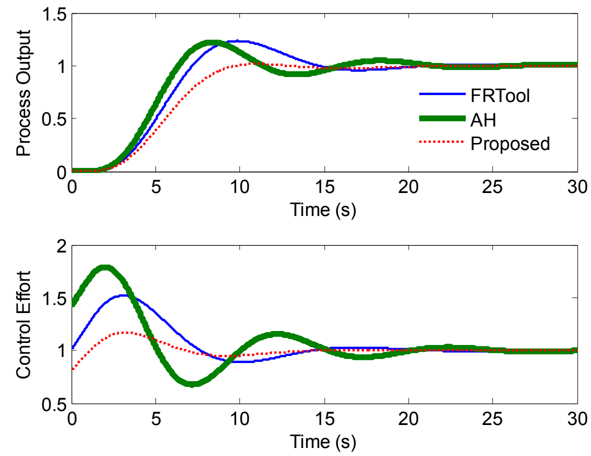

Fig. 3: Comparison between the FRtool based PID controller, the AH autotuner and the proposed specification based autotuner, for $P_1(s)$.

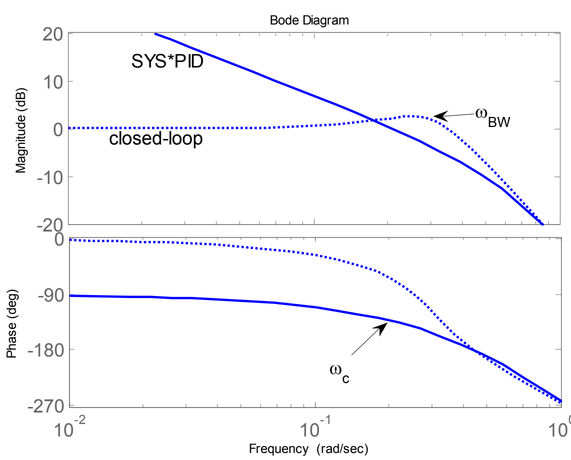

Fig. 4: Validation of controller in open loop (SYS*PID) and closed loop for the specified cross-over frequency (phase margin) and bandwidth frequency (settling time) for $P_1(s)$.

Additionally, the well-known Åström-Hägglund autotuner [2] is employed, giving the following parameters: K_p=1.41; T_i=5.45 and T_d=1.36. Figure 3 shows the comparison between the 'best' design possible (FRtool), and the two PID auto-tuners. Figure 4 validates the cross-over frequency of $\omega_c = 0.18$ rad/s and bandwidth frequency

$\omega_{BW} = 0.34$ rad/s, both close to specifications.

V. Results on Real Life Processes

A. Nonlinear 2-Coupled Water Tanks

The 2 coupled water tank from Quanser has been used as a real-life example, illustrated in figure 5. The setup of the two tanks corresponds to the configuration #1 from the Quanser manual (state coupled). The system is highly nonlinear in changes in the gain and in the time constant parameters. The control objective is to maintain the water level in the 2nd tank at a desired value. Disturbance is applied at the output of the first tank (volume loss).

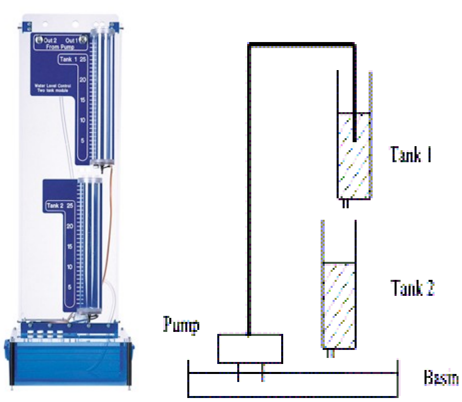

Fig. 5: Photo and schematic of the state-coupled water tanks.

For the specifications of no overshoot *%OS*<0.001 and T_s=100 seconds we have that $\omega_c = 0.0628$ rad/s and it follows *M*=1.66, $\varphi = -97°$ and *PM*=45.6°. The controller parameters K_p=0.44, T_i=38.9 and T_d=9.7. For the FRTool design, a transfer function has been identified using the prediction error method around the operating point of 15cm level in the second tank, as given by:

$$P_2(s) = \frac{3.8}{(18s+1)(18s+1)} \qquad (19)$$

The PID parameters tuned in FRTool for this transfer function are: K_p=0.38, T_i=25, T_d=9.37. The results on the real plant are shown in figure 6 below. Figure 7 validates the cross-over frequency of $\omega_c = 0.06$ rad/s and bandwidth frequency $\omega_{BW} = 0.1$ rad/s, both very close to specifications.

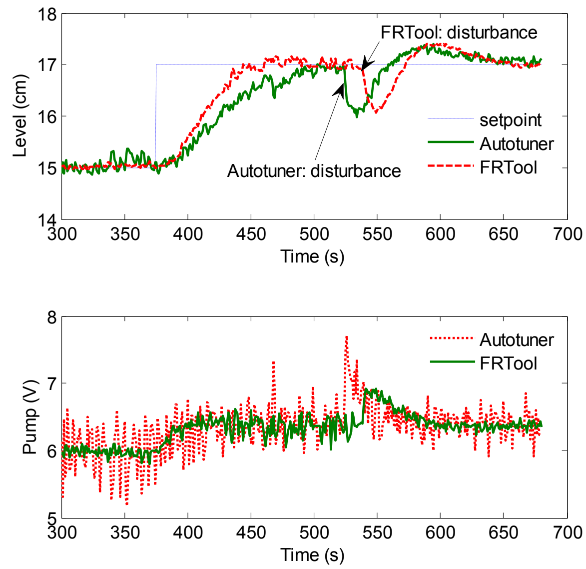

Fig. 6: Comparison between the FRtool based PID controller and the proposed autotuner with the same specifications for $P_2(s)$.

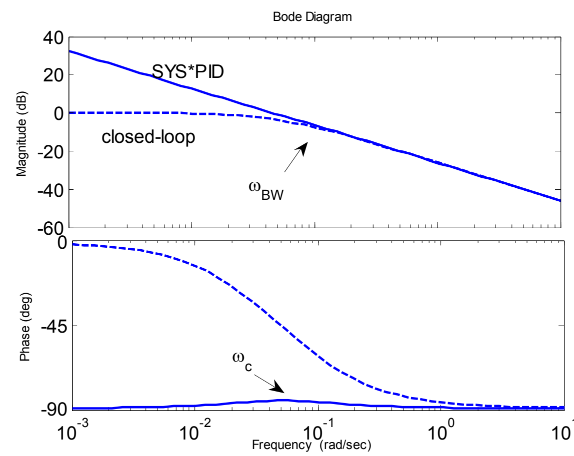

Fig. 7: Validation of controller in open loop (SYS*PID) and closed loop for the specified cross-over frequency (phase margin) and bandwidth frequency (settling time) for $P_2(s)$.

B. Poorly Damped Electromechanical System

Consider now a mass-spring-damper system driven by an electrical motor, with two masses, three springs and one damper as illustrated in figure 8 and given by the transfer function:

$$P_3(s) = \frac{800}{2.498s^4 + 16.65s^3 + 4473s^2 + 14400s + 1360000} \qquad (20)$$

The control objective is to maintain the position of the second mass at a desired setpoint. Disturbance is applied to the first mass. The input of the system is the voltage to the motor $u(t)$ and the outputs are the mass displacements $y_1(t)$ and $y_2(t)$ expressed in centimeters. Therefore a complete model of the electromechanical plant should describe the dynamics from $u(t)$ to $y_1(t)$ and from $u(t)$ to $y_2(t)$. The (fast) dynamics of the electrical motor can be neglected; hence, the motor can be represented by a pure static gain $F(t) = K \cdot u(t)$, with $F(t)$ the force on the 1st mass. The parameters of the set-up are: m_1=1.7 Kg, m_2=1.2 Kg, $k_1 = k_2$=800 N/m, k_3=450 N/m, c_1=9 N/(m/s).

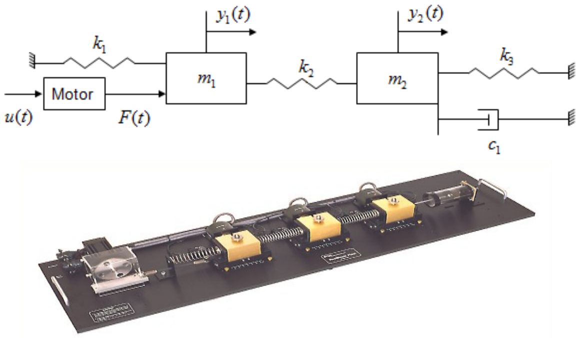

Fig. 8: Photo and schematic of the mass-spring-damper system. In our setup, the 3rd mass has been kept fixed and the damper has been connected to the 2nd mass

This system has two eigen-frequencies $\omega_1 = 20.8$ rad/s and $\omega_2 = 39.1$ rad/s, and damping factors $\zeta_1 = 0.08$ and $\zeta_1 = 0.05$. Using the CAD package, one obtains the 'best' possible PID controller Kp=41.76, Ti=0.023 and Td=0.0057.

For specifications as no overshoot %OS<0.001 and Ts=3 seconds, it follows that $\omega_c = 1.29$ rad/s and consequently $M = 0.0005$, $\varphi = -1°$ and PM=91°. The controller parameters are K_p=53.5, T_i=0.0245 and T_d=0.0061. It should be noted that for this system, the AH auto-tuner gives unstable results.

Figure 9 shows the performance of the proposed auto-tuner, which gives results comparable to those provided by the PID controller designed via the CAD-FRtool.

Figure 10 validates the cross-over frequency of $\omega_c = 1.3$ rad/s and bandwidth frequency $\omega_{BW} = 1.31$ rad/s, both close to specifications.

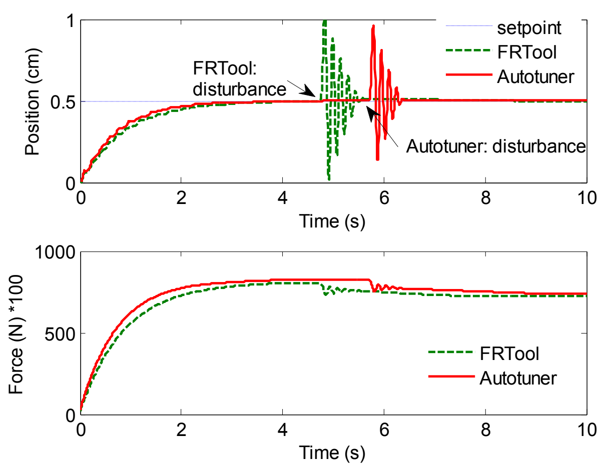

Fig. 9: Comparison between the FRtool based PID controller and the proposed autotuner with the same specifications for $P_3(s)$.

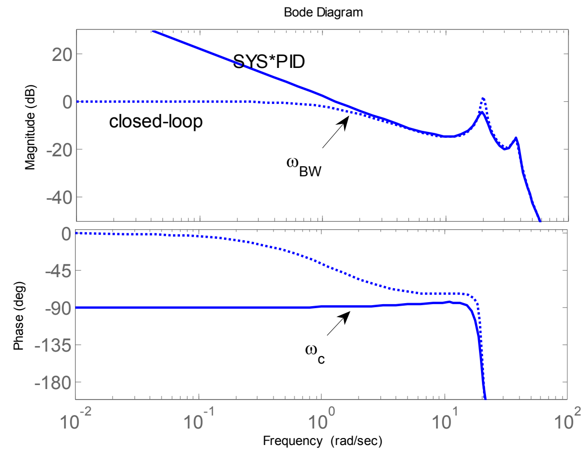

Fig. 10: Validation of controller in open loop (SYS*PID) and closed loop for the specified cross-over frequency (phase margin) and bandwidth frequency (settling time) for $P_3(s)$.

C. *Variable Time delay Thermal Plant*

Many processes include time delay phenomena in their inner dynamics, representative examples being found in biology, chemistry, mechanics, physics, population dynamics, as well as in engineering sciences. The presence of time delay (dead time) in the control loop is always a serious obstacle to good performance. Hence, it is interesting to test the performance of the proposed auto-tuner in a process with time delay. In fact, the auto-tuner will be tested on a system with variable time delay comparable to the time constant of the system.

The process consists of a heated tank of which the level is controlled by a mechanical float switch, thus resulting in a constant water volume as depicted in Figure 11.

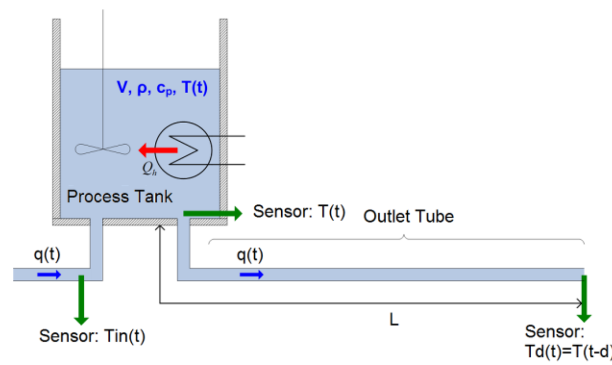

Fig. 11: Process with variable time delay: a heating tank system

A submerged electrical heater delivers a *constant* heat flow Q_h, which causes the liquid to warm up. Temperature control of the outlet water is achieved by changing the outflow of hot tank water, which allows an equal amount of cold tap water to flow in. A variable transport delay, which depends on the outflow *q(t)* - the manipulated variable - is introduced in the system by measuring the temperature of the effluent stream at a distance *L* from the tank. It is important to mention that the volume of the tube is a little larger than the volume of the tank. Therefore, the time delay is comparable to the time constant of the system.

For a chosen setpoint of 32ºC, the identified model of the process using prediction error method is given by the following transfer function:

$$P_4(s) = \frac{-1192}{(45\,s+1)(29\,s+1)} e^{-72s} \tag{21}$$

The closed loop specifications have been set as %OS<10 and Ts=300 seconds. The PID parameters tuned in FRTool for the transfer function (21) are: K_p=-0.00061, T_i=84.657, T_d=21.16. By applying the autotuner methodology and based on the specifications it follows that $\omega_c = 0.0105$ rad/s and consequently $M = 1031.7$, $\varphi = 101.8$ and PM=59.1°. The controller parameters are K_p=-0.0007, T_i=83.51 and T_d=20.88. The performance of both controllers in the real plant is presented in figure 12.

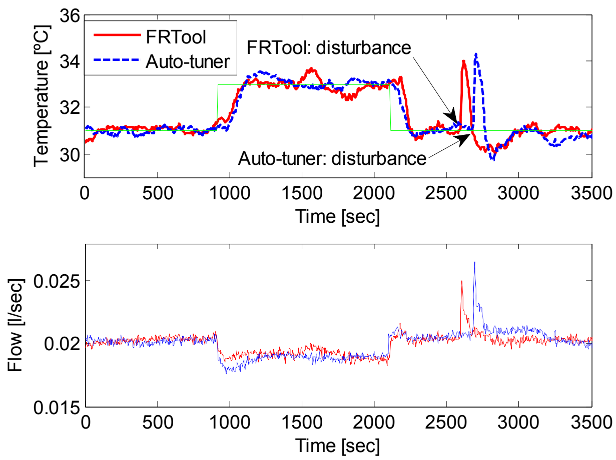

Fig. 12: Comparison between the FRtool based PID controller and the proposed autotuner with the same specifications for $P_4(s)$.

A validation is performed in Figure 13, by checking the cross-over frequency $\omega_c = 0.0105$ rad/s and bandwidth frequency $\omega_{BW} = 0.0262$ rad/s. The controllers fulfill the specifications.

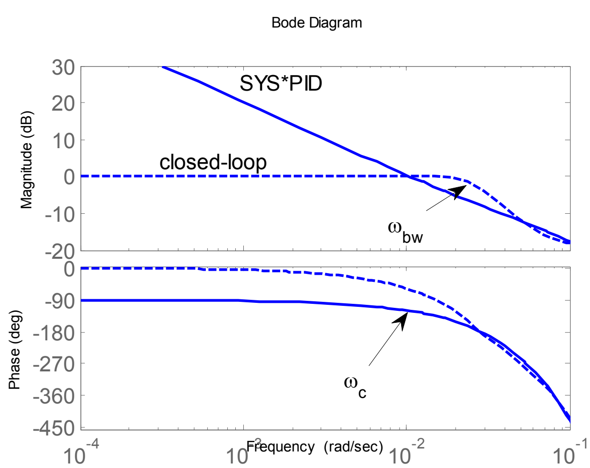

Fig: 13 Validation of controller in open loop (SYS*PID) and closed loop for the specified cross-over frequency (phase margin) and bandwidth frequency (settling time) for $P_4(s)$.

VI. CONCLUSIONS

The development and validation of a specifications based novel PID autotuner have been presented. The tuning of the PID parameters is based on overshoot and settling time specifications, using an experiment with a sinusoid as input to the process. The method is simple to implement in practice.

The effectiveness of the algorithm has been tested in simulation on a high-order process where a PID control is obviously not an optimal choice. Nevertheless, the results were as good as when a model-based PID tuning is employed.

Furthermore, 3 real-life setups have been used as practical examples where the proposed autotuning algorithm has delivered good results for both setpoint tracking and disturbance rejection.

REFERENCES

[1] D. Bernstein & L. Bushnell, The history of control: from idea to technology, *IEEE Control Systems Magazine, 22*(2), 2002, 21-23
[2] K. J., Åström & T. Hägglund. *PID Controllers: Theory, Design and Tuning*. Instrument Society of America, Research Triangle Park, NC, USA, 1995
[3] A., Leva, C. Cox & A. Ruano. Hands-on Pid autotuning: a guide to better utilisation. *IFAC Professional Brief*, 2002.
[4] K.J., Åström & T. Hägglund. Automatic tuning of simple regulators with specifications on phase and amplitude margins, *Automatica*, *20*, 1984, 645-651.
[5] C.C., Hang, K.J. Åström, & W.K., Ho, Refinements of the Ziegler-Nichols Tuning Formula. In: *IEE Proc. Design, Control Theory and Appl. 138*, 1991. pp. 111–118
[6] K.J., Åström, C.C. Hang, P. Persson & W.K. Ho. Towards intelligent PID control. *Automatica*, *28,* 1992, 1-9.
[7] J. G. Ziegler, & N.B., Nichols, Optimum settings for automatic controllers. *Transactions of ASME 64*, 1942, 759–768.
[8] T. S. Schei, Automatic tuning of PID controllers based on transfer function estimation. *Automatica*, *30,* 1994, 1983-1986.
[9] C. Scali, G. Marchetii, & D. Semino, Relay with additional delay for identification and autotuning of completely unknown processes, *Ind. Eng. Chem. Res.*, *38*, 1999, 1987-1997.
[10] R. Bandyopadhyay & D. Patranabis, A new autotuning algorithm for PID controllers using dead-beat format, *ISA Transactions,* 2001, 255-266.
[11] W. D. Chang, R. C. Hwang, & J. G. Hsieh, A multivariable on-line adaptive PID controller using auto-tuning neurons, *Engineering Applications of Artificial Intelligence, 16*, 2003, 57-63
[12] W. K. Ho, Y. Hong, A. Hansson, H. Hjalmarsson, & J. W. Deng, Relay auto-tuning of PID controllers using iterative feedback tuning, *Automatica, 39,* 2003, 149-157.
[13] C. Dey & R. Mudi, An improved auto-tuning scheme for PID controllers, *ISA Transactions, 48,* 2009, 396-409.
[14] N.S. Nise, *Control systems engineering* (4th ed., Wiley India Pvt. Ltd., 2007).
[15] C.M. Ionescu, R. De Keyser, The next generation of relay-based PID autotuners (PART1): some insights on the performance of simple relay-based PID autotuners. *PID'12 IFAC Conf. on Advances in PID Control*, Brescia, 28-30 March 2012.
[16] R. De Keyser & C.M. Ionescu, FRTool: a frequency response tool for CACSD in Matlab, in *Proc. of the IEEE Conf. on Computer Aided Control Systems Design*, Munich, Germany, 2006, 2275-2280.