ISA Transactions 99 (2020) 339–350

Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

Practice article

Controlling industrial dead-time systems: When to use a PID or an advanced controller

Lucian Ribeiro da Silva [a], Rodolfo César Costa Flesch [b,*], Julio Elias Normey-Rico [b]

[a] Programa de Pós-Graduação em Engenharia de Automação e Sistemas, Universidade Federal de Santa Catarina, 88040-900 Florianópolis, SC, Brazil
[b] Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina, 88040-900 Florianópolis, SC, Brazil

ARTICLE INFO

Article history:
Received 17 October 2018
Received in revised form 6 September 2019
Accepted 6 September 2019
Available online 12 September 2019

Keywords:
Constraints
Dead-time compensators
Model predictive control
PID control
Time delay

ABSTRACT

This work presents a comparative analysis of PID, DTC and MPC strategies when used to control SISO processes with dead time considering characteristics commonly found in industry, such as noisy measurements in the process output and modeling error. For unconstrained processes, it is shown that the performance improvement obtained by using a more advanced control strategy instead of a PID is small or nonexistent for cases which require high robustness. However, for cases with well-known process models it is shown that the improvement obtained by using a more complex control structure is justified even for small delays. For constrained processes it was demonstrated that a PID with anti-windup is able to provide similar or even better results than MPC when robust solutions are considered.

1. Introduction

Many processes in industry have dead times in their dynamic behavior [1]. Examples of this kind of processes can be seen in distillation columns, heat exchangers, solar collector fields, among others [2]. The design of controllers for processes with dead time is not an easy task mainly due to the negative phase introduced by the dead time, which reduces the stability margin [3]. Many control strategies have been proposed in literature to deal with dead-time processes, being three of them the ones most used in practice: proportional, integral and derivative (PID) controllers, dead-time compensators (DTC), and model predictive control (MPC) [1,2,4].

In most cases simple models such as first order plus dead time (FOPDT), integrating plus dead time (IPDT) and unstable first order plus dead time (UFOPDT) can be used with success to represent the process dynamics of single-input single-output (SISO) systems. For these cases, over the years, several PID tuning rules have been proposed, to allow a satisfactory trade-off between robustness and performance, while keeping a relatively simple structure [5,6]. This is one of the main reasons why PID controllers are widely used in industry [7,8].

One of the first DTC strategies, proposed by Smith [9] and still widely used today, is the Smith predictor (SP). The main advantage of this technique over PID controllers is that the effects of dead time can be ideally eliminated from the closed-loop characteristic equation, thus allowing faster responses [2]. Many extensions of the original SP are presented in literature to extend its ideas to a broader class of problems, such as open-loop unstable processes or multi-input multi-output (MIMO) processes [10–12]. Moreover, tuning can be derived for general cases, not being limited to simple models with dead time.

MPC is a control technique which has made a significant impact in industry due to its capability to deal with process constraints and also for presenting good performance when used to control processes with significant dead time [1]. The main drawback of MPC is that an online optimizer is necessary to compute the control action at each sampling instant, which may require a considerable computational effort depending on the process characteristics, thus limiting its use to processes with slows dynamics. There are many works in literature which present strategies to reduce the computation time of the optimization problem, but most of these techniques are based on specific decompositions, which are limited to certain classes of problems, such as those covered in [13,14].

In the past, when only simple and low-memory hardware were available at affordable prices, more complex control strategies, such as DTC and MPC, were only considered for complex SISO or MIMO plants, where the need for high performance justified higher implementation costs. Nowadays, the availability of low-cost hardware allows the implementation of advanced controllers at lower levels. Recently, several researches were conducted to study the application of MPC in SISO processes, for example in the renewable energy area [15,16], heating processes [17], pneumatic processes [18], and power converters and

* Corresponding author.
E-mail addresses: lucian.ribeiro@posgrad.ufsc.br (L.R. da Silva), rodolfo.flesch@ufsc.br (R.C.C. Flesch), julio.normey@ufsc.br (J.E. Normey-Rico).

https://doi.org/10.1016/j.isatra.2019.09.008
0019-0578/

Nomenclature

2DOF	Two-degree-of-freedom
AW	Anti-windup
DTC	Dead-time compensator
ER	Error recalculation
FOPDT	First order plus dead time
FSP	Filtered Smith predictor
GPC	Generalized predictive control
IAE	Integrated absolute error
IFAC	International Federation of Automatic Control
IPDT	Integrating plus dead time
MIMO	Multi-input multi-output
MPC	Model predictive control
PI	Proportional and integral
PID	Proportional, integral and derivative
QP	Quadratic programming
SDD	Stable delay dominant
SISO	Single-input single-output
SLD	Stable lag dominant
SP	Smith predictor
UFOPDT	Unstable first order plus dead time

drives [19–21]. Therefore, it is interesting to analyze the advantages of using these more advanced control structures over the classical PID solutions, considering the real situation in industry.

Comparisons between PID and DTC have been presented in literature since the 1980's. Rivera et al. [22] present a comparison between a PID designed for FOPDT models and an ideal DTC (i.e. with infinity gain) and show that for large ratios of dead time and time constant it is possible to obtain significant performance improvement by using a DTC instead of a PID. In [2] a discussion about when to use a DTC or PID is presented, showing that the performance gains of DTC are more affected by the dead-time estimate error than by the absolute value of dead time. The work of Grimholt and Skogestad [8] compares the performance of SP, PI and PID strategies for FOPDT models, using the integrated absolute error (IAE) performance index and delay margin (D_M) as function of the maximum sensitivity index (M_s) in the interval from 1.2 to 2.0. The results show that, for a fixed M_s value, the improvement obtained by using an SP with a PI as primary controller instead of a PID is almost insignificant for most cases. In [23] a comparison between PID and MPC strategies considering model mismatch and measurements noise is presented, showing that for first order processes the PID performance degrades almost linearly when dead time exceeds twice of the time constant, and for second order processes the same behavior is observed when dead time exceeds about 10% of the equivalent time constant. For these cases, MPC presented significantly better performance than PID. In [24] a comparison between PID and MPC is presented considering no modeling errors. The results show that for all the case studies MPC presented better performance than PID. In [25], the performances of a PI and an MPC are compared for a reactive distillation MIMO process, showing that MPC presents better results than the PI controller. A comparative analysis between a MIMO filtered Smith predictor (FSP) and a PI controller is presented in [26], showing that the main advantage of FSP over the PI strategy is related with the disturbance rejection response of the system if the modeling error is small.

Among the cited works, the main conclusion which can be drawn is that PID controllers still have significant importance due their applicability in a wide range of processes with different characteristics. Although the previously cited works focus on comparing PID and DTC strategies and PID and MPC strategies, none of them compares the three techniques in terms of performance and robustness under the same conditions, including characteristics commonly found in industry, such as measurement noise, and constraints in the magnitude and rate of control signal, and in the magnitude of the process output. In addition, in most cases the comparisons are done for specific case studies and no general conclusion is obtained for other processes. To fill this gap, this paper presents the contributions listed below:

1. A comparative study between PID, DTC and MPC when controlling SISO dead-time processes, considering important aspects, such as performance, robustness and handling of constraints;
2. A formalization of some of the results recently presented in the plenary of Normey-Rico [27] at the IFAC PID 18 conference, where the study of PID, DTC and MPC strategies for typical industrial SISO models was presented;
3. A formulation for dealing with process variable amplitude constraints in PID and DTC controllers;
4. A straightforward method to decide whether to use classical PID controllers or advanced control strategies in typical process control problems, which is perhaps the main contribution of this paper, being of great importance for practitioners.

The MPC strategy used in this work is the generalized predictive controller (GPC) and the DTC is the FSP, which is a strategy that can solve some of the limitations of the original SP structure. The choice for FSP is also based on the fact that it can provide an ideal response for all cases (stable, integrating and unstable plants with dead-time) when a controller with infinity gain is considered.

The rest of this paper is organized as follows. Section 2 presents an analysis of the ideal solution for control of dead-time processes, comparing the original SP and the FSP structures. Section 3 presents a discussion of PID, FSP and GPC structures and also a comparative analysis between the performance and robustness of these control structures considering unconstrained control problems. Section 4 presents a formulation that allows the PID to deal with several types of constraints by using an anti-windup (AW). Furthermore, the results of Section 3 are extended for the constrained case in Section 4. An experimental analysis which compares the performance of a PID and a GPC considering a constrained process modeled as FOPDT is presented in Section 5. The conclusions of the paper are presented in Section 6.

2. Ideal control of dead-time processes

It is well known that the original SP, shown in Fig. 1 if $F_r(s) = 1$, presents good setpoint performance when controlling dead-time processes [8]. In Fig. 1, $C_{SP}(s)$ is the primary controller, $P(s)$ is the plant, $P_n(s) = G_n(s)e^{-L_n s}$ is the nominal model, $G_n(s)$ is the dead-time-free model, L_n is the nominal dead time, $r(t)$ is the reference, $u(t)$ is the control signal, $y(t)$ is the plant output, $e(t)$ is the error, $q(t)$ is the input disturbance, $\hat{y}(t)$ is the model output, $e_p(t)$ is the prediction error, and $F_r(s)$ is a unity gain robustness filter (or prediction filter). For a pure delay process ($P(s) = e^{-Ls}$) and considering an ideal case (i.e. the primary controller $C_{SP}(s)$ has infinity gain), the SP results in an ideal solution, which is a step-like response after dead time (L_n) has elapsed for setpoint changes and after twice the dead time for input (load) disturbances. This ideal solution is presented in (1) and (2) in terms

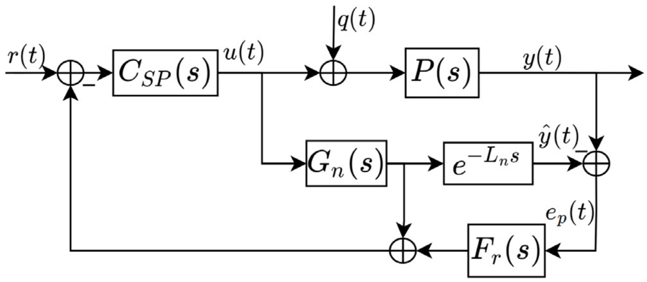


Fig. 1. Filtered Smith predictor, which is equivalent to SP when $F_r(s) = 1$.

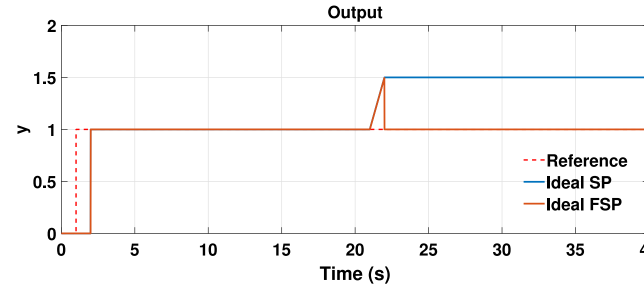


Fig. 2. Performance comparison between SP and FSP for an ideal case.

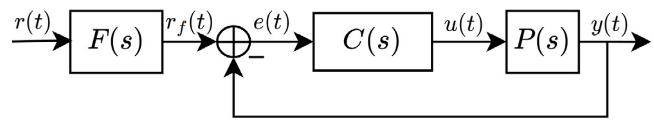


Fig. 3. 2DOF PID structure.

of the closed-loop transfer functions for reference, $H_{yr}(s)$, and for input disturbance rejection, $H_{yq}(s)$.

$$H_{yr}(s) = \mathrm{e}^{-L_n s} \tag{1}$$

$$H_{yq}(s) = \mathrm{e}^{-L_n s}(1 - \mathrm{e}^{-L_n s}) \tag{2}$$

An important point to be highlighted is that even in the ideal case it is not possible to eliminate one delay from the setpoint response and two delays from the disturbance response. However, for systems with any dynamics other than a pure delay, the disturbance rejection performance of SP is degraded, and it is not able provide an ideal response because the open-loop poles appear in the closed-loop response. For example, the transfer function from an input disturbance to the process output, $H_{yq}(s)$, of the ideal SP for a FOPDT process, $P(s) = \frac{1}{\tau s+1}\mathrm{e}^{-L_n s}$, is given by

$$H_{yq}(s) = \frac{1}{\tau s + 1}\mathrm{e}^{-L_n s}(1 - \mathrm{e}^{-L_n s}), \tag{3}$$

which shows that the time constant of the process defines the closed-loop dynamics. Furthermore, the original SP structure is not able to control integrating and unstable processes [2]. A simple solution for these drawbacks was proposed in [28] and is known as FSP. It adds a filter, $F_r(s)$, in the prediction error of the SP structure, as shown in Fig. 1.

Considering the nominal case (i.e. $P_n(s) = P(s)$), the closed-loop transfer functions for the reference and disturbance rejection of the FSP strategy are given by

$$H_{yr}(s) = \frac{C_{SP}(s)P_n(s)}{1 + C_{SP}(s)G_n(s)}, \tag{4}$$

$$H_{yq}(s) = P_n(s)\left[1 - \frac{C_{SP}(s)P_n(s)F_r(s)}{1 + C_{SP}(s)G_n(s)}\right]. \tag{5}$$

The tuning of $F_r(s)$ in the FSP allows to eliminate the open-loop poles of $P_n(s)$ from $H_{yq}(s)$. Thus, if an ideal primary controller, $C_{SP}(s)$, with infinity gain and ideal filter, $F_r(s)$, are used, the ideal closed-loop response can be obtained.

An IPDT example is used to analyze the ideal case. The process model is given by

$$P(s) = \frac{\mathrm{e}^{-s}}{s}, \tag{6}$$

with time given in seconds. Consider that a controller $C_{SP}(s) = K_c$ with infinity gain and a prediction filter $F_r(s) = 1+s$ are used and the plant model is perfect. In this case, the following closed-loop transfer functions are obtained:

$$H_{yr}(s) = \mathrm{e}^{-s} \tag{7}$$

$$H_{yq}(s) = \frac{\mathrm{e}^{-s}}{s} - \frac{\mathrm{e}^{-2s}}{s} - \mathrm{e}^{-2s}, \tag{8}$$

which are the ideal responses for both setpoint tracking and disturbance rejection, as in the pure delay system.

Fig. 2 shows a simulation which compares the performance of SP and FSP for the system, where a unit step reference is applied at 1 s and a step disturbance of amplitude 0.5 is applied at 20 s.

As shown in Fig. 2, the SP cannot reject the disturbance. On the other hand, the FSP rejects the disturbance immediately after 2 s. This can be done due to a property of the FSP that makes it possible to tune the filter $F_r(s)$ so that the equivalent controller has integral action, which is not possible to achieve in the original SP structure [2].

Based on the previous analysis, in this paper, a cost function J that considers the IAE performance index for setpoint tracking and load disturbance rejection is used to quantify how much the output response deviates from the ideal one. The proposed cost function is given by

$$J = \frac{1}{2}\left[\int_{t_s+L_n}^{t_d} |r(t) - y(t)|\mathrm{d}t + \int_{t_d+2L_n}^{\infty} |r(t) - y(t)|\mathrm{d}t\right], \tag{9}$$

where t_s is the time at which the reference change is commanded, and t_d is the time at which the disturbance is applied.[1] It is important to note that the IAE is measured from $t_s{+}L_n$ to t_d for the setpoint tracking and from $t_d + 2L_n$ to infinity for the disturbance rejection because no controller is able to change the output of the system before $t_s + L_n$ for a reference change and $t_d + 2L_n$ for disturbance rejection. Note that by using the FSP structure it is possible to achieve the ideal response, i.e. the performance index $J = 0$ can be reached in the nominal case.

3. Performance and robustness analysis for unconstrained processes

This section compares the two-degree-of-freedom (2DOF) PID of Fig. 3 and an FSP for unconstrained processes. The aim of this section is to aid in the decision of which structure to choose to control dead-time processes when only a low-order model of the plant is available. The MPC strategy is not explicitly considered in the analysis because the DTC structure used is able to provide exactly the same response as GPC for unconstrained cases, as shown in Section 3.2.

[1] t_d is chosen such that $t_d > t_{ss}$, being t_{ss} the settling time of the closed-loop system.

The performance analysis considers the cost function presented in (9). For the robustness analysis, two indices are used: the robustness index, $R_I(\omega)$, and the delay margin, D_M. The first one is given by [2]

$$R_I(\omega) = \frac{|1 + C(j\omega)P_n(j\omega)|}{|C(j\omega)P_n(j\omega)|} \quad \forall\, \omega \geq 0 \tag{10}$$

and is used to check the robust stability condition $R_I(\omega) > \overline{\delta P}(\omega)$, $\forall\, \omega \geq 0$, where $\overline{\delta P}(\omega)$ is the multiplicative uncertainty, $P(\omega) = P_n(j\omega)[1 + \delta P(j\omega)]$, $\overline{\delta P}(\omega) \geq |\delta P(j\omega)|$, $\forall \omega \geq 0$, $C(s)$ is the controller and $P_n(s)$ is the process model used to tune $C(s)$. The delay margin is given by

$$D_M = \frac{P_M}{\omega_c}, \tag{11}$$

where P_M is the phase margin (given in rad) and ω_c is the crossover frequency (given in rad/s). It is used to evaluate the robustness against modeling errors in dead time. D_M represents the smallest amount of time delay which causes the closed-loop system to become unstable [29].

3.1. PID as a low-frequency approximation of FSP

The work of Normey-Rico and Guzmán [6] proposes a PID tuning rule that is based on a low-frequency approximation of the FSP. This technique can be used to control processes which can be modeled as FOPDT, IPDT or UFOPDT. The main idea of this rule is to obtain the equivalent controller of the FSP in a 2DOF structure, as in Fig. 3, and then use Padé approximation for the dead time. The result is a PID series controller,

$$C(s) = \frac{K_c(sT_i + 1)(sT_d + 1)}{sT_i(s\alpha T_d + 1)}, \tag{12}$$

which is tuned using only one parameter, T_0, which represents the desired closed-loop time constant. This parameter is used to define the desired balance in the trade-off between performance and robustness. In this case, the PID parameters, K_c, T_i, T_d and α, are computed based on the chosen value for T_0 and also on the parameters of the model used to represent the process dynamics (see [6]).

Even though this tuning rule is simple, it provides satisfactory results for a wide range of processes with dead time. In addition, since this tuning rule is derived from the FSP, which can provide the ideal solution for dead-time processes, it will be used as a baseline for comparison.

3.2. Relation between FSP and GPC

GPC uses the discrete-time model

$$A(z^{-1})y(k) = z^{-d}B(z^{-1})u(k-1) + \frac{T(z^{-1})\eta(k)}{\Delta} \tag{13}$$

to predict the plant future outputs which are used to calculate, by minimizing a cost function, the control action [1]. In (13), $u(k)$ and $y(k)$ are the input and output of the process, k is the discrete time in samples, $T(z^{-1})$ is a polynomial in the backshift operator z^{-1} that represents the stochastic characteristics of the noise, d is the dead time, $\eta(k)$ is a zero-mean white noise, $\Delta = (1 - z^{-1})$, and $A(z^{-1})$ and $B(z^{-1})$ are polynomials in z^{-1} with order n_a and n_b respectively. The function to be minimized is given by

$$\Gamma = \sum_{j=d+1}^{d+N} [\hat{y}(k+j|k) - r(k+j)]^2 + \sum_{j=1}^{N_u} \lambda[\Delta u(k+j-1)]^2, \tag{14}$$

where N is the prediction horizon, N_u is the control horizon, λ is the control increment weight, $\hat{y}(k+j|k)$ is the predicted output for $k+j$ at time instant k, $r(k+j)$ is the future reference, and $\Delta u(k)$ is the control increment.

The future output predictions computed using (13) can be written in a vector form as

$$\hat{\mathbf{y}} = \mathbf{G}\Delta\mathbf{u} + \mathbf{H}\mathbf{u_1} + \mathbf{S}\mathbf{y_1}, \tag{15}$$

where $\hat{\mathbf{y}} = [\hat{y}(k+d+1|k), \hat{y}(k+d+2|k), \ldots, \hat{y}(k+d+N|k)]^T$, $\Delta\mathbf{u} = [\Delta u(k), \Delta u(k+1), \ldots, \Delta u(k+N_u-1)]^T$, $\mathbf{u_1} = [\Delta u(k-1), \Delta u(k-2), \ldots, \Delta u(k-n_b)]^T$, $\mathbf{y_1} = [\hat{y}(k+d|k), \hat{y}(k+d-1|k), \ldots, \hat{y}(k+d-n_a|k)]^T$, and $\mathbf{G} \in \mathbb{R}^{N \times N_u}$, $\mathbf{H} \in \mathbb{R}^{N \times n_b}$ and $\mathbf{S} \in \mathbb{R}^{N \times n_a+1}$ are constant matrices [2]. By considering the future reference signals $r(k+j) = r(k)$, for $j \geq 1$ and substituting (15) in (14), it is possible find an analytical solution that minimizes Γ, for the unconstrained case, which is given by [2]

$$\begin{aligned}\Delta u(k) = {} & l_{y_1}y(k+d|k) + \cdots + l_{y_{n_a+1}}y(k+d-n_a|k) \\ & + l_{u_1}\Delta u(k-1) + l_{u_2}\Delta u(k-2) + \cdots + l_{u_{n_b}}\Delta u(k-n_b) \\ & + \sum_{i=1}^{N} v_i r(k),\end{aligned} \tag{16}$$

where the coefficients $[l_{y_1}, \ldots, l_{y_{n_a+1}}]$, $[l_{u_1}, \ldots, l_{u_{n_b}}]$ and $[v_1, \ldots, v_N]$ can be calculated from the model parameters, the prediction horizon, N, and the control increment weight, λ.

Reference [2] shows that in the unconstrained case GPC can be represented as a 2DOF FSP in the discrete-time domain with a reference filter $F(z)$, a primary controller $C_{SP}(z)$ and a predictor filter $F_r(z)$ given by

$$C_{SP}(z) = -\frac{l_{y_1} + \cdots + l_{y_{n_a+1}}z^{-n_a}}{(1 - z^{-1})(1 - l_{u_1}z^{-1} - \cdots - l_{u_{n_b}}z^{-n_b})}, \tag{17}$$

$$F(z) = -\frac{v_1 z^{d+1} + \cdots + v_N z^{d+N}}{l_{y_1} + \cdots + l_{y_{n_a+1}}z^{-n_a}}, \tag{18}$$

$$F_r(z) = \frac{l_{y_1}F_d(z^{-1}) + \cdots + l_{y_{n_a+1}}F_{d-n_a}(z^{-1})}{l_{y_1} + \cdots + l_{y_{n_a+1}}z^{-n_a}}, \tag{19}$$

where $[F_d, F_{d-1} \ldots F_{d-n_a}]$ are obtained by solving a Diophtantine equation (see [2] for details). This analysis shows that an FSP can obtain the same responses as an unconstrained GPC for any dead-time process if a proper tuning is considered. Furthermore, in [2] it is demonstrated that using this formulation to tune $F(z)$ and $C_{SP}(z)$, but choosing a different $F_r(z)$ (which is not derived from the optimal predictor model), it is possible either to improve the dynamics of disturbance rejection or the closed-loop robustness. This structure became known as DTC-GPC, and presents the same nominal setpoint response as the original unconstrained GPC.

Remark. Although in this paper the focus is on GPC, a linear model-based strategy, many of the ideas related to the robustness of the MPC strategy when controlling a dead-time process can be extended to other linear and nonlinear model based MPC strategies [30–32]. Using classical predictors instead of the optimal one, usually adopted in most MPC strategies, can provide a better balance between robustness and performance in MPC when it is used to control dead-time processes.

3.3. Comparative analysis between PID and FSP

This section compares the PID described in Section 3.1 and FSP for dead-time processes which can be modeled as FOPDT, IPDT or UFOPDT. The objective of this analysis is to give some guidelines for the control designer to define whether to use a PID or an FSP.

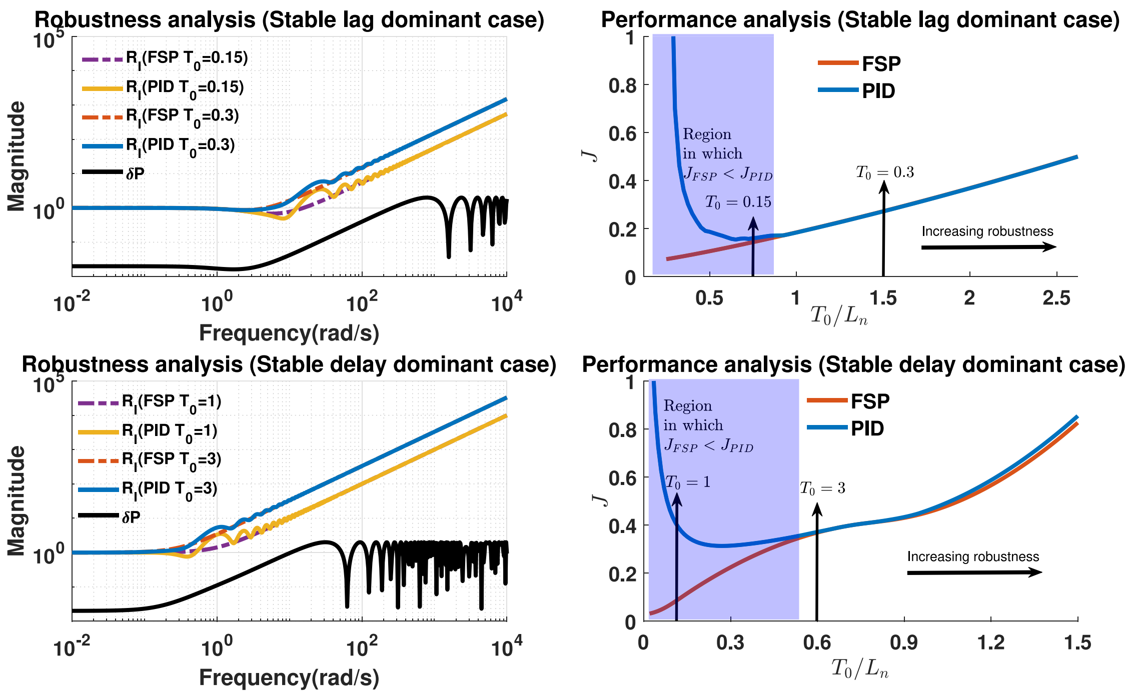


Fig. 4. Comparative analysis between the PID and FSP for stable lag dominant and stable delay dominant cases.

Remark. The results of Section 3.2 show that, for the unconstrained case, the comparison between PID and FSP can also be considered as a comparison between PID and GPC, since it is possible to represent any GPC as an FSP in this case.

As the PID tuning rule used in this study and FSP have the same tuning parameter, T_0, it becomes easier to compare aggressive and conservative tuning rules for both controllers. The analysis considers a trade-off between robustness, measured with $R_I(\omega)$ and D_M, and performance, using J as a function of the ratio T_0/L_n. Moreover, for robustness analysis, the modeling error $\delta P(\omega)$ is also shown in the figures. For all cases $\delta P(\omega)$ is computed for a modeling error of 2% in the gain, time constant (when applicable) and dead time. In order to present a general analysis, a normalized model of the process with unity gain and time constant (except for IPDT) is considered. The gain does not change the dynamics of the process, since the controller gain can be tuned accordingly to compensate for different gains of the process. The relation which really matters is between the fast model dynamics and the time delay, so it is possible to keep the dynamics fixed and change just the delay value without loss of generality. The three process models considered are: $P_s(s) = \frac{e^{-L_n s}}{s+1}$, $P_i(s) = \frac{e^{-L_n s}}{s}$ and $P_u(s) = \frac{e^{-L_n s}}{s-1}$, with time given in seconds.

The first part of the comparative analysis considers a stable lag dominant (SLD) process, with $L_n = 0.2$ s, and a stable delay dominant (SDD) process, with $L_n = 5$ s. Fig. 4 shows a comparative analysis between PID and FSP for both cases.

As shown in Fig. 4, for the SLD case, considering a robust tuning with $T_0 = 0.3$ s, FSP and PID present similar performance and similar robustness characteristics. In fact, it can be observed that the more robust is the tuning of the controllers, the greater are their similarities in terms of robustness and performance. In this case, the D_M values obtained for the controllers were $D_{M_{PID}} = 0.271$ s and $D_{M_{FSP}} = 0.281$ s, which show the equivalence in the robustness properties of both strategies for this tuning. On the other hand, for tuning solutions aiming fast responses, with $T_0 = 0.15$ s for example, there is a significant performance improvement if FSP is used (as can be seen in the shaded area). Furthermore, high differences in robustness are observed for small T_0/L_n values. The D_M values obtained for PID and FSP were respectively $D_{M_{PID}} = 0.11$ s and $D_{M_{FSP}} = 0.06$ s, which show that FSP is more sensitive to dead-time uncertainties than PID. The results for the SDD case, when a robust tuning is considered ($T_0 = 3$ s), are very similar to the ones obtained for the SLD case in terms of both robustness and performance. The D_M values obtained for this tuning were close to each other, resulting in $D_{M_{PID}} = 8.94$ s and $D_{M_{FSP}} = 9.23$ s. In addition, for a fast tuning ($T_0 = 1$ s) again PID presents worse performance and better robustness in terms of dead-time uncertainties when compared with FSP. The D_M obtained for both controllers are $D_{M_{PID}} = 5.52$ s and $D_{M_{FSP}} = 1.49$ s.

The second part of the comparative analysis considers integrating and unstable processes, both with dead time of $L_n = 1$ s. Fig. 5 shows the robustness and the performance analysis for both cases.

As can be seen in Fig. 5, the results for the integrating case are very similar to the ones obtained for the SLD case, where the D_M values obtained for a robust tuning ($T_0 = 1$ s) were $D_{M_{PID}} = 0.53$ s and $D_{M_{FSP}} = 0.48$ s, and for a fast tuning ($T_0 = 0.5$ s) were $D_{M_{PID}} = 0.192$ s and $D_{M_{FSP}} = 0.142$ s. For the unstable case, it is possible to note that FSP presents a significant advantage in terms of both robustness and performance over the PID when robust and fast tuning solutions are considered. It is important to note that along the whole interval of T_0/L_n analyzed in this paper the performance of the FSP is better than the one of PID for the unstable case. The D_M values obtained for a robust tuning ($T_0 = 2$ s) were $D_{M_{PID}} = 0.14$ s and $D_{M_{FSP}} = 0.18$ s, and for a fast tuning ($T_0 = 1.2$ s) were $D_{M_{PID}} = 0.06$ s and $D_{M_{FSP}} = 0.12$ s.

As can be seen in the previous examples, except for the unstable case, when robust solutions are considered the performance and robustness of PID and FSP are essentially the same. For fast tuning solutions, the performance of FSP is better than the one of PID even for processes with small delays (lag dominant). On the other hand, when aggressive tunings are considered, the delay margin values for FSP are considerably smaller than the ones of PID. This is caused by an oscillatory behavior next to the region of magnitude 1 in the FSP loop-function frequency response, which

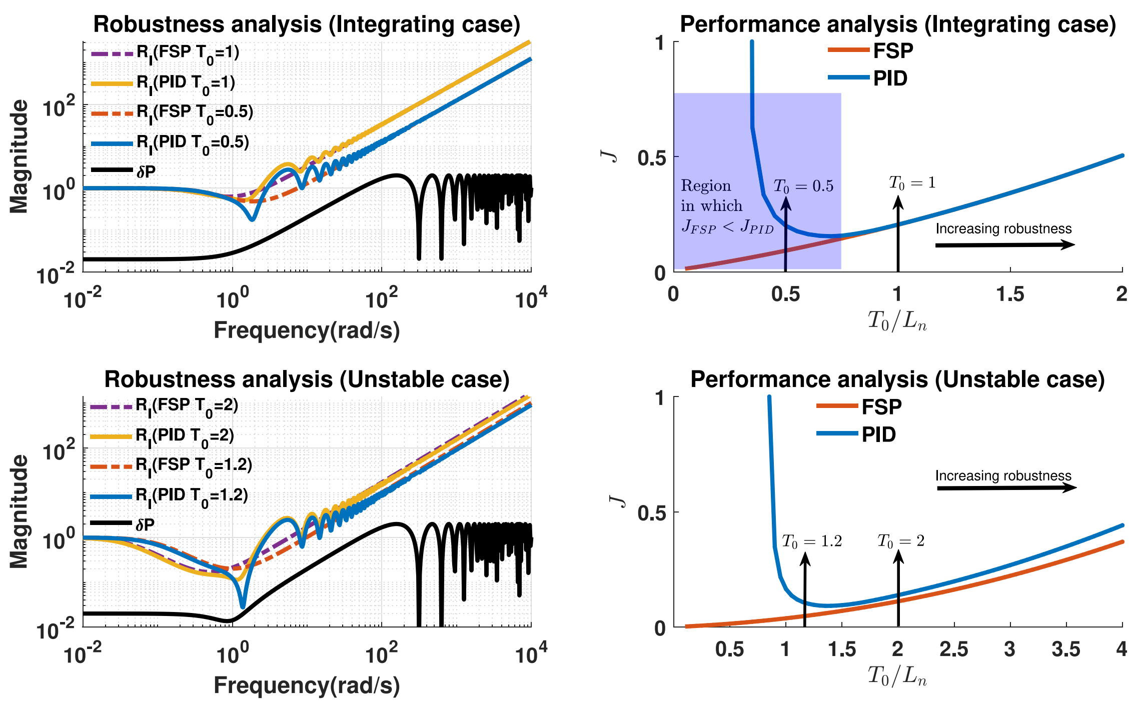


Fig. 5. Comparative analysis between the PID and FSP for integrating and unstable cases.

causes a jump of the crossover frequency ω_c to a higher frequency resulting in a lower D_M value. The study of this phenomenon, known as *crossover proliferation*, can be seen in [33]. Similar drop in D_M was observed for the SP in [8,34].

As a general conclusion of the results presented in this section it can be stated that in most cases, when a high robustness system is necessary, a well-tuned PID is able to provide similar results to FSP, even if the process is delay dominant. On the other hand, when robustness is not a mandatory requirement and a fast tuning is required, the performance improvement of using an FSP instead of a PID is relevant and this result is valid even if the process is lag dominant. As this conclusion is valid for any constant ratio L_n/τ, it is possible to state that the advantages of using an FSP are more associated with the process modeling error than with the absolute value of dead time.

3.4. Unconstrained case example

To better illustrate the results presented in Section 3.3, a comparison between PID and FSP for a particular example is presented. The unconstrained case considered in this analysis is a second order SDD process given by

$$P(s) = \frac{1}{s^2 + 0.8s + 1} e^{-2s}, \tag{20}$$

with time given in seconds, which was approximated by the following FOPDT

$$P_m(s) = \frac{1}{0.62s + 1} e^{-2.7s}. \tag{21}$$

The example also considers measurement noise with normal distribution and variance of 0.02. The first part of this case study compares a PID and an FSP in terms of performance, considering the same tuning rule discussed in Section 3.1. Using a robust tuning with $T_0 = 2$ s the PID parameters obtained are $K_c = 0.355$, $T_i = 1.758$ s, $T_d = 1.35$ s, $\alpha = 1.305$, and the resulting reference filter is given by

$$F(s) = \frac{1.14(s + 0.5)}{(s + 0.57)}. \tag{22}$$

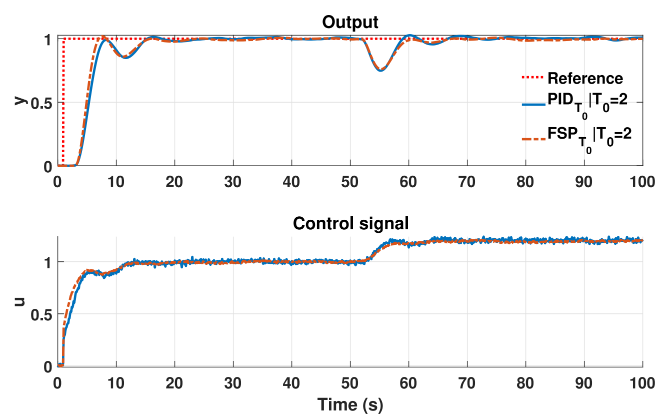


Fig. 6. Comparative analysis between the performance of PID and FSP for the unconstrained case (robust tuning).

The resulting primary controller of the FSP, $C_{SP}(s)$, is a PI with $K_c = 0.31$ and $T_i = 0.62$ s, and the prediction filter, $F_r(s)$, is given by

$$F_r(s) = \frac{0.88(s + 0.57)}{s + 0.5}. \tag{23}$$

Fig. 6 shows the simulation of the system and compares the performance of the two strategies for a unit step reference at $t = 1$ s and a step load disturbance of amplitude -0.2 at $t = 50$ s. As shown in Fig. 6, the performance of the two strategies is similar for both reference tracking and disturbance rejection. This similarity can also be observed by the equivalent performance indices for PID and FSP, which are respectively $J_{PID} = 21.1$ and $J_{FSP} = 20.2$. The D_M values obtained for PID and FSP are $D_{M_{PID}} = 5.53$ s and $D_{M_{FSP}} = 5.66$ s, respectively.

In order to obtain a faster response, a new tuning of the controllers was performed. In this case the tuning of the controllers was performed using a perfect model of the process. The PID was tuned using SWORD, which is a tool for optimal PID design

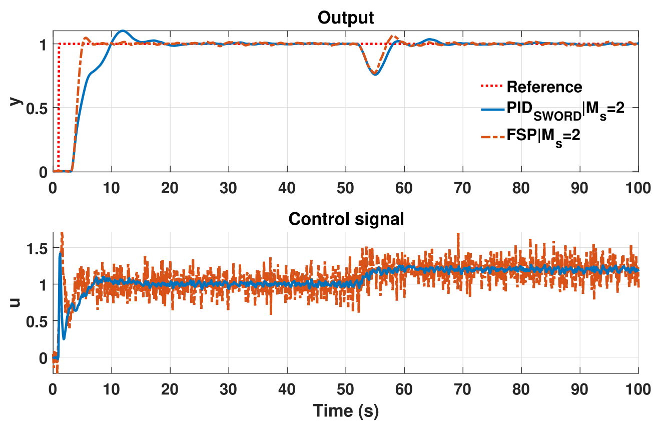


Fig. 7. Comparative analysis between the performance of PID and FSP for the unconstrained case (fast tuning).

of robust PID controllers [35]. Both controllers were tuned to achieve $M_s = 2$. The PID parameters are $K_c = 0.282$, $T_i = 0.992$ s, $T_d = 1.984$ s and it also considers a second-order low-pass filter $F_{LP}(s)$ in the feedback-loop given by

$$F_{LP}(s) = \frac{1}{0.035s^2 + 0.25s + 1}. \tag{24}$$

As SWORD focuses on optimizing only the disturbance rejection response, the resulting setpoint response is oscillatory. A second-order reference filter given by (25) is used to reduce the oscillations in the transient setpoint response of the PID.

$$F(s) = \frac{s^2 + 1.4s + 1}{(s+1)^2} \tag{25}$$

For the FSP, the resulting primary controller $C_{SP}(s)$ and robustness filter $F_r(s)$ are given by

$$C_{SP}(s) = \frac{75.13(s^2 + 2.03s + 1.96)}{s(s + 26.96)}, \tag{26}$$

$$F_r(s) = \frac{0.2(s + 11.16)}{s + 2.23}. \tag{27}$$

Fig. 7 shows the simulation of the system for the new tuning. As can be seen, using a tuning to obtain a fast response causes an oscillatory response in the reference tracking and disturbance rejection responses of the PID. On the other hand, the FSP tuned using the perfect model presents a significant improvement in the performance with less overshoot and smaller oscillations in the disturbance rejection response when compared to PID. When FSP is considered the system is more sensitive to noise due to the large gain of the controller at high frequencies. The performance indices for this case are $J_{PID} = 18.7$ and $J_{FSP} = 11.8$. The delay margin values for PID and FSP are $D_{M_{PID}} = 3.87$ s and $D_{M_{FSP}} = 0.52$ s, which show that FSP is less robust than PID in terms of dead time uncertainties.

This case study demonstrates that when only a low-order model of the process is available and a robust solution is necessary, the performance improvement of using an FSP instead of a PID is small. On the other hand, when a good model of the process is available, a well-tuned FSP is able to provide much better results than PID, as shown in Fig. 7. These results allow the control designer to choose between a PID controller or a more complex structure, such as FSP or GPC, taking into account performance and robustness specifications. However, this analysis is sometimes not enough, since in real processes there are physical limitations. Section 4 extends the analysis of this section to processes with constraints.

4. Performance and robustness analysis for constrained processes

This section compares PID with an AW scheme and GPC strategies in terms of performance and robustness, where constraints in the magnitude and rate of change of the control signal and in the output of the process are considered. Firstly, a formulation that allows the PID with AW to deal with several types of constraints is proposed. Then, a comparative analysis between PID with the proposed formulation and GPC for a process subjected to constraints is presented. For the sake of brevity, the simulation case study presented in this section just considers an IPDT process, but it is also valid for the FOPDT and UFOPDT cases.

4.1. Control with constraints

GPC considers the process constraints in the minimization of the cost function (14), which results in a quadratic programming (QP) problem that needs to be solved online to find the optimal control signal at each sampling instant. In such cases, the length of the control horizon, N_u, affects directly the complexity of the QP problem and consequently the computational effort to find the solution. The literature of MPC presents some works which show that typically the choice of $N_u = 1$ is enough to control simple plants, such as stable dead-time and nonminimum-phase processes [36–38]. In this case, it is much easier to solve the optimization problem.

On the other hand, PID controllers consider control constraints *a posteriori*, that is, using AW schemes to avoid undesirable effects of the constraints on the process output. In this work, based on the behavior of the GPC with $N_u = 1$, an algorithm to compute the control action of PID controllers for constrained systems is presented.

Consider the following discrete-time FOPDT model[2]

$$(1 - az^{-1})y(k) = z^{-d}bu(k-1), \tag{28}$$

subjected to the following constraints

$$\begin{aligned} &u_{min} \le u(k) \le u_{max}, && \forall k,\\ &\Delta u_{min} \le \Delta u(k) \le \Delta u_{max}, && \forall k,\\ &y_{min} \le y(k) \le y_{max}, && \forall k, \end{aligned} \tag{29}$$

where u_{min} and u_{max} are the minimum and maximum magnitudes of the control signal, Δu_{min} and Δu_{max} are the minimum and maximum rates of change in the control signal, and y_{min} and y_{max} are the minimum and maximum allowed outputs of the process. The main idea to compute a feasible control action is to rewrite all the constraints in terms of $u(k)$. This can be done by considering $\Delta u(k) = u(k) - u(k-1)$, and also using the prediction model of GPC with $N_u = 1$. Thus, the constraints in (29) can be rewritten as

$$\begin{aligned} u_{min} \le\, & u(k) \le u_{max}, && \forall k,\\ \Delta u_{min} + u(k-1) \le\, & u(k) \le \Delta u_{max} + u(k-1), && \forall k,\\ u_{y_{min}} \le\, & u(k) \le u_{y_{max}}, && \forall k, \end{aligned} \tag{30}$$

where $u_{y_{min}}$ and $u_{y_{max}}$ are the minimum and maximum control signals to maintain the predictions of the future outputs inside the constraint boundaries and must be determined at each sampling instant. To obtain these values, firstly the future output predictions need to be computed by using (13), with $T(z^{-1}) = 1$. Future increments of the control action are necessary for computing the output predictions from $k+d+1$ to $k+d+N$, where N is the control horizon, however if $N_u = 1$, $\Delta u(k+j) = 0$ for $j \ge 1$, so the

[2] For simplicity, in this paper only this model is considered, but the results are also valid for other models.

future predictions can be computed in terms of the past outputs, the past and the current increments of the control action. For the particular model (28) the predictions after the dead time can be computed using

$$\hat{y}(k+d+j|k) = (a-1)\hat{y}(k+d+j-1|k) + a\hat{y}(k+d+j-2|k) + b\Delta u(k+j-1), \quad (31)$$

for $j = 1 \dots N$ and $\Delta u(k+j-1) = 0$ for $j > 1$. Another aspect that must be taken into account is that for processes with time delay, the current control action affects the output only after $d+1$ sampling periods. For that reason, the future output predictions used to compute $u_{y_{min}}$ and $u_{y_{max}}$ must be taken from $k+d+1$ to $k+d+N$. Eq. (31) can be rewritten as function of $\hat{y}(k+d|k)$, $\hat{y}(k+d-1|k)$ and $\Delta u(k)$, thus resulting in

$$\hat{y}(k+d+j|k) = s_{j1}\hat{y}(k+d|k) + s_{j2}\hat{y}(k+d-1|k) + g_j\Delta u(k), \quad (32)$$

where

$$\begin{aligned} s_{11} &= 1-a, \\ s_{12} &= a, \\ s_{ji} &= \sum_{k=1}^{2} s_{1k}s_{(j-k)i}, \text{ with } s_{j0} = s_{0i} = 1, \\ g_j &= \sum_{k=0}^{j-1} a^k b, \end{aligned} \quad (33)$$

for $j = 1 \dots N$ and $i = 1 \dots 2$.

To compute $u_{y_{min}}$ and $u_{y_{max}}$, it is necessary guarantee that the future output predictions satisfy

$$y_{min} \le \hat{y}(k+d+j|k) \le y_{max}, \forall j = 1 \dots N. \quad (34)$$

By substituting (32) in (34) and considering that $\Delta u(k) = u(k) - u(k-1)$, it is possible to obtain the minimum and maximum control signal values to be applied at time instant k to satisfy the output constraints at time instant $k+d+j$, $u_{y_{min}}(j)$ and $u_{y_{max}}(j)$ respectively, as

$$\begin{aligned} u_{y_{min}}(j) &\ge \frac{y_{min} - s_{j1}\hat{y}(k+d|k) - s_{j2}\hat{y}(k+d-1|k)}{g_j} + u(k-1), \\ u_{y_{max}}(j) &\le \frac{y_{max} - s_{j1}\hat{y}(k+d|k) - s_{j2}\hat{y}(k+d-1|k)}{g_j} + u(k-1), \\ &\text{for } j = 1 \dots N. \end{aligned} \quad (35)$$

From (35) it is possible to compute $u_{y_{min}}$ and $u_{y_{max}}$ as

$$\begin{aligned} u_{y_{min}} &= \max\left\{u_{y_{min}}(j)\right\}, \\ u_{y_{max}} &= \min\left\{u_{y_{max}}(j)\right\}, \\ &\text{for } j = 1 \dots N, \end{aligned} \quad (36)$$

since a value in the feasible region of (36) guarantees that all the constraints defined in (35) are satisfied.

Thus, after computing the control signal limits for each constraint, it is possible to reformulate (30) as

$$U_{min} \le u(k) \le U_{max}, \ \forall k \quad (37)$$

where

$$\begin{aligned} U_{min} &= \max\{u_{min}, \Delta u_{min} + u(k-1), u_{y_{min}}\}, \\ U_{max} &= \min\{u_{max}, \Delta u_{max} + u(k-1), u_{y_{max}}\} \end{aligned} \quad (38)$$

are the boundaries that define a feasible region for all the constraints defined in (29). Moreover, it is also possible to include AW techniques in the previous formulation for cases in which the control action computed is outside the feasible region defined by U_{min} and U_{max}.

Finally, Algorithm 1 shows a systematic way to compute the optimal control signal for processes subjected to the constraints presented in (29). It is important to note that it is possible to use the same algorithm for high-order process models by modifying (28) and recomputing Eqs. (31) to (35).

Algorithm 1: control with constraints

```
repeat
    Compute u(k)
    for j = 1 to d + N do
        Compute the predictions ŷ(k + j)
    Compute u_ymin and u_ymax using (36)
    Compute U_min and U_max using (38)
    if u(k) ≤ U_min or u(k) ≥ U_max then
        Use AW to recalculate u(k)
    Apply u(k) to the plant
until controller is stopped
```

4.2. PID and GPC comparative analysis

This section presents a comparative analysis between a PID with the formulation of Section 4.1 and GPC. The study considers both robustness and performance of the controllers and aims to demonstrate that the analysis presented in Section 3 is also valid for processes subjected to constraints.

For the comparative analysis an IPDT process, $P_i(s) = \frac{e^{-L_n s}}{s}$, with $L_n = 1$ s is considered. In the first part of the analysis, the perfect model is used to tune the controllers, which allows a high value of M_s to be used. The discrete-time representation of the process with a zero-order hold and sampling time of 0.1 s is

$$P_i(z) = \frac{0.1z^{-10}}{z-1}. \quad (39)$$

The controllers were both tuned to obtain a fast response, with $M_s = 3.8$. For this tuning, the parameters obtained for the PID are $K_c = 0.99$, $T_i = 2.44$ s, $T_d = 0.5$ s, $\alpha = 0.21$, $T_0 = 0.72$ s and a reference filter given by

$$F(s) = \frac{0.05(s+10)}{s+0.47}. \quad (40)$$

For the GPC, the tuning parameters are $N = 9$, $N_u = 1$ and $\lambda = 5$.

This case also considers constraints in magnitude and rate of change of the control signal $u_{min} = -0.1$, $u_{max} = +0.6$, $\Delta u_{min} = -0.1$, $\Delta u_{max} = +0.1$, and in the output of the process $y_{min} = 0$ and $y_{max} = 1.1$. To deal with the constraints, a PID using the formulation presented in Section 4.1 with $N = 9$ and with the error recalculation (ER) AW strategy is used. The ER AW strategy modifies both the current control and error signals to keep the controller states consistent with the input signal effectively applied to the process [39]. This AW technique is easy to implement and presents good results especially in process subjected to noisy measurements [40].

Fig. 8 shows the simulation for a unit step reference at $t = 1$ s and a step load disturbance of amplitude -0.2 at $t = 50$ s. As can be seen, both controllers were able to deal with all the constraints, however the GPC presented a better performance than PID, presenting a response with no overshoot and disturbance rejection without oscillations. The performance indices obtained for this case are $J_{GPC} = 7.7$ and $J_{PID} = 9.8$. However, in terms of robustness, the GPC presented a very small delay margin $D_{M_{GPC}} = 0.09$ s when compared to PID, which presented $D_{M_{PID}} = 0.37$ s. Thus, the price to pay for the better performance of GPC is a less robust closed-loop system. It is important to note that, for a given

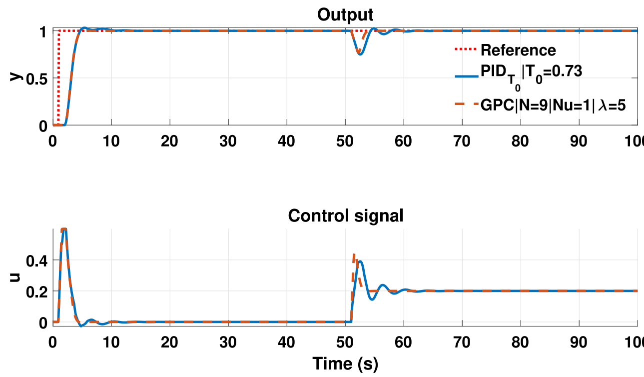


Fig. 8. Comparative analysis between PID and GPC considering no modeling error.

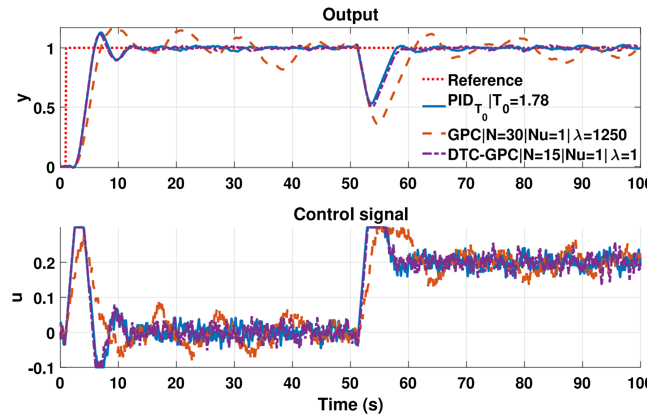


Fig. 9. Comparative analysis between PID, GPC and DTC-GPC considering modeling error and measurement noise.

M_s, even using a longer control horizon N_u the performance of the GPC cannot be improved.

The second part of the analysis aims to evaluate a tuning which would be more likely to be used in a real industrial application, taking into account 20% of modeling error in the gain and in the dead time, and measurement noise with normal distribution and variance of 0.02. As the original GPC is very sensitive to noisy measurements, this case also compares the performance and robustness of a DTC-GPC strategy. The controllers were tuned to obtain a robust solution with $M_s = 1.9$ for the unconstrained case, which is inside the typical design range of values used in industry [7]. The PID parameters for this tuning are $K_c = 0.63$, $T_i = 4.56$ s, $T_d = 0.5$ s, $\alpha = 0.44$, $T_0 = 1.78$ s and the resulting reference filter is given by

$$F(s) = \frac{0.39(s + 0.56)}{s + 0.22}. \tag{41}$$

For the GPC, the tuning parameters are $N = 30$, $N_u = 1$ and $\lambda = 1250$. Using a small value of λ leads to a more aggressive control action and a poorer performance. For the DTC-GPC, the tuning parameters are $N = 15$, $N_u = 1$ and $\lambda = 1$, with a robustness filter given by

$$F_r(z) = \frac{0.149(z - 0.977)}{(z - 0.944)^2}. \tag{42}$$

The DTC-GPC robustness filter improves system robustness and allows tuning the controller with a smaller λ than the one used for GPC. The improvement in the robustness obtained in this example with the DTC-GPC can be generalized for other cases and other MPC formulations, but in all cases the MPC tuning must respect a trade-off between performance and robustness.

As the controllers were tuned to obtain a robust solution, the control signal is smoother and does not reach the saturation limits considered in the previous analysis. In order for the control signal to saturate, the limits considered in this example are $u_{min} = -0.1$, $u_{max} = +0.3$, $\Delta u_{min} = -0.02$, and $\Delta u_{max} = +0.02$. In addition, the output constraints were implemented as soft constraints in GPC to avoid the infeasibility of the optimization procedure that can be caused by the large modeling error [41]. In order to have an equivalent response with the PID controller, the output constraints were also relaxed in this case.

Fig. 9 shows the simulation of the system for the new tuning of the controllers. As can be seen, PID and DTC-GPC obtained very similar performances, while GPC presented an oscillatory response caused by measurement noise. The performance indices for the three cases were $J_{PID} = 24.1$, $J_{GPC} = 66.5$, and $J_{DTC-GPC} = 24.4$. In terms of delay margin, the three controllers obtained similar values: $D_{M_{PID}} = 1.09$ s, $D_{M_{GPC}} = 0.93$ s, and $D_{M_{DTC-GPC}} = 1.08$ s.

The results of this example reinforce the central idea of this work: even for constrained systems, for typical robust industrial solutions based on simple models with delay, PID is the best choice due to its simplicity and a very good trade-off between robustness and performance.

5. Experimental case study

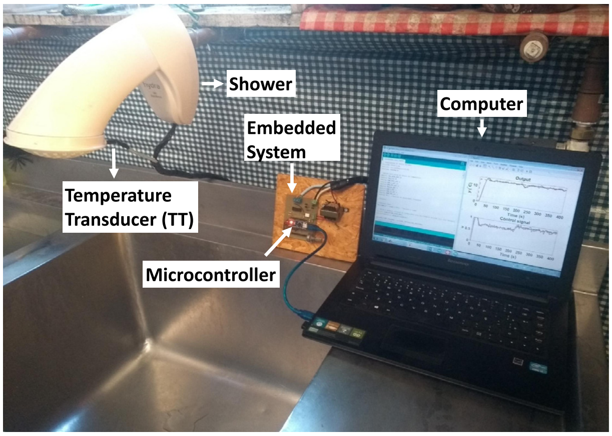
This section presents an experimental analysis which compares the performance of PID with ER AW and GPC for a particular process subjected to constraints. The process is composed of an electric shower and an embedded system, in which the process variable is the variation of the shower water temperature and the manipulated variable is the number of trigger pulses, in a period of 1 s, applied to the TRIAC for controlling the power delivered to the shower heating resistor. A picture of the process and a general diagram are shown in Fig. 10, where TT is the temperature transducer, TC is the temperature controller and R is the shower heating resistor. The embedded system contains a microcontroller ATmega328P and the TRIAC driver circuit, which is composed of a zero crossing detector, an optoisolator and passive electrical components. In both cases, the embedded system is responsible for reading the shower water temperature from the temperature transducer and driving the TRIAC using the TRIAC driver circuit. The PID algorithm was implemented directly in the embedded system, but the GPC algorithm was implemented in a desktop computer which uses an optimization algorithm to deal with the constraints. In both cases the sampling time is 1 s.

5.1. Model identification

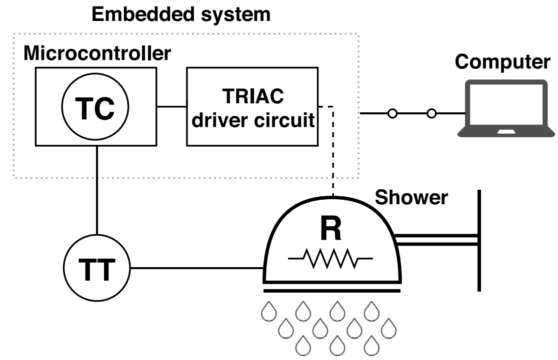
The control action $u(t)$ was normalized so that $u(t) = 1$ is equivalent to the maximum number of trigger pulses (120 power-line semi-cycles), which represents the maximum power applied to the shower heating resistor, and $u(t) = 0$ represents no trigger pulse, i.e. the minimum power. The process output $y(t)$ is the temperature variation, in degree Celsius, in relation to the initial water temperature. The identification of the process model was performed using power steps with different amplitudes. The resulting model is given by

$$\frac{Y(s)}{U(s)} = \frac{18.7e^{-8s}}{13.7s + 1}, \tag{43}$$

with time given in seconds.



(a) Picture of the process



(b) General process diagram

Fig. 10. Illustration of the experiment.

The discrete-time representation of the process with a zero-order hold is

$$\frac{Y(z)}{U(z)} = \frac{1.316z^{-8}}{z - 0.923}. \quad (44)$$

The only constraints considered in this case are the minimum and maximum magnitudes of the control signal, i.e. $u_{min} = 0$ and $u_{max} = 1$.

5.2. Experimental results

Based on the identified model of the process, the PID and the GPC were tuned to obtain reference tracking and disturbance rejection responses with no oscillations. The PID tuning parameters are $K_c = 0.06$, $T_i = 12.41$ s, $T_d = 4$ s, $\alpha = 0.41$, with $T_0 = 8$ s. The PID also uses the formulation presented in Section 4.1 and the ER AW strategy to deal with the saturation constraints imposed by the system actuator. The tuning parameters for the GPC are $N = 60$, $N_u = 10$ and a weight of control increment $\lambda = 2100$. For this tuning, the obtained M_s and D_M for the PID and the GPC were $M_{s_{PID}} = 1.8$, $D_{M_{PID}} = 12$ s, $M_{s_{GPC}} = 2.1$, and $D_{M_{GPC}} = 5$ s. In addition, a second tuning of the GPC aiming a faster response than the first one, with $\lambda = 700$, is considered, resulting in $M_{s_{GPC}} = 2.54$ and $D_{M_{GPC}} = 2.15$ s.

Fig. 11 shows the experimental results for a reference step with amplitude of 14 °C at $t = 5$ s and a load disturbance step with amplitude -0.15 at $t = 200$ s. As shown, the performance of the PID and GPC with the first tuning are similar both for reference and disturbance responses ($J_{PID} = 108.2$ and $J_{GPC} = 114.9$). Although both responses can be considered equivalent, PID presents a control signal with smaller variability, which results in a better performance index value. It is possible to improve the response presented by GPC by using a more aggressive tuning ($J_{GPC} = 107.3$), but the performance improvement is small when compared to both the GPC with the first tuning and the PID. Furthermore, the measurement noise affects more the control signal for the second tuning.

The experimental results presented in this section are in agreement with the analysis presented in this work. They show that when a robust solution is necessary, even in cases where the process is subjected to constraints, the advantages of using a complex control strategy, such as MPC, rather than a well-tuned PID with AW are practically insignificant or do not exist.

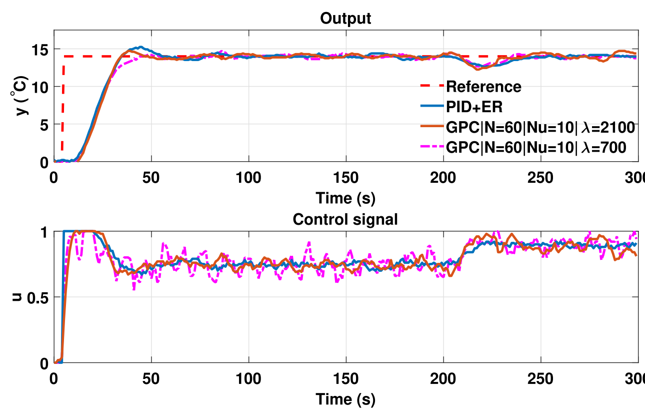


Fig. 11. Performance comparison between PID and GPC for the experimental case.

6. Conclusions

This work presented a study of PID, DTC and MPC strategies used to control SISO processes with time delay aiming to provide some ideas to facilitate the choice of the best control strategy to be used based on the characteristics of the process. Firstly it was shown that when a perfect model of the plant is available, FSP is able to provide the ideal response for dead-time processes, that is, an instantaneous response after the input delay. Using this ideal response as target, comparative analyses in terms of performance and robustness between PID and FSP for unconstrained processes, and between PID and GPC for a constrained case were presented. Furthermore, an algorithm that allows the PID to deal with several types of process constraints was proposed.

Considering the performance and robustness analysis of the PID and FSP, it was shown that in most cases both controllers present the same performance when a robust tuning is considered. On the other hand, when a good process model is available and a fast tuning (with small robustness margins) is allowed, FSP presents a significant performance improvement when compared to PID.

Moreover, an analysis which considers a constrained case was presented to show that when the system characteristics are similar to those of an industrial environment (with large modeling error and measurement noise) there is practically no advantage in using a more complex control structure, such as MPC or DTC, rather then the well-know PID with an AW technique, which is easy to implement and is also able to provide a good trade-off between performance and robustness. A performance comparison for a constrained case study with measurement noise showed that for a given M_s the performance of the PID with AW is better than the one of GPC and very similar to the one of DTC-GPC. On the other hand, when a good process model is available, more complex control structures, such as GPC and FSP, are able to provide better performance than PID, especially for disturbance rejection response.

The experimental analysis presented in Section 5 considers a FOPDT process with constraints in magnitude of the control action. The results show that PID with AW and GPC presented similar performance when a robust tuning was considered. For a fast tuning of the GPC, the performance improvement when compared to the PID was almost insignificant, since it is limited by the quality of the prediction model. On the other hand, it was observed that the control signal was more affected by measurement noise for the fast tuning.

The main conclusion that can be drawn from this work is that the choice between a more complex control strategy or a PID depends more on the model quality than on the absolute value of the dead time. For an industrial environment, where typically robust solutions are needed, a PID with AW is able to provide good or even better results than more complex strategies, such as DTC or MPC. On the other hand, when robustness is not necessary and a fast tuning is allowed, DTC and MPC structures allow substantial performance improvement over PID even for cases in which the dead time is small. Both conclusions are valid almost independently of the magnitude of the dead time if a proper PID tuning is considered.

In future work the following points will be developed: (i) the time varying delay case will be investigated, in order to compare the studied strategies for this type of processes; (ii) the proposed PID with AW will be extended for MIMO processes; and (iii) the AW strategy proposed in this study to deal with output constraints will be extended to cases with a control horizon different from one.

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.

Acknowledgments

This work was supported in part by the National Council for Scientific and Technological Development (CNPq/Brazil) under Grants 309244/2018-8 and 305785/2015-0, an in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) — under Finance Code 001 and the PrInt/UFSC Program.

References

[1] Camacho EF, Bordons C. Model predictive control. London: Springer; 2013.
[2] Normey-Rico JE, Camacho EF. Control of dead-time processes. London: Springer; 2007.
[3] Visioli A, Zhong Q. Control of integral processes with dead time. Springer Science & Business Media; 2010.
[4] Samad T. A survey on industry impact and challenges thereof. IEEE Control Syst 2017;37:17–8.
[5] Panda RC, Yu C-C, Huang H-P. PID tuning rules for SOPDT systems: Review and some new results. ISA Trans 2004;43:283–95.
[6] Normey-Rico JE, Guzmán JL. Unified PID tuning approach for stable, integrative, and unstable dead-time processes. Ind Eng Chem Res 2013;52:16811–9.
[7] Åström KJ, Hägglund T. The future of PID control. Control Eng Pract 2001;9:1163–75.
[8] Grimholt C, Skogestad S. Should we forget the Smith predictor?. In: 3rd IFAC conference on advances in proportional-integral-derivative control. Ghent, Belgium; 2018, p. 769–74.
[9] Smith JO. Closed control of loops with dead time. Chem Eng Prog 1957;53:217–9.
[10] Sanz R, García P, Albertos P. A generalized Smith predictor for unstable time-delay SISO systems. ISA Trans 2018;72:197–204.
[11] Liu T, Tian H, Rong S, Zhong C. Heating-up control with delay-free output prediction for industrial jacketed reactors based on step response identification. ISA Trans 2018;83:227–38.
[12] Kaya I. A new smith predictor and controller for control of processes with long dead time. ISA Trans 2003;42:101–10.
[13] Wang Y, Boyd S. Fast model predictive control using online optimization. IEEE Trans Control Syst Technol 2010;18:267–78.
[14] O'Donoghue B, Stathopoulos G, Boyd S. A splitting method for optimal control. IEEE Trans Control Syst Technol 2013;21:2432–42.
[15] Núñez-Reyes A, Normey-Rico JE, Bordons C, Camacho EF. A Smith predictive based MPC in a solar air conditioning plant. J Process Control 2005;15(1):1–10.
[16] Roca L, Guzman JL, Normey-Rico JE, Berenguel M, Yebra L. Robust constrained predictive feedback linearization controller in a solar desalination plant collector field. Control Eng Pract 2009;17(9):1076–88.
[17] Valencia-Palomo G, Rossiter J. Programmable logic controller implementation of an auto-tuned predictive control based on minimal plant information. ISA Trans 2011;50(1):92–100.
[18] bin Osman K, Azman MA, Suzumori K, et al. GPC controller design for an intelligent pneumatic actuator. Procedia Eng 2012;41:657–63.
[19] Vazquez S, Rodriguez J, Rivera M, Franquelo LG, Norambuena M. Model predictive control for power converters and drives: Advances and trends. IEEE Trans Ind Electron 2017;64(2):935–47.
[20] Ipoum-Ngome PG, Mon-Nzongo DL, Song-Manguelle J, Flesch RCC, Jin T. Optimal finite state predictive direct torque control without weighting factors for motor drive applications. IET Power Electron 2019;12(6):1434–44.
[21] Cavanini L, Cimini G, Ippoliti G. Model predictive control for pre-compensated power converters: application to current mode control. J Franklin Inst B 2019;356(4):2015–30.
[22] Rivera DE, Morari M, Skogestad S. Internal model control: PID controller design. Ind Eng Chem Process Design Dev 1986;25:252–65.
[23] Sha'aban YA, Lennox B, Laurí D. PID versus MPC performance for SISO dead-time dominant processes. In: 10th IFAC international symposium on dynamics and control of process systems. Mumbai, India; 2013, p. 241–6.
[24] Salem F, Mosaad MI. A comparison between MPC and optimal PID controllers: Case studies. In: Michael faraday IET international summit. Kolkata, India; 2015, p. 59–65.
[25] Sudibyo, Iqbal I, Murat M, Aziz N. Comparison of MIMO MPC and PI decoupling in controlling methyl tert-butyl ether process. Comput Aided Chem Eng 2012;31:345–9.
[26] Flesch RC, Santos TL, Normey-Rico JE. Unified approach for minimal output dead time compensation in MIMO non-square processes. In: 51st annual conference on decision and control. Maui, Hawaii; 2012, p. 2376–81.
[27] Normey-Rico JE. PID control of dead-time processes: robustness, dead-time compensation and constraints handling. In: 3rd IFAC conference on advances in proportional-integral-derivative control PID (plenary of May 10). Ghent, Belgium; 2018.
[28] Normey-Rico JE, Camacho EF. Simple robust dead-time compensator for first-order plus dead-time unstable processes. Ind Eng Chem Res 2008;47:4784–90.
[29] Palmor Z. Stability properties of Smith dead-time compensator controllers. Internat J Control 1980;32:937–49.
[30] Santos TL, Limon D, Normey-Rico JE, Alamo T. On the explicit dead-time compensation for robust model predictive control. J Process Control 2012;22:236–46.
[31] Santos TL, Limon D, Normey-Rico JE, Raffo GV. Dead-time compensation of constrained linear systems with bounded disturbances: output feedback case. IET Control Theory Appl 2013;7:52–9.
[32] Lima DM, Santos TLM, Normey-Rico JE. Robust nonlinear predictor for dead-time systems with input nonlinearities. J Process Control 2015;27:1–14.
[33] Horowitz I. Some properties of delayed controls (Smith regulator). Internat J Control 1983;38:977–90.
[34] Gudin R, Mirkin L. On the delay margin of dead-time compensators. Internat J Control 2007;80:1316–32.
[35] Garpinger O, Hägglund T. Software-based optimal PID design with robustness and noise sensitivity constraints. J Process Control 2015;33:90–101.
[36] Clarke DW, Mohtadi C, Tuffs P. Generalized predictive control—Part I. The basic algorithm. Automatica 1987;23:137–48.
[37] De Keyser R, Ionescu CM. The disturbance model in model based predictive control. In: IEEE conference on control applications. 2003, p. 446–51.

[38] Castano JA, Hernandez A, Li Z, Tsagarakis NG, Caldwell DG, De Keyser R. Enhancing the robustness of the EPSAC predictive control using a singular value decomposition approach. Robot Auton Syst 2015;74:283–95.
[39] Flesch RC, Normey-Rico JE, Flesch CA. A unified anti-windup strategy for SISO discrete dead-time compensators. Control Eng Pract 2017;69:50–60.
[40] Silva LR, Flesch RC, Normey-Rico JE. Analysis of anti-windup techniques in PID control of processes with measurement noise. In: 3rd IFAC conference on advances in proportional-integral-derivative control. Ghent, Belgium; 2018, p. 948–53.
[41] Scokaert PO, Rawlings JB. Feasibility issues in linear model predictive control. AIChE J 1999;45:1649–59.