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

PID algorithm based on GPC for second-order models with input and output constraints ⋆

Lucian Ribeiro da Silva * **Rodolfo César Costa Flesch** † **Julio Elias Normey-Rico** †

* *Programa de Pós-Graduação em Engenharia de Automação e Sistemas, Universidade Federal de Santa Catarina, 88040-900 Florianópolis, SC, Brazil (e-mail:lucian.ribeiro@ufsc.br).*
† *Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina, 88040-900 Florianópolis, SC, Brazil (e-mail: rodolfo.flesch@ufsc.br, julio.normey@ufsc.br)*

Abstract: This paper presents a novel constrained PID algorithm that performs almost the same as a constrained GPC tuned with a unit control horizon and an output soft constraint. The proposed approach computes the PID tuning parameters using the unconstrained GPC solution for first or second-order process models and uses a geometric approach, which avoids the use of an optimiser, to compute the final control action combining the PID tuning with a constraint-mapping law. Simulation results are used to illustrate the simplicity and good performance that can be obtained with the proposed approach, which is an interesting solution for PID applications in which output constraints are considered and low computation time of the control law is required.

Keywords: PID control, model predictive control, constraints, optimisation

1. INTRODUCTION

Most processes in industry have physical constraints (Camacho and Bordons, 2013). The most common types of constraints are saturation in the input variable and in the process output. Typical examples of input constraints are mechanical limits on servomotor position, on valve position, and on voltage from a power amplifier. Such actuator saturations will appear in any linear control loop on a real plant (Wang, 2020). Examples of output constraints include limits in tank level control, temperature in boilers, flow and dynamic overpressure in hydraulic systems, among others (Camacho and Bordons, 2013). These limits reflect issues such as safety requirements or are there to prevent excessive maintenance to system components (Hewing et al., 2020).

Designing a controller in such a way that all the process constraints are respected is very important for practical applications. Model Predictive Control (MPC) is a versatile method that integrates constraint handling into its formulation during the design phase. This approach uses a process model to predict future outputs and uses this information to calculate a control sequence that minimises a cost function, allowing for the consideration of constraints in the process input and output (Camacho and Bordons, 2013). In the MPC framework, input constraints can, in general, be considered as hard constraints, but output constraints are treated as soft constraints, considering a slack variable, in order to ensure feasibility. Then, an extra term is added in the MPC cost function, considering a quadratic term in the slack variable, so that any violation of the constraints is penalised. This strategy allows the controller to use a non null value of the slack variable in the transients, when this condition is necessary to avoid infeasibility, and, in the long run, the penalising term in the objective function will take the auxiliary variable to zero (Camacho and Bordons, 2013). Although MPC has several advantages and can be considered the most used advanced control strategy in practice, it should be noted that MPC imposes heavy computational burden for on-line computations when constraints are considered, thereby limiting its applicability to systems with slow dynamics or requiring high-cost computational infrastructure for implementation (Silva et al., 2020).

On the other hand, conventional proportional-integral-derivative (PID) controllers are widely used in industry, have fast computation of its control action, and can deal with many types of processes in practice, thus achieving a good compromise between performance and computational cost (Silva et al., 2020). Moreover, several methods for PID controllers to deal with input constraints are presented in the literature, mainly focused on anti-windup strategies or reference governor ideas (Wang, 2020; Lawrence et al., 2020). However, considering the PID framework, there are only few studies on how to deal with output constraints. Aboelhassan et al. (2020) present an adaptation of PID controllers for a MIMO system with input and output constraints based on an MPC controller.

⋆ The authors thank the Brazilian agencies CNPq and CAPES that have partially funded the research under projects: CNPq 403949/2021-1, CNPq 406477/2022-1, CNPq 304032/2019-0, and CAPES/Print-UFSC

The proposed algorithm is executed in a hierarchical structure of two levels: one to implement PID controllers and the other to identify the PID controller gains using the recursive least squares identification method in order to achieve a similar performance as an MPC. In Konstantopoulos and Baldivieso-Monasterios (2020), a nonlinear PID control approach is introduced to achieve reference tracking and ensure that a desired output remains below a specified threshold for a broad spectrum of nonlinear systems. An analytical method is presented for selecting controller gains to ensure closed-loop system stability. Additionally, the ultimate boundedness theory is used to demonstrate that the proposed PID maintains the output within a specified bound. Another approach, simpler than the previous ones, and that can be used with PID controllers, is the one based on override control (Imani and Montazeri, 2020; Kumar et al., 2022), which has a structure closely related to anti-windup strategies. The idea behind override control is to design an extra loop whose control action is added to the main controller in such a way that, when the controlled variable violates the limits, the control is changed to bring this output below its limit again. Although these strategies give good results and are simple, they do not offer the optimal solution.

A very interesting approach is to use the MPC ideas to propose a constrained PID algorithm capable of dealing with stable and unstable processes modelled by simple transfer functions. As shown in Camacho and Bordons (2013), an unconstrained Generalised Predictive Control (GPC) can be posed as a PID controller when the process dynamics is modelled by a second order model. In this case, the PID tuning is done in a predictive manner, based on the ideas underlying MPC. To consider process constraints in PID, a simple procedure, based on a constraint-mapping approach, was recently proposed in (Silva et al., 2020). In this approach, the input, incremental-input, and output constraints are mapped to a simple input constraint, and an anti-windup strategy is used in the PID controller to obtain a constrained PID that emulates the solution of the constrained GPC tuned with a unit control horizon ($N_u = 1$). The disadvantage of this approach it that the output constraint is considered as a hard constraint and all the advantages of the soft constraint approach used in the MPC context are missing. Therefore, to design a constrained PID algorithm that performs almost equal to a constrained GPC tuned with $N_u = 1$ and an output soft constraint, this paper proposes the following approach: (i) to compute the PID tuning parameters using the unconstrained GPC solution considering a second order model of the process; (ii) to formulate the optimisation problem of the constrained GPC based on the two decision variables of the problem: $u(k)$ (the control action to be applied to the process) and $\epsilon(k)$ (the slack variable of the soft constraint, considered constant over the prediction horizon); (iii) to compute the optimal solution using a geometric approach, that avoids the need for an optimiser; (iv) to compute the final control action combining the PID tuning with a control governor law that achieves the optimal solution with low computation time.

The rest of the paper is organised as follows. In Section 2 the simple constrained GPC solution for second-order processes is presented. Section 3 describes the geometric procedure to obtain the solution of the GPC optimisation problem. In Section 4 the proposed control algorithm for the constrained PID is presented, and in Section 5 simulation results illustrate the advantages of the new constrained PID algorithm. The paper ends with some conclusions, in Section 6.

2. CONSTRAINED GPC FOR SECOND ORDER MODELS

In this paper, a discrete-time second-order CARIMA model, given by:

$$(1+a_1z^{-1}+a_2z^{-2})y(k) = (b_0+b_1z^{-1})u(k-1)+\frac{\eta(k)}{\Delta} \quad (1)$$

is used by GPC to predict the plant future outputs, which are used to calculate the control action by minimising a cost function (Camacho and Bordons, 2013). In (1), $u(k)$ and $y(k)$ are the input and output of the process, k is the discrete time in samples, z^{-1} is the back-shift operator, $\eta(k)$ is a zero-mean white noise, and $\Delta = (1 - z^{-1})$. The function to be minimised is given by:

$$J_{\mathrm{MPC}} = \sum_{j=1}^{N} [\hat{y}(k+j|k) - r(k)]^2 + \lambda[\Delta u(k)]^2, \quad (2)$$

where N is the prediction horizon, the control horizon is 1, λ is the control increment weight, $\hat{y}(k+j|k)$ is the predicted output for $k + j$ at time instant k, $r(k)$ is the reference signal considered constant in the prediction horizon, and $\Delta u(k)$ is the control increment.

For this particular case, the cost function can be written as:

$$J_{\mathrm{MPC}} = \frac{H}{2}[\Delta u(k)]^2 + b\Delta u(k) + c_0, \quad (3)$$

where $H = 2(\mathbf{g}^T\mathbf{g}+\lambda)$, $b = 2(\mathbf{f}(k)-\mathbf{1}r(k))^T\mathbf{g}$, $c_0 = (\mathbf{f}(k)-\mathbf{1}r(k))^T(\mathbf{f}(k) - \mathbf{1}r(k))$, $\mathbf{g}$ is the vector with the N step response coefficients of the process, $\mathbf{1} \in \mathbb{R}^N$ is a vector of ones, and $\mathbf{f}(k)$ is the free response vector (Camacho and Bordons, 2013).

An analytical solution that minimises J_{MPC}, for the unconstrained case, can be obtained differentiating J_{MPC} with respect to $\Delta u(k)$ and equating it to zero. Thus, the obtained $\Delta u(k)$ is $\Delta u^\star_{\mathrm{UC}}(k) = -\frac{b}{H}$, calculated as (Normey-Rico and Camacho, 2007):

$$\Delta u^\star_{\mathrm{UC}}(k) = \sum_{j=1}^{3} l_{y_j}y(k-j)+l_{u_1}\Delta u(k-1)+\sum_{i=1}^{N} v_i r(k), \quad (4)$$

where the coefficients l_{y_j}, l_{u_1}, and v_i can be calculated from the model parameters, the prediction horizon, N, and the control increment weight, λ. That is, the unconstrained-GPC control law is equivalent to a 2-DOF PID structure with a controller $C_{\mathrm{PID}}(z)$ and a reference filter $F_{\mathrm{PID}}(z)$ given by:

$$C_{\mathrm{PID}}(z) = -\frac{l_{y_1} + l_{y_2}z^{-1} + l_{y_3}z^{-2}}{(1 - z^{-1})(1 - l_{u_1}z^{-1})}, \quad (5)$$

$$F_{\mathrm{PID}}(z) = \frac{\sum_{i=1}^{N} v_i}{l_{y_1} + l_{y_2}z^{-1} + l_{y_3}z^{-2}}, \quad (6)$$

where $F_{\mathrm{PID}}(z)$ has static gain equal to 1. In case a first order transfer function is used to model the process, the equivalent controller is a PI plus a reference filter, just using $l_{y_3} = 0$ and $l_{u_1} = 0$ in $C_{\mathrm{PID}}(z)$ and $F_{\mathrm{PID}}(z)$.

If constraints in the amplitude and increment of the control signal and in the amplitude of the process output are considered, the objective function in (2) must be minimised taking into account these constraints, which can be mathematically represented as:

$$\begin{aligned}&\mathbf{1}(y_{\min}-\epsilon(k)) \le \mathbf{g}\Delta u(k)+\mathbf{f} \le \mathbf{1}(y_{\max}+\epsilon(k)),\\ &u_{\min} \le \Delta u(k)+u(k-1) \le u_{\max},\\ &\Delta u_{\min} \le \Delta u(k) \le \Delta u_{\max},\end{aligned} \tag{7}$$

where $\epsilon(k)$ is a slack variable, introduced to prevent infeasible solutions. Simultaneously, the objective function in (2) must incorporate $\epsilon(k)$ as a decision variable and λ_ϵ as a penalising weight, resulting in:

$$J_{\mathrm{MPC}} = \frac{H}{2}[\Delta u(k)]^2 + b\Delta u(k) + c_0 + \lambda_\epsilon[\epsilon(k)]^2. \tag{8}$$

Since both $\Delta u(k)$ and $\epsilon(k)$ are decision variables, the cost function can be expressed in the standard form with a new variable $\mathbf{w}$ as:

$$J_{\mathrm{MPC}_\epsilon} = \frac{1}{2}\mathbf{w}^T\mathbf{H}_w\mathbf{w} + \mathbf{b}_w^T\mathbf{w} + c_0, \tag{9}$$

where

$$\mathbf{w} = \begin{bmatrix}\Delta u(k)\\ \epsilon(k)\end{bmatrix}, \mathbf{H}_w = \begin{bmatrix}H & 0\\ 0 & 2\lambda_\epsilon\end{bmatrix}, \text{and } \mathbf{b}_w^T = [b \;\; 0]. \tag{10}$$

Thus, the constraints described in (7) can be represented in a simplified form as:

$$\mathbf{A}_c\mathbf{w} \le \mathbf{b}_c, \tag{11}$$

with

$$\mathbf{A}_c = \begin{bmatrix}\mathbf{g} & -\mathbf{1}\\ -\mathbf{g} & -\mathbf{1}\\ 1 & 0\\ -1 & 0\\ 1 & 0\\ -1 & 0\end{bmatrix}, \; \mathbf{b}_c = \begin{bmatrix}\mathbf{1}y_{\max}-\mathbf{f}\\ -\mathbf{1}y_{\min}+\mathbf{f}\\ (u_{\max}-u(k-1))\\ -(u_{\min}-u(k-1))\\ \Delta u_{\max}\\ -\Delta u_{\min}\end{bmatrix}. \tag{12}$$

Finally, to compute the optimal control action, the minimisation of a quadratic function with affine constraints given by:

$$\begin{aligned}\min \quad & J_{\mathrm{MPC}\epsilon}\\ \text{s.t.} \quad & \mathbf{A}_c\mathbf{w} \le \mathbf{b}_c\end{aligned} \tag{13}$$

can be done using any QP solver * . However, this approach can require a long computation time, and is unappropriated for many practical cases. Alternatively, the optimal solution can be obtained through a geometric approach, as detailed in the next section.

3. GEOMETRIC INTERPRETATION OF THE QP PROBLEM

Consider the cost $J_{\mathrm{MPC}_\epsilon}$ in (9). The values of $\mathbf{w}$ that make $J_{\mathrm{MPC}_\epsilon} = c$, where c is a positive real constant, define ellipses in the plane $[\Delta u(k),\ \epsilon(k)]$, with centre at $\mathbf{w}^\star_{\mathrm{UC}} = -\mathbf{H}_w^{-1}\mathbf{b}_w$ (which represents the optimal solution for the unconstrained case, and is obtained with $\epsilon = 0$). Note that, as $\mathbf{H}_w$ is diagonal, the ellipses are aligned with the $\Delta u(k)$ or ϵ axis, depending on the values of H and λ_ϵ.

For the constrained case, the minimisation of (13) may be interpreted as finding the point at which the smallest ellipse touches the polygon of the feasible region (Seron et al., 2000). For a better illustration of the previous statement, Fig. 1 shows the geometric representation of the optimal solution of the QP problem considering constraints in the process output (using a slack variable) and the increment of control action, where $\mathbf{w}^\star_{\mathrm{UC}}$ is the optimal solution for the unconstrained case and $\mathbf{w}^\star$ is the optimal solution for the constrained case. In this figure, a hypothetical case is drawn, showing only the constraints that define the feasible region.

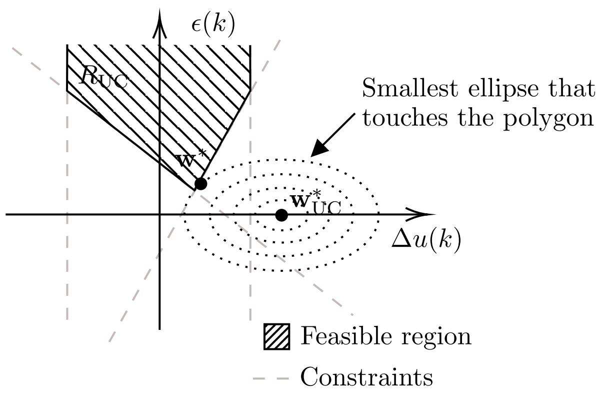

Fig. 1. Geometric interpretation of the optimisation problem for the constrained case.

Note that it is possible to simplify the analysis applying a simple linear transformation T so that the ellipses become circumferences by adjusting only their scaling in the ϵ axis. Thus, using:

$$\begin{aligned}&T : \mathbb{R}^2 \to \mathbb{R}^2\\ &T(\mathbf{w}) = \tilde{\mathbf{w}} = \mathbf{H}_T\mathbf{w},\end{aligned} \tag{14}$$

where

$$\mathbf{H}_T = \begin{bmatrix}1 & 0\\ 0 & \sqrt{\dfrac{H}{2\lambda_\epsilon}}\end{bmatrix}, \tag{15}$$

to map the coordinate system from $\mathbf{w}$ to $\tilde{\mathbf{w}} = [\Delta u(k)\ \tilde{\epsilon}(k)]$, the values of $\tilde{\mathbf{w}}$ (in the $\Delta u(k) \times \tilde{\epsilon}$ plane) that give constant $J_{\mathrm{MPC}_\epsilon}$ are located in circumferences with centre at $\tilde{\mathbf{w}}^\star_{\mathrm{UC}} = \mathbf{H}_T\mathbf{w}^\star_{\mathrm{UC}} = -\mathbf{H}_T\mathbf{H}_w^{-1}\mathbf{b}_w$ (which represents the optimal solution for the unconstrained case in the transformed coordinates).

In the modified coordinates, the constrained minimisation problem involves locating the smallest circumference that intersects one of the edges of the polygon, which is equivalent to identifying the edge that provides the minimum Euclidean distance to $\tilde{\mathbf{w}}^\star_{\mathrm{UC}}$. If the line that defines the edge with the closest point to $\tilde{\mathbf{w}}^\star_{\mathrm{UC}}$, in the transformed coordinates, is given by the i^{th} constraint, it can be expressed as:

$$\tilde{\epsilon}_i(k) = \alpha_i\Delta u(k) + \beta_i, \tag{16}$$

where

$$\alpha_i = -\frac{\tilde{a}_{i1}}{\tilde{a}_{i2}}, \qquad \beta_i = \frac{b_{c_i}}{\tilde{a}_{i2}}, \tag{17}$$

$\tilde{a}_{ij}$ is the j^{th} element in the i^{th} row of matrix $\tilde{\mathbf{A}}_c = \mathbf{A}_c\mathbf{H}_T$, and b_{c_i} is the i^{th} element of vector $\mathbf{b}_c$. The projection of $\tilde{\mathbf{w}}^\star_{\mathrm{UC}}$ to the line in (16), denoted as $P_{\tilde{\epsilon}_i(k)}(\tilde{\mathbf{w}}^\star_{\mathrm{UC}})$, can be obtained as (Boyd and Vandenberghe, 2004):

$$P_{\tilde{\epsilon}_i(k)}(\tilde{\mathbf{w}}^\star_{\mathrm{UC}}) = \tilde{\mathbf{w}}^\star = \boldsymbol{\epsilon}_0 + \frac{(\tilde{\mathbf{w}}^\star_{\mathrm{UC}} - \boldsymbol{\epsilon}_0)\cdot\mathbf{v}}{||\mathbf{v}||^2}\mathbf{v}, \tag{18}$$

* Note that other output constraints (such as monotonic response and overshoot constraints) can be also represented with the presented form.

where $\mathbf{v}$ represents the direction vector of the line $\tilde{\epsilon}_i(k)$, $\boldsymbol{\epsilon}_0$ represents any point in the line $\tilde{\epsilon}_i(k)$ and $\cdot$ represents dot product. MPC techniques exclusively apply the first increment of the control action, $\Delta u^\star(k)$, to the plant. Given that $\tilde{\mathbf{w}}^\star = [\Delta u^\star(k) \quad \tilde{\epsilon}^\star(k)]$ and choosing $\mathbf{v} = [1 \quad \alpha_i]$ and $\boldsymbol{\epsilon}_0 = [1 \quad \alpha_i + \beta_i]$ for convenience, it is possible to compute only the first element in (18), which leads to:

$$\Delta u^\star(k) = \frac{\Delta u^\star_{\mathrm{UC}}(k) - \alpha_i\beta_i}{\gamma_i} = -\frac{b + H\alpha_i\beta_i}{H\gamma_i}, \tag{19}$$

where

$$\gamma_i = 1 + \alpha_i^2. \tag{20}$$

As can be observed in (19), the optimal constrained solution, $\Delta u^\star(k)$, is a function of the optimal unconstrained solution, $\Delta u^\star_{\mathrm{UC}}$, and other terms that depend on the values of α_i and β_i. Note that only β_i and $\Delta u^\star_{\mathrm{UC}}$ are computed online, since the values of α_i are constant.

The presented solution is valid for any MPC formulation that results in a QP problem with any kind or combination of input and output amplitude constraints, considering any size of the prediction horizon N (with a constant slack variable, $\epsilon(k)$, over the horizon) and $N_u = 1$. Furthermore, the problem can be easily generalised to accommodate various forms of output constraints, such as ensuring a monotonic response, eliminating overshoot and undershoot, or imposing other conditions on the future values of the process output. However, determining the line onto which the unconstrained solution $\tilde{\mathbf{w}}_{\mathrm{UC}}$ must be projected is not an easy task because the number of lines defined by the output constraints increases linearly with the size of the prediction horizon and not all of them define the edges of the feasible region. Nevertheless, a simple method to obtain a solution which is very close to the one provided by the GPC is presented in the next section.

4. SUBOPTIMAL SOLUTION FOR THE CONSTRAINED QP PROBLEM

When amplitude constraints are considered, the feasible region changes from iteration to iteration, so it is not possible to determine the edges of the feasible region offline. Instead, this process must be done at each sample, which typically makes use of another optimisation problem (Boyd and Vandenberghe, 2004). This section presents a suboptimal solution for the QP problem that does not require any numerical solver or procedure to determine the edges of the feasible region. The idea comes from the discussion in Silva et al. (2020), where it is shown that for problems considering a control horizon of one sample, $N_u = 1$, when increment and/or magnitude control constraints are violated, the optimal constrained solution can be obtained by saturating the optimal unconstrained solution considering the upper and lower limits as the most rigid constraint for increment and magnitude of control action, which means:

$$\Delta u^\star(k) = \mathrm{sat}(\Delta u^\star_{\mathrm{UC}}(k), U_{g_{\min}}, U_{g_{\max}}), \tag{21}$$

where

$$\begin{aligned} U_{g_{\min}} &= \max(\Delta u_{\min}, u_{\min} - u(k-1)), \\ U_{g_{\max}} &= \min(\Delta u_{\max}, u_{\max} - u(k-1)). \end{aligned} \tag{22}$$

The proposed method consists of verifying and correcting the optimal unconstrained solution, $\tilde{\mathbf{w}}^\star_{\mathrm{UC}}(k)$, only if any output constraints are violated. If a violation in the process output is detected, the optimal unconstrained solution can be projected onto the line defined by:

$$\tilde{\epsilon}_i(k) = \alpha_{i_{\max}}\Delta u(k) + \beta_{i_{\max}},$$

where $i_{\max}$ represents the most rigid output constraint for $\Delta u^\star_{\mathrm{UC}}(k)$, i.e., the line which has the maximum value of $\tilde{\epsilon}(k)$ when evaluated at $\Delta u^\star_{\mathrm{UC}}(k)$. This process is illustrated in Fig. 2 for the same case discussed in Fig. 1, but now in the transformed coordinates. The most rigid constraint can be identified by locating the uppermost line in a vertical direction from the optimal unconstrained solution, identified by the point $\tilde{\mathbf{w}}^\star_{\mathrm{UC}}(k)$. After that, $\tilde{\mathbf{w}}^\star_{\mathrm{UC}}(k)$ is projected onto this line in the direction of the least Euclidean distance to obtain the proposed suboptimal solution, $\tilde{\mathbf{w}}^\star_{\mathrm{SO}}(k)$. Since the linear transformation T in (14) does not scale the $\Delta u(k)$ axis, the suboptimal value of $\Delta u(k)$ is obtained by just checking the first element of $\tilde{\mathbf{w}}^\star_{\mathrm{SO}}(k)$. In this particular case illustrated in Fig. 2, the solution is exactly the same one obtained by a regular optimisation algorithm, i.e., the proposed solution is optimal.

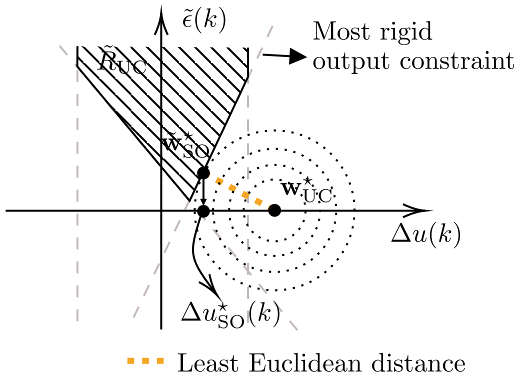

Fig. 2. Geometric interpretation of the minimisation problem with constraints in the transformed coordinates.

Even though the proposed method can provide a computationally efficient solution for the problem, the solution is not always optimal. Suboptimality occurs either when the line that contains the most rigid output constraint does not define an edge of the feasible region or when the slopes and intercepts of the lines are such that there is another point over the edge which has a lower Euclidean distance to $\tilde{\mathbf{w}}^\star_{\mathrm{UC}}(k)$ than the projected one.

Once the suboptimal control increment is obtained considering only the soft output constraints, it can be considered as the optimum value in (21) to take the control rate and magnitude constraints into account. By following this procedure, the resulting controller can deal with amplitude constraints on both the manipulated and the process variables, as well as with rate constraints on the control signal.

As shown in Section 2, the unconstrained GPC control law, when considering first or second-order models, can be represented as a PID structure. Thus, the main idea of the proposed method is to compute the optimal unconstrained solution, $\Delta u^\star_{\mathrm{UC}}$, by using the GPC-based PID controller and then applying the method discussed previously. The method described earlier can be presented in a pseudo-code form, as shown in Algorithm 1.

Algorithm 1: PID algorithm for processes with input and output constraints

initialise variables;
compute α_i for $i = 1$ to $2N$ using Equation (17);
compute γ_i for $i = 1$ to $2N$ using Equation (20);
repeat
  compute $\Delta u^{\star}_{\mathrm{UC}}(k)$ using GPC-based PID;
  compute $U_{g_{\min}}$ and $U_{g_{\max}}$ using Equation (22);
  $\tilde{\epsilon}_{\max} \leftarrow -10^{10}$;
  for $i = 1$ **to** $2N$ **do**
    $\tilde{\epsilon}_i(k) \leftarrow \alpha_i \Delta u^{\star}_{\mathrm{UC}}(k) + \beta_i$;
    if $\tilde{\epsilon}_i(k) > \tilde{\epsilon}_{\max}$ **then**
      $\tilde{\epsilon}_{\max} \leftarrow \tilde{\epsilon}_i(k)$;
      $i_{\max} \leftarrow i$;
    end
  end
  if $\tilde{\epsilon}_{\max} > 0$ **then**
    $\Delta u^{\star}_{\mathrm{SO}}(k) \leftarrow \frac{\Delta u^{\star}_{\mathrm{UC}}(k) - \alpha_{i_{\max}} \beta_{i_{\max}}}{\gamma_{i_{\max}}}$;
    $\Delta u_{\mathrm{aux}} \leftarrow \Delta u^{\star}_{\mathrm{SO}}(k)$;
  else
    $\Delta u_{\mathrm{aux}} \leftarrow \Delta u^{\star}_{\mathrm{UC}}(k)$;
  end
  $\Delta u(k) \leftarrow \mathrm{sat}(\Delta u_{\mathrm{aux}}, U_{g_{\min}}, U_{g_{\max}})$;
  $u(k) \leftarrow u(k-1) + \Delta u(k)$;
  apply $u(k)$ to the plant;
  update the variables;
  $k \leftarrow k + 1$;
  wait T_s;
until *controller is stopped*;

5. CASE STUDY

In order to evaluate the proposed approach, a case study consisting of the control of a second-order non-minimum phase process given by:

$$P(s) = \frac{-0.8s + 1}{(1.5s + 1)^2} \tag{23}$$

is considered. This type of process was chosen for the case study because the GPC optimisation problem becomes infeasible when a constraint aimed at preventing the inverse response, caused by the zero outside the unit circle, is included without the consideration of a slack variable, making this case study more challenging than other types of processes. The discrete-time process model obtained using a zero-order hold and a sampling time of $T_s = 0.1\,\mathrm{s}$ is:

$$P(z) = \frac{-0.031z + 0.035}{z^2 - 1.871z + 0.875}. \tag{24}$$

The proposed approach was compared with the original GPC and a PID with constraints mapping, presented in (Silva et al., 2020). For all cases, the controller tuning parameters were set as: $N = 20$, $N_u = 1$, and $\lambda = 0$. The slack variable penalising weight used for the GPC and the proposed PID was $\lambda_\epsilon = 1000$. The simulation of the closed-loop system considers constraints on increment and magnitude of control action and on the process output as: $\Delta u_{\min} = -0.5$, $\Delta u_{\max} = 0.5$, $u_{\min} = 0$, $u_{\max} = 0.9$, $y_{\min} = 0$, and $y_{\max} = 0.7$. The PID controller and the reference filter obtained based on the GPC-based tuning are:

$$\begin{aligned} C_{\mathrm{PID}}(z) &= \frac{379.25(1 - 0.975z^{-1})(1 - 0.867z^{-1})}{(1 + 12.94z^{-1})(1 - z^{-1})}, \\ F_{\mathrm{PID}}(z) &= \frac{0.003}{(1 - 0.975z^{-1})(1 - 0.867z^{-1})}. \end{aligned} \tag{25}$$

Fig. 3 shows the simulation of the closed-loop system, comparing the performance of the three approaches. The simulation considers a step reference of amplitude 0.5 at $t = 1\,\mathrm{s}$, a load disturbance step of amplitude -0.1 at $t = 30\,\mathrm{s}$ and a final change in the reference considering a step of amplitude 0.3 at $t = 45\,\mathrm{s}$. From the results, it is clear that that GPC-based PID with mapping was unable to mitigate the inverse response, which leads to violating the constraint on the lower bound of the process variable. Furthermore, the same method was unable to avoid the violation of the upper bound constraint, when a change in the reference occurred towards the end of the simulation. Both the GPC and the proposed PID presented very similar responses for reference tracking and were able to provide responses which slightly violate the lower bound constraint to avoid infeasibility while successfully preventing a violation of the upper bound constraints by the end of the simulation. In terms of disturbance rejection, all the controllers exhibited exactly the same performance.

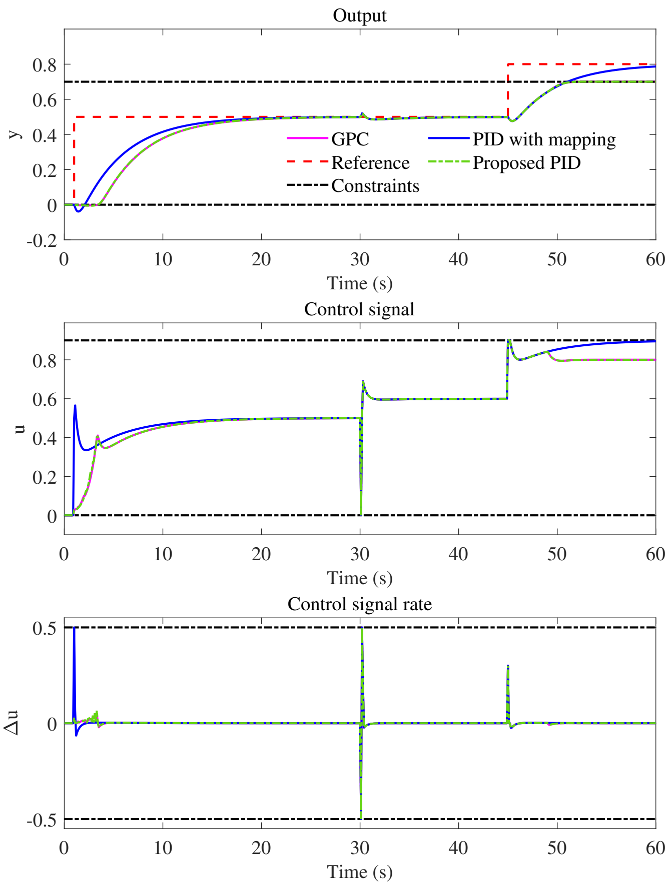

Fig. 3. Performance comparison between the GPC, PID with mapping, and the proposed PID.

In order to quantify the similarity between both PID controllers and the original GPC, Table 1 shows the integral of $J_{\mathrm{MPC}_\epsilon}$ over the set-point tracking interval, from $t = 1\,\mathrm{s}$ to

$t = 25\,\text{s}$. This timeframe chosen for integration highlights the differences of the controllers when the lower bound output constraint violation occurs. The other windows were not considered because the disturbance rejection responses are the same and the last reference change increase too much the cost associated with the PID with mapping, since the constraint is violated for a long period of time.

Table 1. Integral of cost function J_{MPC_ϵ} for reference tracking.

Method	$J_{\mathbf{MPC}_\epsilon}$
GPC	0.10
Proposed PID	0.15
PID with mapping	4.76

As shown in Table 1, the proposed PID exhibits a cost which is similar to the one of the GPC. In contrast, the PID with mapping demonstrates a higher cost value, mainly attributed to the penalisation for the violation of output constraints after the first reference change.

Even though the responses of the GPC and the proposed PID are quite similar, the computational effort required by the GPC is significantly higher. The simulations were executed on a computer equipped with an Intel Core i5 processor operating at 2.0 GHz and 16 GB of RAM. Execution time measurements were obtained using the tic-toc command provided by MATLAB®. Table 2 presents a comparative analysis of mean and maximum (worst-case) execution times considering both controllers. When considering both average and worst-case execution times, the proposed PID outperforms GPC by a factor of approximately 100. This difference can be attributed to the computational efficiency of the proposed algorithm, which requires only if statements, a single loop, and straightforward arithmetic operations. Moreover, when considering the application of GPC in practical scenarios, using a microcontroller, the proposed method exhibits notably lower implementation complexity compared to a conventional QP solver.

Table 2. Mean and worst-case execution times of GPC and proposed PID with $N = 20$ and $N_u = 1$.

Method	Mean	Worst case
GPC (*quadprog*)	$1900\,\mu s$	$5000\,\mu s$
Proposed PID	$18\,\mu s$	$43\,\mu s$

6. CONCLUSION

This paper presented a novel GPC-based PID controller capable to achieve almost optimal performance for processes with input and output constraints. This method is derived from the geometric interpretation of the QP problem associated with GPC. The resulting PID algorithm, characterised by its simplified tuning and effective constraint handling, proves efficient for controlling processes modelled by either a first or second-order transfer function. In the presented case study, considering a second-order non-minimum phase process, the proposed PID was able to deal with all the constraints considered and achieves almost the same performance as the GPC while demonstrating a execution time improvement of two orders of magnitude. Also, a comparison of the integral of the cost function shows that the proposed PID performs very closely to the constrained GPC. The proposed method emerges as a effective and powerful tool for practical applications, especially well-suited for implementation in microcontrollers with low-power computation requirements, as it avoids the need for an online optimiser. This effectiveness is primarily attributed to its inherent simplicity in implementation, making it an efficient choice in scenarios where constraints on the process variable and inputs are critical considerations.

REFERENCES

Aboelhassan, A., Abdelgeliel, M., Zakzouk, E.E., and Galea, M. (2020). Design and implementation of model predictive control based PID controller for industrial applications. *Energies*, 13(24), 6594.

Boyd, S.P. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Camacho, E.F. and Bordons, C. (2013). *Model Predictive Control*. Springer, London.

Hewing, L., Wabersich, K.P., Menner, M., and Zeilinger, M.N. (2020). Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3, 269–296.

Imani, A. and Montazeri, M. (2020). Stability analysis of override logic system containing state feedback regulators and its application to gas turbine engines. *European Journal of Control*, 52, 97–107.

Konstantopoulos, G.C. and Baldivieso-Monasterios, P.R. (2020). State-limiting PID controller for a class of nonlinear systems with constant uncertainties. *International Journal of Robust and Nonlinear Control*, 30(5), 1770–1787.

Kumar, A., Samavedham, L., Karimi, I.A., and Srinivasan, R. (2022). Critical assessment of control strategies for industrial systems with input-output constraints. *Industrial & Engineering Chemistry Research*, 61(30), 11056–11070.

Lawrence, N.P., Stewart, G.E., Loewen, P.D., Forbes, M.G., Backstrom, J.U., and Gopaluni, R.B. (2020). Optimal PID and antiwindup control design as a reinforcement learning problem. *IFAC-PapersOnLine*, 53(2), 236–241.

Normey-Rico, J.E. and Camacho, E.F. (2007). *Control of Dead-time Processes*. Springer, London.

Seron, M.M., De Dona, J.A., and Goodwin, G.C. (2000). Global analytical model predictive control with input constraints. In *Proceedings of the 39th IEEE Conference on Decision and Control*, volume 1, 154–159. IEEE, Sydney, NSW, Australia.

Silva, L.R., Flesch, R.C.C., and Normey-Rico, J.E. (2020). Controlling industrial dead-time systems: When to use a PID or an advanced controller. *ISA Transactions*, 99, 339–350.

Wang, L. (2020). *PID Control System Design and Automatic Tuning Using MATLAB/Simulink*. John Wiley & Sons.