TEACHING CONTROL WITH FIRST ORDER TIME DELAY MODEL AND PI CONTROLLERS

M. L. Ruz*, F. Morilla, F. Vázquez***

**Dpto de Informática y Análisis Numérico, Universidad de Córdoba, Campus de Rabanales 14071 Córdoba, Spain. (Tel: 34-957-218729; e-mails: p12rurum@uco.es, fvazquez@uco.es)*
***Dpto de Informática y Automática, UNED, C/. Juan del Rosal 16, 28040 Madrid, Spain (Tel: 34-91-3987156; e-mail: fmorilla@dia.uned.es)*

Abstract: This paper presents a Matlab GUI to teach basic control concepts based on a set of PI tuning rules for stable and unstable first order lag plus time delay models. The GUI shows the stabilizing region in the K_I-K_P parameter space and the control parameters obtained when applying a set of tuning rules, as well as the temporal response and the frequency response. The tool is prepared to work with two models, the nominal model and the simulation model, providing important information about robustness, the validity range of the tuning formulae and the performance that can be achieved with each one.

Keywords: PI control, education, robust control, tuning rules, processes with time delay

1. INTRODUCTION

There are important reasons to teach control engineering (at universities, at technical colleges, at industries) using first-order models with dead-time and PI controllers (Åström and Hägglund, 2005; Silva *et al.,* 2005; O'Dwyer, 2006):

- The popularity of PI controllers among industrial practitioners.
- PI controllers are adequate for processes where the dynamics are essentially of first order.
- The first order lag plus time delay (FOLPD) model captures many real-world practical and industrial situations.
- PI controllers remain poorly understood and, in particular, poorly tuned in many applications. It is clear that many controller tuning rules proposed in the literature are not having an impact on industrial practice.
- The most direct way to set up controller parameters is the use of tuning rules, generally based on stable and unstable FOLPD models.
- It is a challenge to allow students to gain perspective on the advantages and limitations inherent in the tuning rules.

There are also important reasons to teach control using interactive tools (Guzmán *et al*., 2008):

- The use of computer aided design and analysis tools is one way to increase learning efficiency.
- The idea to change properties and immediately being able to see the effects of the changes is very powerful both for learning and for designing.
- Interactive tools are recommended as support to consolidated theoretical books looking for the combination between theory and practice.

In this paper we present an interactive tool "TC_FOLPD_PI" intended to provide a set of PI tuning rules. These formulae can be compared in an interactively way by the user, allowing to understand concepts about stability, robustness and stability margins in PI control loop where both stable and unstable FOLPD models are considered. The tool will be available on the web at http://www.dia.uned.es/~fmorilla/ together with others interactive tools such as PID GUI (Morilla *et al*., 2006). It could be used in control lectures and it can be very useful for students to get insight into tuning PI controllers. In Section 2, the PI stabilization problem is revisited. A representative set of tuning rules for PI controllers is presented in Section 3. The graphic user interface developed to put in practice these subjects is described in Section 4. Some basic educational or training uses of the tool are mentioned in Section 5. Conclusions are presented in Section 5.

2. THE PI STABILIZATION PROBLEM

We consider the feedback control system shown in Figure 1. The plant G(s) is a first-order model with dead-time given by the following transfer function:

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

Where K represents the steady-state gain of the plant, τ the time delay, and T the time constant. The controller C(s) is of the PI type, i.e., it has a proportional term and an integral term:

$$C(s) = K_P + \frac{K_I}{s} = K_P\left(1 + \frac{1}{T_I\,s}\right) \tag{2}$$

K_P is the proportional gain, and K_I is the integral gain. Therefore the integral time constant is $T_I = K_P/K_I$.

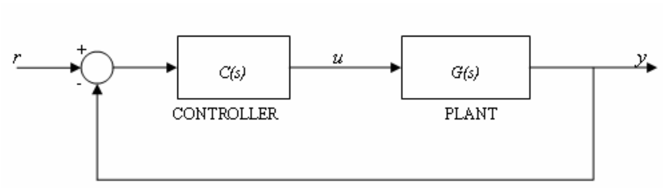

Fig. 1. Feedback control system

Our objective is to analytically determine the region in the K_I-K_P parameter space for which the closed-loop system is stable. We consider two different cases: open-loop stable plants (T>0) and open-loop unstable plants (T<0). The frequency response of the model is characterized by the magnitude and the phase:

$$r(\omega)=\frac{K}{\sqrt{1+\omega^2\ T^2}} \tag{3}$$

$$\phi(\omega)=\pi - \omega\ \tau - \text{atan}(\omega\ T) \tag{4}$$

Following the results of Silva *et al.*, (2005) and Morilla *et al.*, (2006), we know that the region is bounded by a curve ($K_P(\omega)$,$K_I(\omega)$) and the axis K_I=0. Figure 2 shows the stabilizing sets for an open-loop stable plant and for an open-loop unstable plant. Both plants have the same steady-state gain (K=1) and time delay (τ=0.2), but the time constants are of different sign (T=1 and T=-1):

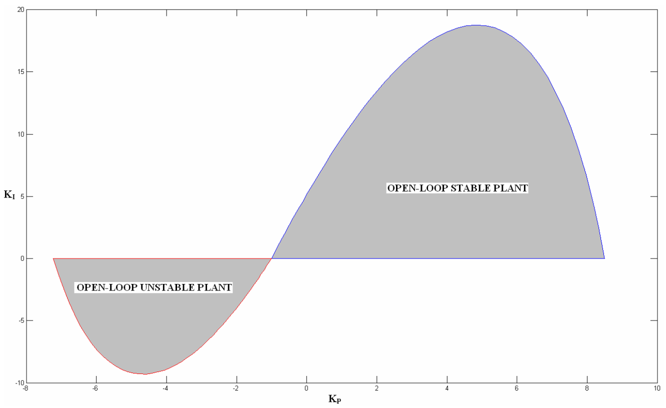

Fig. 2. Stabilizing PI regions for the open-loop unstable plant (K=1, T=-1, τ=0.2) and the open-loop stable plant (K=1, T=1, τ=0.2)

Because the efficiency is important for developing software design interactive packages, we only need to determine the boundary curve. We decide to use the following algorithm:

Step 1 (based on the results of Silva *et al.*, 2005): First we find the frequency ω_{max}, in the interval $\left(\frac{\pi}{2},\pi\right)$ when T>0 and in the interval $\left(0,\frac{\pi}{2}\right)$ when T<0 that verifies:

$$\phi(\omega_{max})=\pi \quad\Rightarrow\quad \omega_{max}\ \tau + \text{atan}(\omega_{max}\ T) = 0 \tag{5}$$

Step 2 (based on PI tuning by gain margin A_m=1, Morilla *et al.*, 2006): Now we determine the (K_P,K_I) values, sweeping the frequency range (0, ω_{max}), as follows:

$$K_P = \frac{\cos\ (\phi(\omega))}{r(\omega)} \tag{6}$$

$$K_I = \omega\ \frac{\text{sen}\ (\phi(\omega))}{r(\omega)} \tag{7}$$

This algorithm has been tested with a lot of stable and unstable cases, providing the same stabilizing sets than Silva *et al*. (2005) with less computational cost.

3. TUNING RULES FOR PI CONTROLLERS

The most direct way to set up controller parameters is the use of tuning rules. Although the classical tuning rules were proposed by Ziegler and Nichols in 1942, there are many tuning rules defined over the past decade, reflecting the upsurge of interest in the use of the PID controller recently. There are many well known PI tuning formulae for stable processes (O'Dwyer, 2006); however, PI tuning formulae for unstable processes are less common (Ho and Xu, 1998; Padma Sree *et al.*, 2004). The FOLPD model is generally used in both cases; stable and unstable.

To test this tool we have selected several representative PI tuning rules, shown in Tables 1 and 2 at the end of the paper, but others formulae can be easily added. The first column of these tables details the author of the rule. The second and third columns detail the formulae for the controller parameters, K_P and T_I respectively. The final column facilitates information about the tuning: performance criteria (quarter decay ratio, minimum integral error) or performance specification (gain margin, closed-loop time constant) and its validity range based on the model parameters or on some specification of the formula. Almost all these tuning rules are mentioned in the book of O'Dwyer (2006), but we have also selected others recent tuning rules such as AMIGO (Approximate M constrained Integral Gain Optimization by Aström and Hägglund, 2005) and SIMC ("Simple control" or "Skogestad IMC" by Skogestad, 2003).

The tuning rules enable us to analyse aspects well known, like these:

- The criterion "a quarter decay ratio" gives a small settling time.
- The tuning by minimum integral error gives different setting times for load disturbances and setpoint changes.
- The IMC tuning rule gives an excellent setpoint response but a slow settling time for a load disturbance. However, the SIMC tuning rule is presented to work well for both cases.
- AMIGO rule recommends a specified controller structure depending on the models parameters.
- The performance specifications that are normally obtained for stable models cannot be obtained for unstable systems. The overshoot and settling time are larger for unstable systems.
- PI controllers do not produce good results for the control of an unstable process.

- There are somewhat complex tuning rules, but some authors propose simple rules.

- Some settings using unstable model can make the closed loop system unstable even for a small change in the system parameters. It rarely happens with stable processes.

4. THE TC_FOLPD_PI GUI

A graphic user interface (GUI) has been developed in Matlab. The main window, shown in Figure 3, has been designed to provide simultaneously: the stabilizing region in the K_I-K_P parameter space, the model parameter space τ-T, and the time response simulation. The secondary windows, shown in Figure 4, with the frequency response features, the Nyquist plot and the signal control, only appear when the user requests them using the respective push buttons located on the centre of the main window. All windows are quickly updated when the user makes changes in the interface.

The TC_FOLPD_PI GUI is prepared to work with one or two models, the first one is the “nominal” model and the second model is the “simulation” model, the user can switch between these options by clicking the radio-button “Mismatch models” located on the bottom of the main window. The “simulation” model is used for analysis and simulation and the “nominal” model is used when the user requests the tuning rules, but both models are used to generate the corresponding stability regions. When the “Mismatch models” push button is unselected, the simulation model is the same that the nominal model, providing the ideal case. But when the “Mismatch models” button is selected, the user can choose different parameters for these models in order to make a robust analysis of the closed-loop system. This feature is one of the most valuable information this GUI provides. The three parameters (steady-state gain, time constant and time delay) of both models can be modified using the text fields located in the panel “model parameters”. Besides, the main parameters (time constant and time delay) can be jointly modified by dragging the corresponding point (black for the “nominal” model and red for the “simulation” model) in the model parameter space τ-T.

The TC_FOLPD_PI GUI is prepared to work with two kinds of PI structures (PI and I-P), in the second structure the reference is only introduced in the integral term. The values of the control parameters (K_P and K_I) are always visible in the two text fields located on the lower right corner, and they are also represented by a red point in the K_I-K_P parameter space. The user has three ways to change the control parameters: by editing the text fields, by dragging the red point in the K_I-K_P parameter space, or by clicking the button “tuning point”. In the last option, the control parameters are updated with the values shown in the text fields K_P and K_I of pink colour, given by the last tuning rule selection. In all cases temporal and frequency responses will be updated.

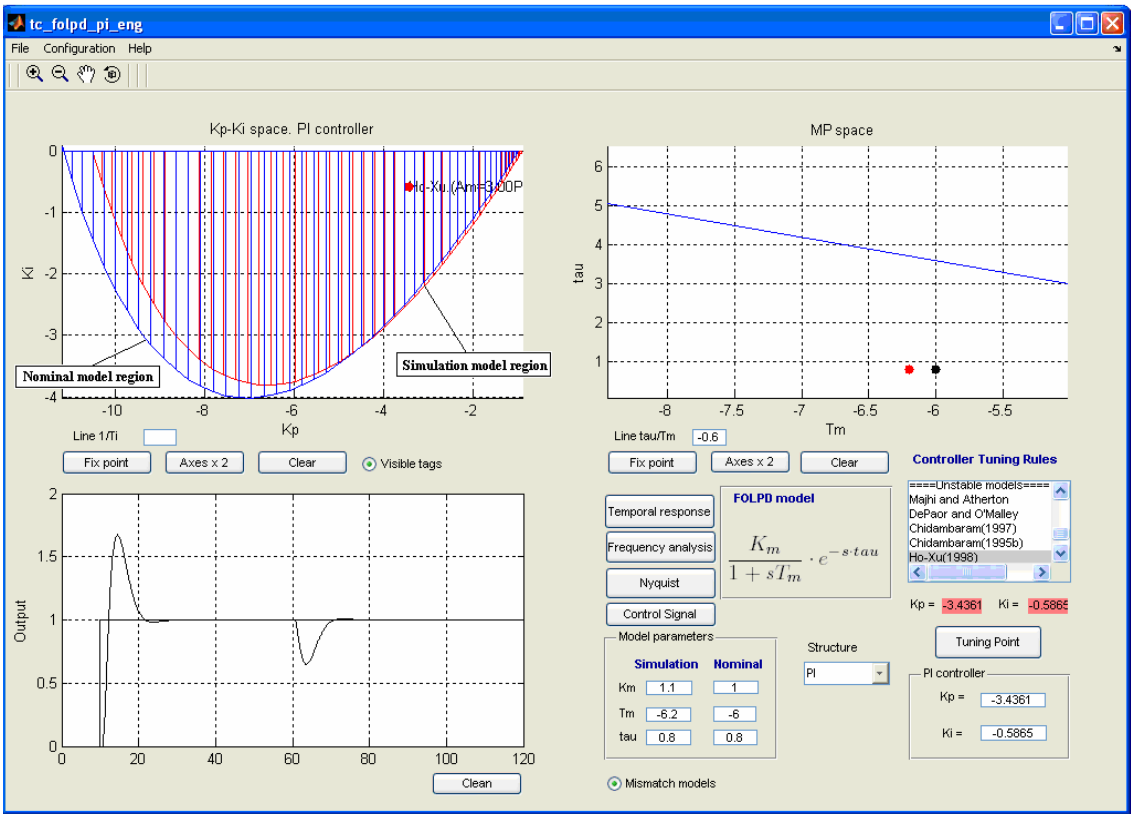

Fig. 3. Main window of the Matlab TC_FOLPD_PI GUI

A list-box containing the tuning rules is on the right of the main window. When the user selects a tuning rule, the GUI automatically generates (if the nominal model parameters meets the validity range of the formula) a point (or a set of points) in the K_I-K_P parameter space. Depending on the formula selected, a pop-up menu could appear requesting some specifications, for example the gain margin, the phase margin, or the closed-loop time constant. Therefore, the user can generate several sets of parameters using the same or different tuning rules. Each point generated will be tagged in the K_I-K_P parameter space, describing the name of the tuning rule and the specifications chosen (when the formula requests it); the user can also hide these labels. All these points are updated when the black point associated with the "nominal" model is released.

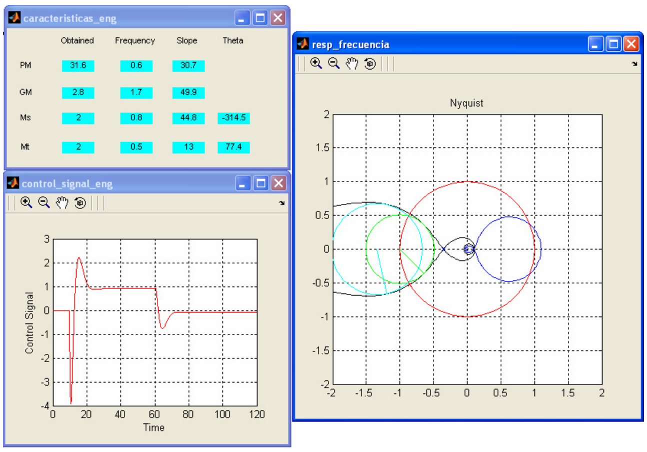

Fig. 4. Secondary windows of the TC_FOLPD_PI GUI

Several temporal responses can be displayed at the main window. This is useful for example in order to compare different tuning rules. The user can do it dragging and releasing the red point of the K_I-K_P parameter space to the specific points generated by the tuning rules.

In Figure 3 a nominal unstable FOLPD model is used, its parameters are K=1, T=-6 and τ=0.8 (Silva *et al.*, 2005). However, a deviation of K and T parameters is supposed, selecting K=1.1, T=-6.2 and τ=0.8 for the simulation model. Therefore, its stabilizing regions are lightly different. The tuning rule for PI controllers proposed by Ho and Xu (1998) has been applied, specifying a gain margin of 3 and a phase margin of 30º, the control parameters obtained are $K_P \cong -3.44$ and $K_I \cong -0.59$. In Figure 4 the frequency features with these specifications has been calculated. Note that the deviations of the parameters between simulation and nominal models produce a gain margin of 2.8 and phase margin of 31.6. Unselecting "mismatch models" we would obtain a gain margin of 3 and a phase margin of 30.9º. Then, in both cases the control parameters produce closed-loop stability and performance quite close to the specified one.

The TC_FOLPD_PI GUI has others interesting elements in order to facilitate students understanding of the tuning rules. An edit text field named "line tau/Tm" is below the model parameter space, it allows to plot lines with different slopes τ/T, serving as reference to test the tuning rules and their validity ranges. Also, an edit text field named "line 1/Ti" is below the K_I-K_P parameter space. It allows to plot lines with different slopes K_I/K_P in the K_I-K_P parameter space and it can be used to analyse pole cancellation methodologies.

There are other push buttons which can clear the axes, clean the time response or fix points in both parameter spaces, making possible to have a reference of models or controller parameters previously used. Also, the toolbar of the main window includes a menu, where the user can load and save sessions, export a controller or a process to the Matlab workspace, to choice between two languages (English or Spanish), to set the simulation parameters and several properties of the axes associated with K_I-K_P and model parameter spaces, as well as some functions of Matlab, like zoom or panning.

5. EDUCATIONAL OR TRAINING USES

The tool can have many educational or training uses, here are some basic uses:

1) **Tuning by trial and error inside the stabilizing region**. The tool presents the stabilizing region and the user can test it moving the red point in the K_I-K_P parameter space. Releasing the red point in different zones, the user can get enough information to select the control parameters.
2) **Comparing tuning PI rules**. The tool offers a great variety of tuning rules, so the user can easily compare the settings obtained with different criteria or specifications.
3) **Analyzing tuning rules characteristics**. Almost all the tuning rules set the proportional gain in function of the ratio τ/T. Therefore, by choosing a line "tau/Tm" in the model parameter space and moving the "nominal" model above it, the user can observe that the proportional gain remains constant and only the integral gain is changing. Besides, almost all the tuning rules have a validity range based on the ratio τ/T. By choosing one or two lines "tau/Tm" the user can know the zone of the model parameter space where the tuning rule is applicable.
4) **Testing the limitations of PI controllers**. PI controllers have important limitations mainly with unstable FOLPD models. For example, by moving the red point inside the stabilizing region of an unstable FOLPD model, the user can test that it is not possible to get large phase and gain margins.
5) **Analyzing PI controllers designed by pole cancellation**. Some tuning rules make that the controller zero cancels the model pole doing T_I=T. Therefore, by choosing a line "1/Ti" in the K_I-K_P parameter space and moving the red point above it, the user can analyze PI controllers designed by pole cancellation; the integral constant remain constant and only the proportional gain is changing.
6) **Making sensitivity or robust analysis of the closed-loop system**. The tool is prepared to work with the "nominal" model and the "simulation" model simultaneously. Then, in the "mismatch models" mode the user can make perturbations in the model parameter space and watch the stabilizing regions, the temporal response or the frequency response features in order to make sensitivity or robust analysis of the closed-loop system.

7) **The effects of the model reduction techniques**. The TC_FOLPD_PI tool only accepts FOLPD models, but the PID GUI tool accepts models of any order without delay time. Then, starting from a complex model, by approximating to a FOLPD model and combining these interactive tools, the user can compare the stabilizing PI regions or the PI settings in order to evaluate the model reduction advantages or disadvantages.

6. CONCLUSIONS

A Matlab GUI named "TC_FOLPD_PI" has been presented in this paper. It is an interactive tool with many educational or training uses based on the PI tuning rules for stable and unstable FOLPD (first order lag plus time delay) models. The tool uses an efficient algorithm to compute the stabilizing PI region, providing the same stabilizing sets than Silva *et al.* (2005) with less computational cost. The tool include several representative PI tuning rules, but others formulae (O'Dwyer, 2006) can be easily added. TC_FOLPD_PI can be a good complement to others interactive tools, but also the beginning of another more complex tool with PID controllers.

ACKNOWLEDGEMENTS

This work has been supported by the Spanish CICYT under grant DPI 2007-62052. This support is very gratefully acknowledged.

REFERENCES

Åström, K.J. and T. Hägglund (2005). *Advanced PID control*. ISA-The Instrumentation, Systems, and Automation Society. Research Triangle Park, NC 27709.

Guzmán, J.L., K.J. Aström, S. Dormido, T. Hägglund, M. Berenguel and Y. Piguet (2008). Interactive learning modules for PID control. *IEEE Control System Magazine*, **28**, 118–134.

Ho, W.K. and W. Xu (1998). PID tuning for unstable processes based on gain and phase-margin specifications. *IEE Proc.-Control Theory Appl.*, **145**, 392-396.

Morilla, F., F. Vázquez and R. Hernández. (2006). PID control design with guaranteed stability. *7th IFAC Symposium on Advances in Control Education (ACE'06).* Madrid (Spain).

O'Dwyer, A. (2006). *Handbook of PI and PID Controller Tuning Rules*. 2nd Edition, Imperial College Press.

Padma Sree, R., M.N. Srinivas and M. Chidambaram (2004). A simple tuning method of tuning PID controllers for stable and unstable FOPTD systems. *Computers and Chemical Engineering,* **28**, 2201-2218.

Silva, G.J., A. Datta and S.P. Bhattacharyya (2005). PID Controllers for Time-Delay Systems. Birkhäuser.

Skogestad, S. (2003). Simple analytic rules for model reduction and PID controller tuning. *Journal Process Control,* **13**, 291-309.

Appendix A. REPRESENTATIVE PI TUNING RULES

Table 1. PI tuning rules for unstable FOLPD models

Rule	K_P	T_I	Comment
Mahji and Atherton (2000)	$\frac{1}{K}\left(0.889+\frac{e^{-\tau/T}-0.064}{e^{\tau/T}-0.990}\right)$	$\frac{2.6316T\left(e^{\tau/T}-0.966\right)}{e^{-\tau/T}-0.377}$	ISTE optimisation criterion $0<\frac{\tau}{T}<0.693$
De Paor and O'Malley (1989)	$\frac{1}{K}\left(\cos\sqrt{\left(1-\frac{\tau}{T}\right)\frac{\tau}{T}}+\sqrt{\frac{T}{\tau}\left(1-\frac{\tau}{T}\right)}\sin\sqrt{\left(1-\frac{\tau}{T}\right)\frac{\tau}{T}}\right)$	$\frac{T}{\left[\sqrt{\frac{T}{\tau}\left(1-\frac{\tau}{T}\right)}\right]\tan(0.5\varphi)}$ $\varphi=\tan^{-1}\sqrt{\frac{T}{\tau}\left(1-\frac{\tau}{T}\right)}-\sqrt{\left(1-\frac{\tau}{T}\right)\frac{\tau}{T}}$	Gain margin $A_m = 2$ $\frac{\tau}{T}<1$
Venkatashankar and Chidambaram (1994)	$\frac{1}{K}\sqrt{0.98\sqrt{1+\frac{0.04T^2}{(T-\tau)^2}}\left(\frac{25}{\tau}\right)\beta(T-\tau)\sqrt{\frac{1+\frac{\beta^2T^2}{\tau^2}}{1+\beta^2\frac{625}{\tau^2}(T-\tau)^2}}}$	$25(T-\tau)$	β=1.373 when $\frac{\tau}{T}<0.25$ β=0.953 when $0.25\le\frac{\tau}{T}<0.67$
Chidambaram (1995)	$\frac{1}{K}\left(1+0.26\frac{\tau}{T}\right)$	$25T-27\tau$	$\frac{\tau}{T}<0.6$
Chidambaram (1997)	$\frac{1.678}{K}\ln\left(\frac{T}{\tau}\right)$	$0.4015Te^{5.8\tau/T}$	
Ho and Xu (1998)	$\frac{\omega_p T}{A_m K}$ $\omega_p=\frac{A_m\Phi_m+0.5\pi A_m(A_m-1)}{(A_m^2-1)\tau}$	$\frac{1}{1.57\omega_p-\omega_p^2\tau-\frac{1}{T}}$	Φ_m : Phase margin A_m : Gain margin $\frac{\tau}{T}<0.62$

Table 2. PI tuning rules for stable FOLPD models

Rule	K_P	T_I	Comment
Ziegler and Nichols (1942)	$\frac{0.9\, T}{K\, \tau}$	$\frac{\tau}{0.3}$	Quarter decay ratio, $0.1 \le \frac{\tau}{T} \le 1$
AMIGO Aström and Hägglund (2005)	$\frac{0.15}{K} + \left(0.35 - \frac{\tau\, T}{(\tau + T)^2}\right) \frac{T}{K\, \tau}$	$0.35\, \tau + \frac{13\, \tau\, T^2}{T^2 + 12\, \tau\, T + 7\, \tau^2}$	I-P structure when $\frac{\tau}{\tau + T} \le 0.5$ otherwise PI structure
MISE Murrill (1967)	$\frac{1.305}{K} \left(\frac{T}{\tau}\right)^{0.959}$	$\frac{T}{0.492} \left(\frac{\tau}{T}\right)^{0.739}$	Regulator tuning by minimum integral error $0.1 \le \frac{\tau}{T} \le 1$
MIAE Murrill (1967)	$\frac{0.984}{K} \left(\frac{T}{\tau}\right)^{0.986}$	$\frac{T}{0.608} \left(\frac{\tau}{T}\right)^{0.707}$	
MITAE Murrill (1967)	$\frac{1.305}{K} \left(\frac{T}{\tau}\right)^{0.959}$	$\frac{T}{0.492} \left(\frac{\tau}{T}\right)^{0.739}$	
MIAE Rovira et al. (1969)	$\frac{0.758}{K} \left(\frac{T}{\tau}\right)^{0.861}$	$\frac{T}{1.020 - 0.323 \frac{\tau}{T}}$	Servo tuning by minimum integral error $0.1 \le \frac{\tau}{T} \le 1$
MITAE Rovira et al. (1969)	$\frac{0.586}{K} \left(\frac{T}{\tau}\right)^{0.916}$	$\frac{T}{1.030 - 0.165 \frac{\tau}{T}}$	
Cohen and Coon (1953)	$\frac{1}{K} \left(0.9 \frac{T}{\tau} + 0.083\right)$	$T \left(\frac{3.33 \frac{\tau}{T} + 0.31 \left(\frac{\tau}{T}\right)^2}{1 + 2.22 \frac{\tau}{T}}\right)$	Quarter decay ratio, $0 < \frac{\tau}{T} \le 1$
St. Clair (1997)	$\frac{0.333\, T}{K\, \tau}$	T	$\frac{\tau}{T} \ge 0.333$
O'Dwyer (2001)	$\frac{\pi\, T}{2\, A_m\, K\, \tau}$	T	A_m : Gain margin
Skogestad (2003)	$\frac{T}{K\, (T_c + \tau)}$	$\min(T, 4(T_c + \tau))$	T_c : Closed-loop time constant $T_c \le (T + \tau)$
IMC Rivera et al. (1986)	$\frac{\left(T + \frac{\tau}{2}\right)}{K\, T_c}$	$T + \frac{\tau}{2}$	T_c : Closed-loop time constant $1.7\, \tau \le T_c \le (T + \tau)$