**Preprints, 4th IFAC Conference on
Advances in Proportional-Integral-Derivative Control
Almería, Spain | June 12-14, 2024**

IMC-based tuning of PIDA controllers: a comparison with PID control ⋆

Antonio Visioli * **José Sánchez-Moreno** **

* *Dipartimento di Ingegneria Meccanica e Industriale, University of Brescia, Italy*
e-mail: antonio.visioli@unibs.it
** *Departamento de Informática y Automática, UNED, Madrid, Spain*
e-mail: jsanchez@dia.uned.es

Abstract: In this paper we evaluate the performance achieved by a tuning methodology based on Internal Model Control for a Proportional-Integral-Derivative-Acceleration (PIDA) controller. In particular, we compare the results obtained for the set-point and load disturbance step responses with those achieved with a Proportional-Integral-Derivative (PID) controller tuned by applying the well-known SIMC and AMIGO tuning rules. Different high-order processes are considered: self-regulating, distributed lag and non self-regulating. It is shown that, in general, the use of the double derivative (acceleration) action allows the integrated absolute error to be decreased without a decrement of the robustness and with a moderate increment of the control effort.

Keywords: PIDA controllers, PID controllers, Tuning, Internal Model Control.

1. INTRODUCTION

The great success of Proportional-Integral-Derivative (PID) controllers in industry is motivated by their relative simplicity and, at the same time, by the satisfactory performance they are able to provide for a wide variety of applications. Further, the availability of many effective tuning rules (O'Dwyer, 2006), of automatic tuning procedures and of well-established additional functionalities (Visioli, 2006) makes their design easier and, therefore, their overall cost/benefit ratio is difficult to improve by other advanced controllers.

However, it is also recognized that, because of their simple structures, PID controllers might fail to achieve the required performance when there are tight control requirements in a given application and the process has a high-order dynamics. For this reason, there is an interest in investigating controllers with a (slightly) more complex structure in order to deal with higher-order dynamics more effectively, but with the same design simplicity in order to keep the advantageous cost/benefit ratio. In this context, the first choice is to add to the PID controller a second derivative control action, yielding a Proportional-Integral-Derivative-Acceleration (PIDA) controller, also known as Proportional-Integral-Double-Derivative controller (PIDD or PIDD2) (Huba and Vrancic, 2018; Huba, 2019; Kumar and Hote, 2018, 2019; Huba et al., 2021). In fact, the PIDA controller has three zeros instead of two and therefore it can provide a better performance for high-order process as it allows an increment of the control system bandwidth without a decrement of the robustness (for example, in terms of the maximum sensitivity) (Milanesi et al., 2022).

With the aim of keeping the design of a PIDA controller as simple as that of a PID controller, an automatic tuning methodology for this controller has been proposed in (Visioli and Sánchez-Moreno, 2024). It is based on the estimation of a high-order process by means of the n-shifting technique proposed in (Sánchez et al., 2021), which exploits a relay-feedback experiment (Yu, 1999; Wang et al., 2003; Liu and Gao, 2011; Liu et al., 2013; Chidambaram and Sathe, 2014). Then, the PIDA parameters are determined by applying an Internal Model Control (IMC) strategy and by reducing the resulting (high-order) controllers to a PIDA controller by suitably truncating its Mclaurin series expansion. The method has then been extended to integral (non self-regulating) processes in (Visioli and Sánchez-Moreno, 2023), where both cases of PIDA and PDA controllers have been addressed.

In this paper we compare the performance achieved with this methodology with those obtained by a PID controller tuned with well-known effective rules, namely, the SIMC (Skogestad, 2003) and AMIGO (Åström and Hägglund, 2002, 2004) ones. The aim is to better evaluate the advantages provided by adding the acceleration action with respect to classic PID methodologies. In the performed analysis, different (high-order) processes, which are representative of typical industrial applications, are considered. In particular, a dead-time dominant self-regulating process with multiple lags, a distributed-lag process, and a non self-regulating (integral) process with multiple lags are selected. Further, measurement noise is taken into account because this might yield a serious detrimental effect on the control variable when a (double) derivative computation is applied.

The paper is organized as follows. The PIDA tuning procedure is briefly reviewed in Section 2, together with the SIMC and AMIGO tuning rules. The simulation results that have been obtained for the different considered processes are presented in Section 3 and discussed in Section 4. Conclusions are given in Section 5.

⋆ This work was supported in part by the Spanish State Research Agency (AEI) under the Project PID2020-112658RBI00/AEI/10.13039/501100011033

2. PIDA TUNING

The standard unity-feedback control scheme of Figure 1 is considered, where P is the process, C is the controller, y_{sp} is the set-point signal, d is the load disturbance signal, y is the process variable, n is the measurement noise signal, y_n is the feedback signal, e is the control error, and u is the control variable.
The process model can be expressed as

$$P(s) = p_m(s)e^{-Ls}, \tag{1}$$

where L is the (apparent) dead time and $p_m(s)$ is the minimum-phase part, which can be expressed as

$$p_m(s) = \frac{K}{a_n s^n + a_{n-1}s^{n-1} + \cdots + 1} \tag{2}$$

if the process is self-regulating (namely, it is asymptotically stable) and

$$p_m(s) = \frac{K}{s(a_n s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + 1)} \tag{3}$$

if the process is non self-regulating (namely, it is an integral process). The PIDA controller transfer function is expressed as

$$C(s) = K_p\left(1 + \frac{1}{T_i s} + \frac{T_d s}{\frac{T_d}{N}s+1} + \frac{T_a s^2}{\left(\frac{T_a}{M}s+1\right)^2}\right), \tag{4}$$

where K_p is the proportional gain, T_i is the integral time constant, T_d is the derivative time constant, and T_a is the acceleration time constant. The parameters M and N of the low-pass filters are essential to have a proper transfer function and to appropriately filter the measurement noise.
The design of the controller can be done by applying an IMC-based technique, that is, by selecting the desired closed-loop transfer function as

$$F(s) = \frac{Y(s)}{R(s)} = \frac{e^{-Ls}}{(\lambda s+1)^r} \tag{5}$$

for a self-regulating process and as

$$F(s) = \frac{Y(s)}{R(s)} = \frac{(L+\lambda r)s+1}{(\lambda s+1)^r}e^{-Ls} \tag{6}$$

for a non self-regulating process. Based on (5), the controller transfer function can then be calculated as determined as

$$C(s) = \frac{p_m^{-1}(s)}{(\lambda s+1)^r - e^{-Ls}} \tag{7}$$

while, based on (6), the controller transfer function is calculated as

$$C(s) = \frac{((r\lambda+L)s+1)p_m^{-1}(s)}{(\lambda s+1)^r - ((r\lambda+L)s+1)e^{-Ls}}. \tag{8}$$

In both cases, the resulting controller transfer function can be expanded as a truncated Maclaurin series:

$$C(s) = \frac{f(s)}{s} \cong \frac{1}{s}\left[f(0) + f^{(1)}(0)s + \frac{f^{(2)}(0)}{2}s^2 + \frac{f^{(3)}(0)}{6}s^3 + \frac{f^{(4)}(0)}{24}s^4\right]. \tag{9}$$

On the other side, by selecting $M = N$ in (4), the controller transfer function can also be expressed as

$$C(s) \cong \frac{1}{s}\left[\frac{K_p}{T_i} + K_p s + K_p T_d s^2 + K_p\left(T_a - \frac{T_d^2}{N}\right)s^3 - K_p\left(\frac{2T_a^2}{N} - \frac{T_d^3}{N^2}\right)s^4\right] \tag{10}$$

so that the PIDA parameters can be determined by equating (10) and (9).

The choice $M = N$ allow us, if a fourth-order process model is selected (that is, $n = 4$ in (2) and $n = 3$ in (3)), to obtain a closed-form expression of the controller parameters. In other words, tuning rules are available. This allows the user to easily select λ in a range for which all the controller parameters are real and positive (see subsection 3.3), by taking into account that λ is the closed-loop time constant and determines the trade-off between speed of response and robustness (and control effort).
The (high-order) process model can be estimated by applying advanced identification techniques. However, in industrial settings, it is more convenient to perform simple experiments and, in this context, a suitable technique is the so-called n-shifting technique (Sánchez et al., 2021), which exploits a relay-feedback experiment.
In this paper, the PIDA controller is compared with a PID controller tuned with the well-known SIMC and AMIGO rules. The former are based on an IMC approach, while the latter are based on an optimization approach. In case the SIMC tuning rules are employed, the PID controller transfer function is written in series form as

$$C(s) = K_p \frac{T_i s+1}{T_i s}\frac{T_d s+1}{\frac{T_d}{N}s+1}. \tag{11}$$

Then, for self-regulating processes, the process model is a second-order-plus-dead-time (SOPDT) transfer function

$$P(s) = \frac{K}{(T_1 s+1)(T_2 s+1)}e^{-Ls}, \tag{12}$$

which can be obtained, starting from a high-order process model, through the so-called "half rule" (Skogestad, 2003). Then, the PID parameters are determined as

$$K_p = \frac{1}{2KL} \quad T_i = \min\{T_1, 8L\} \quad T_d = T_2. \tag{13}$$

The filter parameter N is then set to 10, as it is typically done in industrial practice (Visioli, 2006). For non self-regulating processes, the process model is

$$P(s) = \frac{K}{s(Ts+1)}e^{-Ls} \tag{14}$$

and the tuning rules are

$$K_p = \frac{1}{2KL} \qquad T_i = 8L \qquad T_d = T \tag{15}$$

Regarding the AMIGO rules, they are applied to a PID controller in ideal form, that is, (4) with $T_a = 0$ and they consider a first-order-plus-dead-time (FOPDT) process model (which can be obtained, for example, by applying the tangent method to the open-loop step response (Visioli, 2006)), that is,

$$P(s) = \frac{K}{Ts+1}e^{-Ls} \tag{16}$$

for a self-regulating process and

$$P(s) = \frac{K}{s}e^{-Ls} \tag{17}$$

if process is non self-regulating. The tuning rules are

$$K_p = \frac{1}{K}\left(0.2 + 0.45\frac{T}{L}\right) \quad T_i = \frac{0.4L+0.8T}{L+0.1T}L \quad T_d = \frac{0.5LT}{0.3L+T} \tag{18}$$

for self-regulating processes and

$$K_p = \frac{0.35}{KL} \qquad T_i = 7L \qquad T_d = 0 \tag{19}$$

for non self-regulating processes (note that this is actually a PI controller).

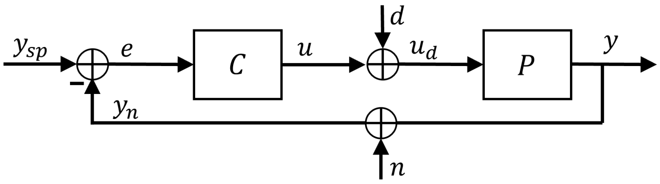

Fig. 1. The considered control scheme.

3. RESULTS

Simulation results related to different processes are shown hereafter. In all the cases, both a set-point unit step signal r and a unit step load disturbance signal d are applied. Then, it has to be remarked that, in all the simulations, measurement noise has been added by applying a random signal n whose values are in the range $[-0.1, 0.1]$. In order to avoid an excessive amplification of the noise in the control signal because of the double derivative (acceleration) action, a first-order low-pass filter

$$H(s) = \frac{1}{T_f s + 1} \tag{20}$$

has been placed in series with the PIDA controller. The value of T_f can be calculated straightforwardly by dividing by 10 the inverse of the gain crossover frequency of the loop transfer function $C(s)P(s)$. In this way the measurement noise is effectively filtered without significantly affecting the dynamics of the control system.

Finally, in order to evaluate the performance of the controller, two indices are calculated. The first one is the integrated absolute error defined as

$$IAE = \int_0^\infty |e(t)|dt, \tag{21}$$

while the second one, which is a robustness measure, is the maximum sensitivity, defined as

$$M_s = \max_{\omega \in [0,+\infty)} \left| \frac{1}{1 + C(j\omega)P(j\omega)} \right|. \tag{22}$$

3.1 Self-regulating multiple lag process

As a first example, we consider a self-regulating multiple lag process

$$P_1(s) = \frac{1}{(s+1)^8}. \tag{23}$$

If we design the PIDA controller, by applying the n-shifting identification procedure, we obtain the following process model:

$$\tilde{P}_1(s) = \frac{1}{14.52s^4 + 21.96s^3 + 17.17s^2 + 6.5s + 1} e^{-1.5s}. \tag{24}$$

If we select $r = 6$ and $\lambda = 0.675$, the tuning procedure yields $K_p = 0.985$, $T_i = 5.473$, $T_d = 1.889$, $T_a = 1.667$, and $N = M = 3.968$. The resulting value of the low-pass filter time constant is then $T_f = 0.186$. The corresponding value of the maximum sensitivity is $M_s = 1.82$. If we design a more robust controller by increasing the value of λ to 0.9, we have $K_p = 0.710$, $T_i = 4.904$, $T_d = 1.458$, $T_a = 1.419$, $N = M = 1.217$, and $T_f = 0.224$, which yields $M_s = 1.76$.

Regarding the PID tuned with the SIMC tuning rules, by applying the half rule to the nominal process model (although this is rather unrealistic, we assume to know the nominal model in order to evaluate the full potentialities of the SIMC rules), we obtain

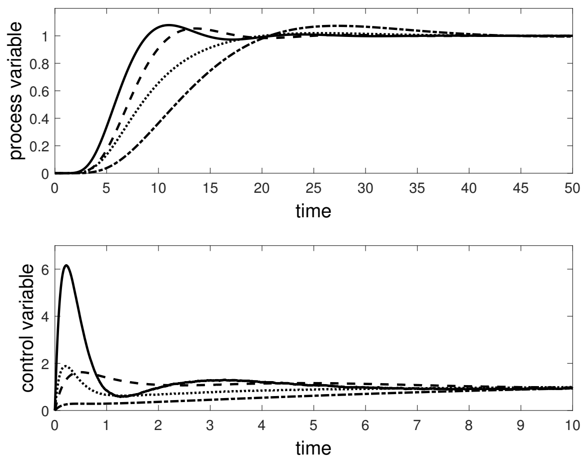

Fig. 2. Set-point step response for process P_1. Solid line: PIDA controller with $\lambda = 0.675$. Dashed line: PIDA controller with $\lambda = 0.9$. Dash-dot line: PID controller tuned with the SIMC tuning rules. Dashed line: PID controller tuned with the AMIGO tuning rules.

$$\tilde{P}_1(s) = \frac{1}{(s+1)(1.5s+1)} e^{-5.5s}. \tag{25}$$

By applying the tuning rules (13), the PID (in series form) parameters results $K_p = 0.091$, $T_i = 1$, $T_d = 1.5$ ($N = 10$), which yields $M_s = 1.81$.

Finally, regarding the AMIGO tuning rules, by applying the tangent method to the process open-loop step response, we obtain

$$\tilde{P}_1(s) = \frac{1}{3.035s + 1} e^{-4.966s} \tag{26}$$

(note that it clearly appears that the process is dead time dominant). The tuning rules (18) yields $K_p = 0.475$, $T_i = 4.16$, $T_d = 1.67$ ($N = 10$) and the corresponding value of the maximum sensitivity is $M_s = 1.43$ (indeed, this controller is slightly more robust than the previous ones).

Results related to the set-point step response are shown in Figure 2. Note that the time scale of the plot of the control variables has been reduced in order to highlight the initial part of the transients, which is more interesting.

The resulting values of the integrated absolute errors are $IAE = 10.4$ for the PIDA controller with $\lambda = 0.675$, $IAE = 11.6$ for the PIDA controller with $\lambda = 0.9$, $IAE = 16.1$ for the PID controller tuned with the SIMC rules, and $IAE = 13.1$ for the PID controller tuned with the AMIGO rules.

Process and control variables related to the load disturbance step response are shown in Figure 3. In this case, the integrated absolute errors are $IAE = 9.75$ for the PIDA controller with $\lambda = 0.675$, $IAE = 11.0$ for the PIDA controller with $\lambda = 0.9$, $IAE = 15.7$ for the PID controller tuned with the SIMC rules, and $IAE = 12.8$ for the PID controller tuned with the AMIGO rules.

3.2 Distributed lag process

Distributed lag processes are often present in industry (Shinskey, 2001). Their dynamics can be modelled as an infinite series of infinitesimally small interacting lags, that is (Shinskey, 1994),

$$P(s) = \frac{2K}{e^{\tau s} + e^{-\tau s}} = \frac{K}{\cosh\sqrt{\tau s}}. \tag{27}$$

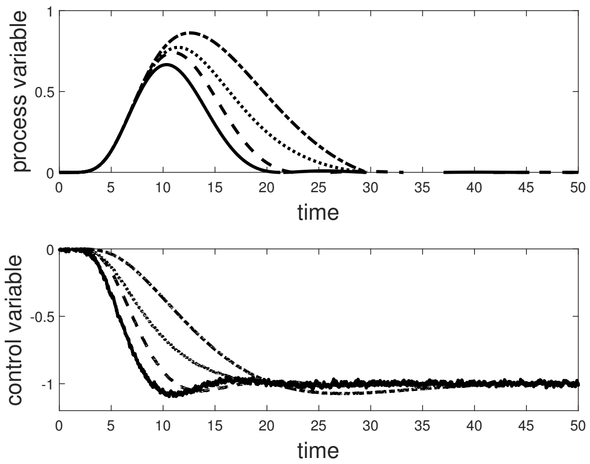

Fig. 3. Load disturbance step response for process P_1. Solid line: PIDA controller with $\lambda = 0.675$. Dashed line: PIDA controller with $\lambda = 0.9$. Dash-dot line: PID controller tuned with the SIMC tuning rules. Dashed line: PID controller tuned with the AMIGO tuning rules.

The hyperbolic cosine can be expanded into an infinite-product series, resulting in

$$P(s) = \frac{K}{[1+(2/\pi)^2\tau s][1+(2/3\pi)^2\tau s][1+(2/5\pi)^2\tau s]\cdots}. \tag{28}$$

In this paper we consider the following process:

$$P_2(s) = \frac{1}{\prod_{i=0}^{n-1}\left[1+\left(\frac{2}{(2i+1)\pi}\right)^2\tau s\right]}, \tag{29}$$

where $\tau = 10$ and we truncate the series in (28) with $n = 20$, by taking into account that the dynamics of the process does not change significantly for $n > 20$ (Shinskey, 2001).

For the purpose of designing the PIDA controller, the following model is first obtained by applying the n-shifting identification procedure:

$$\tilde{P}_2(s) = \frac{1}{(4.053s+1)(0.449s+1)(0.139s+1)^2}e^{-0.17s} \tag{30}$$

(note that, on the contrary of P_1, this can be considered a lag-dominant process). Then, by selecting $r = 6$ and $\lambda = 0.1$, we obtain $K_p = 5.984$, $T_i = 4.605$, $T_d = 0.4898$, $T_a = 0.2507$, and $N = M = 1.019$; the resulting value of the low-pass filter time constant is $T_f = 0.027$. With the designed controller, the maximum sensitivity is $M_s = 1.99$. Another case with $\lambda = 0.2$ is also considered. It results $K_p = 3.176$, $T_i = 4.354$, $T_d = 0.256$, $T_a = 0.191$, $N = M = 0.230$, and $T_f = 0.044$. The corresponding maximum sensitivity is $M_s = 1.64$.

The application of the half rule to the process model (29) yields

$$\tilde{P}_2(s) = \frac{1}{(4.053s+1)(0.531s+1)}e^{-0.36s}, \tag{31}$$

so that the SIMC tuning rules (13) give the following PID parameters: $K_p = 1.37$, $T_i = 4.053$, $T_d = 0.531$ ($N = 10$), with a corresponding value of the maximum sensitivity $M_s = 1.13$.

The application of the tangent method to the process open-loop step response yields the following process model:

$$\tilde{P}_1(s) = \frac{1}{4.05s+1}e^{-0.88s} \tag{32}$$

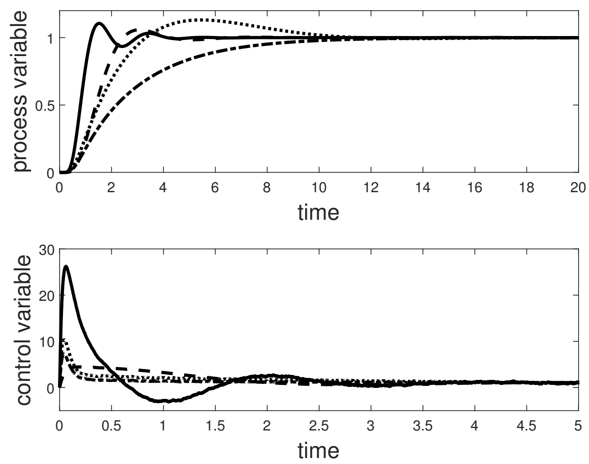

Fig. 4. Set-point step response for process P_2. Solid line: PIDA controller with $\lambda = 0.675$. Dashed line: PIDA controller with $\lambda = 0.9$. Dash-dot line: PID controller tuned with the SIMC tuning rules. Dashed line: PID controller tuned with the AMIGO tuning rules.

(which indicates that the process is lag dominant). The AMIGO tuning rules (18) yields $K_p = 2.27$, $T_i = 2.46$, $T_d = 0.41$ ($N = 10$) with a resulting maximum sensitivity $M_s = 1.46$.

Results related to the set-point response are shown in Figure 4 (note that the time scale of the control variable has been again reduced in order to highlight the initial part of the transients). Those related to the load disturbance step response are shown in Figures 5-6, where the resulting control variables have been split in four different plots in order to better evaluate the effects of the measurement noise.

The resulting values of the integrated absolute errors when a unit set-point step signal is applied are $IAE = 3.27$ for the PIDA controller with $\lambda = 0.1$, $IAE = 3.82$ for the PIDA controller with $\lambda = 0.2$, $IAE = 4.96$ for the PID controller tuned with the SIMC rules, and $IAE = 4.23$ for the PID controller tuned with the AMIGO rules.

On the other hand, when a unit load disturbance step response is applied, we have $IAE = 2.81$ for the PIDA controller with $\lambda = 0.1$, $IAE = 3.28$ for the PIDA controller with $\lambda = 0.2$, $IAE = 4.57$ for the PID controller tuned with the SIMC rules, and $IAE = 3.29$ for the PID controller tuned with the AMIGO rules.

3.3 Non self-regulating process

The final example is related to the high-order non self-regulating process

$$P_1(s) = \frac{1}{s(s+1)^8}. \tag{33}$$

The n-shifting identification procedure yields the process model:

$$\tilde{P}_3(s) = \frac{1}{s(5.43s^3+9.27s^2+5.27s+1)}e^{-2.81s}. \tag{34}$$

The values of the controller parameters for $r = 4$ and different values of λ are plotted in Figure 7. The PIDA controller tuning can therefore be done by selecting $\lambda = 3$, which results in $K_p = 0.213$, $T_i = 19.49$, $T_d = 3.83$, $T_a = 4.79$, and $N = M = 13.11$. The value of the maximum sensitivity is $M_s = 2.49$. If the value of λ is increased to 4, we have $K_p = 0.157$, $T_i = 22.8$, $T_d =$

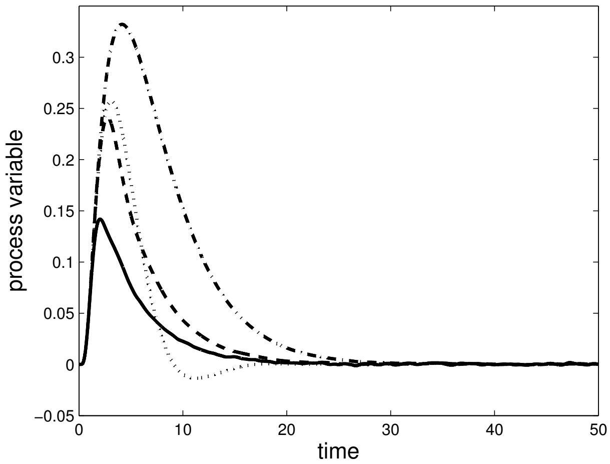

Fig. 5. Process variable of the load disturbance step response for process P_2. Solid line: PIDA controller with $\lambda = 0.675$. Dashed line: PIDA controller with $\lambda = 0.9$. Dash-dot line: PID controller tuned with the SIMC tuning rules. Dashed line: PID controller tuned with the AMIGO tuning rules.

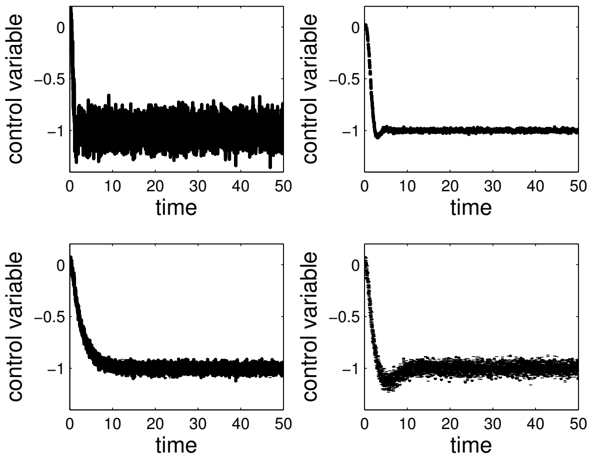

Fig. 6. Control variable of the load disturbance step response for process P_2. Top left: PIDA controller with $\lambda = 0.675$. Top right: PIDA controller with $\lambda = 0.9$. Bottom left: PID controller tuned with the SIMC tuning rules. Bottom right: PID controller tuned with the AMIGO tuning rules.

3.38, $T_a = 3.81$, $N = M = 4.64$, and the maximum sensitivity decreases to the value of $M_s = 2.26$.

The application of the half rule to the process model (33) yields

$$\tilde{P}_3(s) = \frac{1}{s(1.5s+1)}e^{-6.5s} \tag{35}$$

and the SIMC tuning rules (13) give $K_p = 0.077$, $T_i = 52$, $T_d = 1.5$ ($N = 10$), with a corresponding value of the maximum sensitivity $M_s = 1.75$.

Regarding the AMIGO rules, starting from the following process model:

$$\tilde{P}_3(s) = \frac{1}{s}e^{-4.97s} \tag{36}$$

by means of (18) we obtain a PI controller with $K_p = 0.07$ and $T_i = 34.762$ ($T_d = 0$) with a resulting maximum sensitivity $M_s = 1.52$.

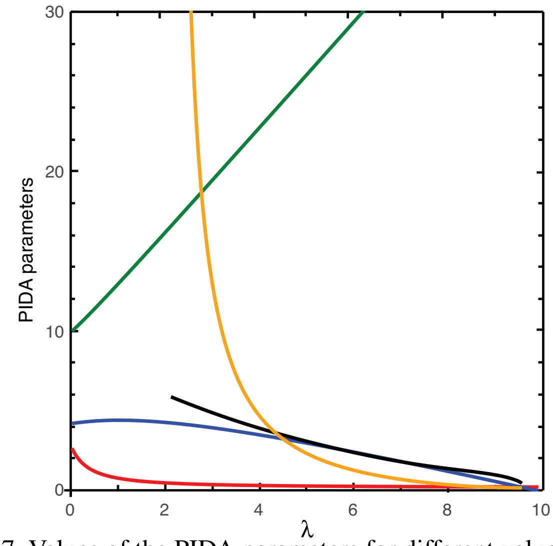

Fig. 7. Values of the PIDA parameters for different values of λ for process P_3; red line: K_p; green line: T_i; blue line: T_d; black line: T_a; orange line: $N = M$. Missing values mean that the parameters are complex.

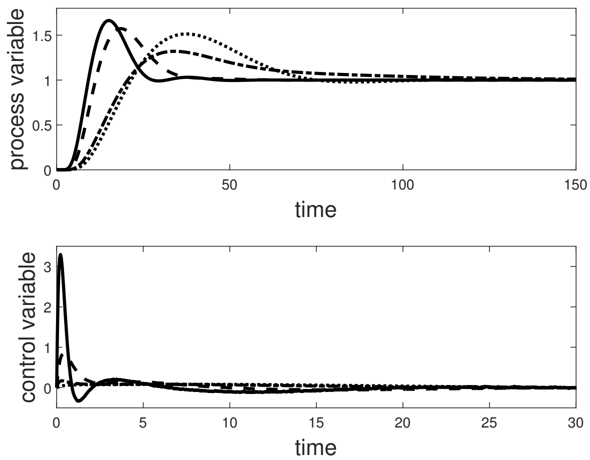

Fig. 8. Set-point step response for process P_3. Solid line: PIDA controller with $\lambda = 0.675$. Dashed line: PIDA controller with $\lambda = 0.9$. Dash-dot line: PID controller tuned with the SIMC tuning rules. Dashed line: PID controller tuned with the AMIGO tuning rules.

The process variable y and the control variable u of the set-point step responses are shown in Figure 8 (once again, the time axis of the control variables is appropriately reduced for the sake of clarity). The resulting values of the integrated absolute error are $IAE = 22.0$ for the PIDA controller with $\lambda = 3$, $IAE = 24.19$ for the PIDA controller with $\lambda = 4$, $IAE = 30.38$ for the PID controller tuned with the SIMC rules and $IAE = 35.87$ for the PID controller tuned with the AMIGO rules.

Results related to the load disturbance step responses are shown in Figures 9. In this case it results $IAE = 99.2$ for the PIDA controller with $\lambda = 3$, $IAE = 152.1$ for the PIDA controller with $\lambda = 3$, $IAE = 671.6$ for the PID controller tuned with the SIMC rules, and $IAE = 502.3$ for the PID controller tuned with the AMIGO rules.

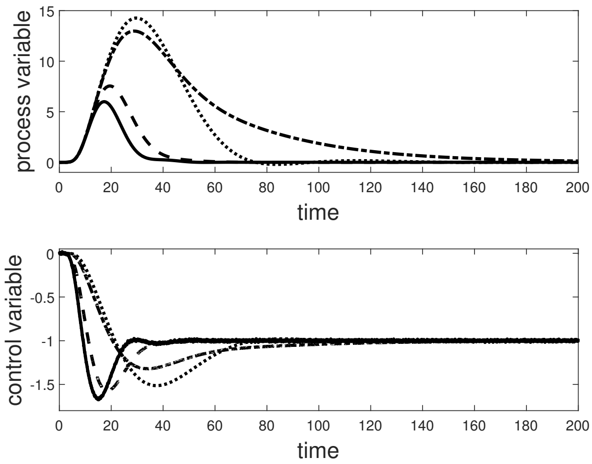

Fig. 9. Load disturbance step response for process P_3. Solid line: PIDA controller with $\lambda = 0.675$. Dashed line: PIDA controller with $\lambda = 0.9$. Dash-dot line: PID controller tuned with the SIMC tuning rules. Dashed line: PID controller tuned with the AMIGO tuning rules.

4. DISCUSSION

From the presented results, related to processes with different characteristics, it appears that the additional parameter of the PIDA controller allows an improvement of the performance in both the set-point following and load disturbance rejection tasks. The tuning procedure makes this controller relatively easy to design and represents therefore a valid alternative option to the PID controller, where two effective tuning rules have been applied, for high-order processes. Indeed, a high-order process model can be usefully exploited. The suitable choice of λ and the additional first-order low-pass filter enable the user to prevent the excessive amplification of the measurement noise and excessively high values of the control variable (namely, the derivative and double derivative kick). Further, the robustness of the PIDA control system is comparable with that of the PID controller.

5. CONCLUSIONS

In this paper we have presented a comparison of the performance achieved by a PIDA controller tuned with an IMC-based methodology with those achieved by a PID controller tuned with the SIMC and AMIGO tuning rules. In the provided analysis, a few high-order processes have been selected, with the aim of representing typical industrial processes. In particular, a dead-time dominant self-regulating process with multiple lags, a distributed-lag process, and a non self-regulating process with multiple lag have been considered.
Although the analysis is clearly not exhaustive, the reported results indicate that the use of PIDA controllers is really promising in those applications where tight control requirements are given and the simplicity of the controller is of concern. This justify a new research effort for such a kind of controllers in order to establish new design methodologies as well as the development of those additional functionalities that are already well established for PID controllers.

REFERENCES

Åström, K.J. and Hägglund, T. (2002). Revisiting the Ziegler–Nichols tuning rules for PI control. *Asian Journal of Control*, 4, 364–380.

Åström, K.J. and Hägglund, T. (2004). Revisiting the Ziegler–Nichols tuning rules for PI control – Part II: The frequency response method. *Asian Journal of Control*, 6, 469–482.

Chidambaram, M. and Sathe, V. (2014). *Relay Autotuning for Identification and Control*. Cambridge University Press, Cambridge, UK.

Huba, M. (2019). Filtered PIDA controller for the double integrator plus dead time. In *16th IFAC Conference on Programmable Devices and Embedded Systems*, 106–113. High Tatras, SK.

Huba, M. and Vrancic, D. (2018). Comparing filtered PI, PID and PIDD control for the FOTD plants. In *Proceedings 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control*, 954–959. Ghent, B.

Huba, M., Vrancic, D., and Bistak, P. (2021). PID control with higher order derivative degrees for IPDT plant models. *IEEE Access*, 9, 2478–2495.

Kumar, M. and Hote, Y. (2018). Robust CDA-PIDA control scheme for load frequency control of interconnected power systems. In *Proceedings 3rd IFAC Conference on Advances in PID Control*, 616–621. Ghent, B.

Kumar, M. and Hote, Y. (2019). Robust IMC-PIDA controller design for load frequency control of a time delayed power system. In *Proceedings of the IEEE Conference on Decision and Control*, 8380–8385. Nice, F.

Liu, T. and Gao, F. (2011). *Industrial Process Identification and Control Design: Step-Test and Relay-Experiment-Based Methods*. Springer, London, UK.

Liu, T., Wang, Q.G., and Huang, H.P. (2013). A tutorial review on process identification from step or relay feedback test. *Journal of Process Control*, 23(10), 1597–1623.

Milanesi, M., Mirandola, E., and Visioli, A. (2022). A comparison between PID and PIDA controllers. In *Proceedings 27th International Conference on Emerging Technologies and Factory Automation*. Stuttgart, D.

O'Dwyer, A. (2006). *Handbook of PI and PID Tuning Rules*. Imperial College Press, London, UK.

Sánchez, J., Dormido, S., and Diaz, J.M. (2021). Fitting of generic process models by an asymmetric short relay feedback experiment— the n-shifting method. *Applied Sciences*, 11(4), 1651.

Shinskey, F.G. (1994). *Feedback Controllers for the Process Industries*. McGraw-Hill, New York, USA.

Shinskey, F.G. (2001). PID-deadtime control of distributed processes. *Control Engineering Practice*, 9, 1177–1183.

Skogestad, S. (2003). Simple analytic rules for model reduction and PID controller tuning. *J. of Proc. Cont.*, 13, 291–309.

Visioli, A. (2006). *Practical PID Control*. Springer, London, UK.

Visioli, A. and Sánchez-Moreno, J. (2023). Design of PIDA controllers for high-order integral processes. In *Proceedings 28th International Conference on Emerging Technologies and Factory Automation*. Sinaia, RO.

Visioli, A. and Sánchez-Moreno, J. (2024). A relay-feedback automatic tuning methodology of PIDA controllers for high-order processes. *International Journal of Control*, 97(1), 51–58.

Wang, Q.G., Lee, T.H., and Lin, C. (2003). *Relay Feedback: Analysis, identification and control*. Springer, London, UK.

Yu, C.C. (1999). *Autotuning of PID Controllers*. Springer-Verlag, London, Great Britain.