



PLC Implementation of Piecewise Affine PI Controller Applied to Industrial Systems with Constraints

Jan Erik Mont Gomery Pinto¹ · Andre Felipe Oliveira de Azevedo Dantas¹ · Andre Laurindo Maitelli¹ · Amanda Danielle Oliveira da Silva Dantas¹ · Carlos Eduardo Trabucco Dórea¹ · João Tiago Loureiro Sousa Campos¹ · Everton José de Castro Rego¹

Received: 21 August 2018 / Revised: 14 February 2019 / Accepted: 19 February 2019 / Published online: 25 February 2019
© Brazilian Society for Automatics–SBA 2019

Abstract

In this paper, the design of a piecewise affine proportional integral (PWA-PI) controller algorithm based on invariant set and multiparametric programming for constrained systems is proposed. We implemented the algorithm in a programmable logic controller (PLC) to control an industrial constrained level plant and analyze its behavior. Structured text routines were programmed and validated while controlling two systems with PLC. The results show that the constraints on the error, integral of the error, system output and control action are respected because PWA-PI controllers are tuned from the solution of an optimization problem. The evaluated performance indexes (such as mean square error, Goodhart, overshoot and settling time) show that PWA-PI can be adjusted for better performance than proportional integral (PI) controller tuned by Ziegler–Nichols (Z–N) rules. In the analyzed cases, a settling time of 108 s was obtained, whereas PI controller with Z–N rules presented a 179 s settling time. All of the analyzed performance indexes that we used to evaluate both controllers show PWA-PI as a better controller for constrained systems.

Keywords PI controller · Multiparametric programming · Constrained systems · Invariant sets · Industrial piecewise affine controller · PLC

1 Introduction

PID controllers are widely used in industrial control applications. Over 50% of all applications use this technique (Puangdownreong 2018). This controller is used, for example, in level (Monteiro et al. 2017), flow (Rai et al. 2017; Allad and Acheli 2018), temperature (Homod 2018) and unmanned autonomous vehicle (UAV) control systems (Sendoya-Losada and Quintero-Polanco 2018). Nowadays, all PLC have PID built-in blocks implemented in software. PID controllers are easy to tune. (There are only three parameters to tune.) They also have well-known techniques for automatic tuning, such as Ziegler and Nichols (1942), Cohen and Coon (1953) and many others (O'Dwyer 2006). These techniques are widely used in industry and are effective to adjust the controller to handle single-input single-output

(SISO) systems. Also, there are methods that even tune it for multiple-input multiple-output (MIMO) coupled systems (Jiao et al. 2008). However, most of the processes in industrial applications are subject to constraints (Tchamna and Lee 2018), which means that they are not well suited for use in PID controllers.

To overcome this issue, researchers (Huyett et al. 2015; Weber 2015) try to consider these constraints using an anti-reset windup technique, which limits the control action and the integral of the error, aiming to adapt the controller to the constrained process. However, these methods do not consider the system constrained conditions at the tuning step. Therefore, the PID-based algorithms are not appropriate to synthesize an optimal control action for these systems.

Researchers have developed many control techniques based on linear and quadratic programming (Schulze Darup and Mönnigmann 2018). Predictive controllers (Short and Abugchem 2017) assume a cost function that consider systems constraints and, at the same time, calculate the optimal control action.

✉ Jan Erik Mont Gomery Pinto
jan.pinto@unp.br

¹ DCA - LAUT, Federal University of Rio Grande do Norte, Natal, RN, Brazil

Even though these controllers are very efficient, predictive techniques have two main problems. The first problem is the computational cost associated with the online resolution of linear and quadratic programming. This difficulty can be circumvented by finding explicit control laws using controlled invariant contractive sets as the solution of a multiparametric programming problem which results in a piecewise affine (PWA) law over a polyhedral set satisfying the constraints. Additionally, new controller techniques such as predictive (Short and Abugchem 2017), fuzzy (Khan et al. 2015) and neural (Zelentsov and Denysiuk 2019) are not as frequently used in the industry as PID.

In this paper, we address the design of a PWA-PI controller based on set invariance technique and multiparametric linear programming, including its implementation in a PLC. The proposed controller is designed via the same methodology that was used to obtain explicit MPC (Xiu and Zhang 2018), but in this paper we tune affine PI controllers by considering system constraints to applying them in restricted systems. The algorithm proposed in this paper uses this approach, where we replace a linear programming problem by a multiparametric programming problem that diminishes the computational cost when performing the online control action computation (Dantas et al. 2018a). The proposed technique is compared with a controller that is tuned using the classic Ziegler and Nichols (1942) rules in an industrial plant. The results show that the proposed technique results in a better time response and the possibility of controlling overshoot by changing the setpoint. Overshoots are not a problem to the plant with the proposed technique because the operator can set constraints with different values, which allows the operator to limit it to values that do not affect the plant's operation.

This rest of this paper is organized as follows: Sect. 2 presents the PWA control idea, where PI tuning is performed using multiparametric programming and invariant sets concepts. Section 3 shows how to implement the PWA-PI controller using a PLC and structured text (ST) language. In Sect. 4, we present the results of the PLC implementation when it controls an electric system that uses resistors, inductors and capacitors (RLC) and a level pant. Finally, Sect. 5, concludes the paper.

2 PWA-PI Controllers

PWA systems are well known in control because many practical control systems have piecewise affine components (Wei et al. 2018; Nie and Zheng 2015). PWA systems can model both real systems and be designed controllers (Shorten et al. 2007). In general, when the goal is to find a PWA control law for the design of constrained controllers, the multiparametric programming technique is often used (Zhang et al. 2016). In

this case, a linear or quadratic programming problem is converted into a multiparametric programming problem (linear or quadratic), which returns as a solution of the PWA control law parameters on a set of polyhedral regions according to Eq. (1) (Gersnoviez et al. 2017).

$$\forall x(k) \in R_j, u(k) = F_j x(k) + g_j, \text{ for } j = 1, \dots, N_q, \quad (1)$$

where $x(k)$ is the state vector, $u(k)$ is the control action obtained by the PWA law, and F_j and g_j are the computed parameters corresponding to the polyhedral region R_j .

Multiparametric programming returns an optimal solution to the constraints set subdivided into polyhedral regions, where the optimal value and the optimizer are expressed as explicit functions parameters $x(k)$ (Bemporad et al. 2002). The optimal solution of the problem is defined as a PWA control law by state feedback over a set of polyhedral regions (Dantas et al. 2018a).

The proposed PI controller is presented in Eq. (1). To tune the PI parameters in similar way as explicit predictive controllers, the state vector $x(k)$ must be rewritten as a function of the error and integral of the error, thus forcing multiparametric programming to find PWA-PI controller parameters. The following steps aim to obtain these parameters. Consider that the system to be controlled is described by:

$$G(s) = \frac{b}{s+a}. \quad (2)$$

Also consider that the system presented in Eq. (2) can be controlled by a PWA-PI, as shown in Fig. 1.

To compute the PWA-PI law for the system shown in Eq. (2), it is necessary to apply state feedback controller design concepts. In this case, the setpoint (r) does not affect the result. Therefore, we can assume that $r = 0$. Let $x_1 = \int_0^t e(t)dt$ and $x_2 = e$ be the states $x = [x_1 \ x_2]^T$. Then, assume the following expressions:

$$\dot{e} = -ae - bu, \quad (3)$$

$$\frac{d}{dt} \int_0^t e(t)dt = e. \quad (4)$$

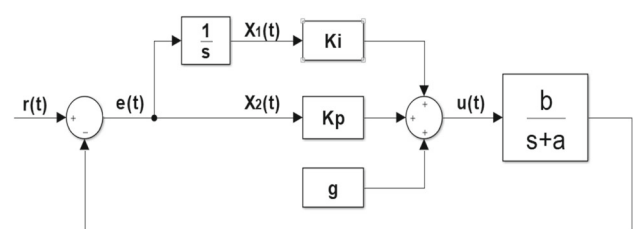


Fig. 1 Block diagram of the proposed PWA-PI controller

Then, Eqs. (3) and (4) can be rewritten in the following equivalent form:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & -a \end{bmatrix} x + \begin{bmatrix} 0 \\ -b \end{bmatrix} u. \quad (5)$$

The system can be expressed in discrete state space form as:

$$x(k+1) = Ax(k) + Bu(k); \quad (6)$$

from this, the state feedback controller parameters can be calculated by the multiparametric programming problem, parameterized in $x(k) \in \Omega = \{Gx \leq \rho\}$ (Weinkeller et al. 2012a):

$$\min_v c^T v + d^T x \quad (7)$$

Subject to: $Dv \leq W + Ex$,

where

$$c^T = [0 \quad 1], d^T = [0 \quad 0], D = \begin{bmatrix} GB & -\rho \\ U_c & 0 \end{bmatrix}, v = \begin{bmatrix} u \\ \varepsilon \end{bmatrix}, \\ W = \begin{bmatrix} 0 \\ \varphi \end{bmatrix}, E = \begin{bmatrix} GA \\ 0 \end{bmatrix},$$

where the polyhedral set of constraints defined in the state space is $\Omega = \{x : Gx \leq \rho\}$, and the control action is subject to the constraints $u(k) \in \mathbb{U} = \{u : U_c u \leq \varphi\}$. In this case, the set Ω defined in state space is a λ -contractive control invariant set where λ is the contraction rate ($0 < \lambda < 1$) which determines the convergence of the system trajectory to the equilibrium point x^* . This set contains all the $x(k)$ states that can be controlled by a control action $u(k)$, $\forall x(k) \in \Omega, \exists u(k) \in \mathbb{U} : x(k+1) = Ax(k) + Bu(k) \in \lambda\Omega$. The λ -contractive controlled invariant set can be computed by using the methods proposed by Blanchini (1994) and Dórea and Hennes (1999), where an admissible set for which constraints are satisfied at time $x(k+1)$ is determined. Then, a geometric condition, described by Farkas' extended lemma, is determined (Castelan and Hennes 1993). Finally, the controlled invariance relations are used, and from the intersection between the two sets, a new set of constraints is obtained. These steps are performed until convergence to the largest controlled invariant set is achieved. This set is used in the design of constrained control because it is considered to be an important tool for stability, regardless of the technique used to compute the control action (Vassilaki et al. 1988; Blanchini 1999; Dórea and Hennes 1999; Blanchini and Miani 2008; Athanasopoulos and Bitsoris 2009d; Li and Liu 2016; Dantas et al. 2018b).

In this paper, the mp-LP solution (1) is defined within the λ -contractive controlled invariant set Ω . In this case, feed-

back system becomes nonlinear when the controller switches between the polyhedral partitions R_i , but the stability of the system is ensured since that control sequence computed from the solution of the optimization problem (7) is able to maintain trajectory of the states inside controlled invariant set with contraction at the same rate λ (Blanchini 1994).

Equation (5) allows the PWA-PI controller tuning. In this case, the system must be discretized using an appropriate sampling period. Then, given a λ -contractive controlled invariant set, the multiparametric programming problem (7) is solved to define the parameters F_j and g_j , from Eq. (1). The term F_j multiplies the state $x(k) = [x_1(k) \ x_2(k)]^T$, which correspond to the integral of the error and the error. Therefore, for this case, the PI controller parameters are defined by $F_j = [Ki_j \ Kp_j]$ and to the PI control term is added an affine term g_j given by the multiparametric linear program (mp-LP) solution. Consequently, a piecewise affine PI control law is designed.

3 PWA-PI Controller Implementation

In the previous sections, we presented a first-order system controlled by an affine PI controller. We then rewrote the closed-loop system in state space form to design a PWA-PI controller using the concept of controlled λ -contractive invariant sets together with the solution of the multiparametric linear programming problem. Despite presenting the theory to better understand PI controller, the main objective is to control industrial systems by synthesizing control actions from industrial controllers (PLC). Therefore, at first, we present the tuning algorithm implemented in The multiparametric toolbox (MPT) for MATLAB® (Herceg et al. 2013).

Algorithm 1 PWA-PI Multiparametric Controller Tuning

Input: Parameters of constraints and the model;

Output: F , g and Ω_c ;

- 1: Convert the continuous-time system (2) into a state space system according to Eq. (5);
- 2: Discretize the system in state space using the desired sampling period;
- 3: Compute the maximal λ -contractive control invariant set contained inside the set of constraints;
- 4: Solve the mp-LP in (7) to obtain the PWA control law.

The presented algorithm summarizes the steps needed to tune a PWA-PI controller. First, we convert the modeled real or identified system from continuous-time transfer function to continuous state space form presented in Eq. (5). Because the proposed approach is in discrete domain, the second step direct is to convert the continuous state space system to the discrete-time state space form using a proper sample time.

After that, we calculate the maximal λ -contractive controlled invariant set considering the real system's constrained set in order to guarantee for any set of error and integral of the error contained inside the invariant set there exist a control action, which belongs to a constrained set that forces the error to zero. Finally, we calculate the tuning parameters from the multiparametric program problem (7).

Note that because of the state feedback controller design, we assume zero reference considering that the states tend to zero. Although the controller design considers zero reference, we are designing a PI controller whose property is to force the error to zero because the integrative factor remains. Also, one can see that the integral of the error module increases as the reference module increases, which is necessary to force the error to zero. However, it approximates the value to the limits of the invariant set. Additionally, if the reference variation is too high, then the error can instantly approximate to the invariant set limits or even overpass. This means that even though the controlled invariant set grants constraints, because the reference difference to zero is acknowledged as a perturbation to the closed-loop system, it is still possible to violate the constraints. This happens when the reference variation is too high or the reference value makes the integral of the error to overpass the constraints.

After designing PWA-PI controller and obtaining its parameters, the next step is to implement control routines in PLC. Consequently, we created three integrated algorithm, the first of which is presented as follows:

Algorithm 2 Control Loop General Routine

Input: System's output (process variable—PV), mode (manual or automatic), setpoint (SP), manual control action (M_{AC});

Output: Control action (manipulated variable—MV);

```

1: if (mode == 1)
2:   Execute Algorithm 3: PWA-PI(SP, PV);
3: else
4:    $MV = M_{AC}$ ;

```

Algorithm 2 resume the most external and general PLC implemented function part. Basically, if the mode is equal to 1, then the algorithm begins the automatic mode, which consists in reading the PLC process variable (PV) and the operator inserted setpoint (SP). It then calls Algorithm 3 and passes the input parameters SP and PV. Any other mode happens if the mode is equal to another value. Then, the manual routine begins which consists in sending to manipulated variable (MV) the manual control Action (M_{AC}) value.

Algorithm 3 aims to calculate the control action, as follows:

In Algorithm 3, the PWA-PI control action is calculated. The function receives as input both the setpoint and process variable from the general routine described in Algorithm 2 and returns the control action calculated by the presented

Algorithm 3 PWA-PI Routine

Input: SP, PV;

Internal Variables: F (Further assigned to K_p and K_i), g (Further assigned to $Bias$);

Output: Control action (MV);

```

1: Compute the value of the integral of the error ( $e_i(k)$ );
2: Compute the value of the error ( $e(k)$ );
3: Assign the errors to the state vector:  $x(k) = [e_i(k) \ e(k)]^T$ ;
4: Execute Algorithm 4:  $j = \text{isInside}(x(k))$ ;
5: if ( $j \neq -1$ )
6:    $K_i = F_{j,1}$ 
7:    $K_p = F_{j,2}$ 
8:    $Bias = g_j$ 
9: end_if
10: Calculate MV:  $MV = (K_p)e(k) + (K_i)e_i(k) + Bias$ .

```

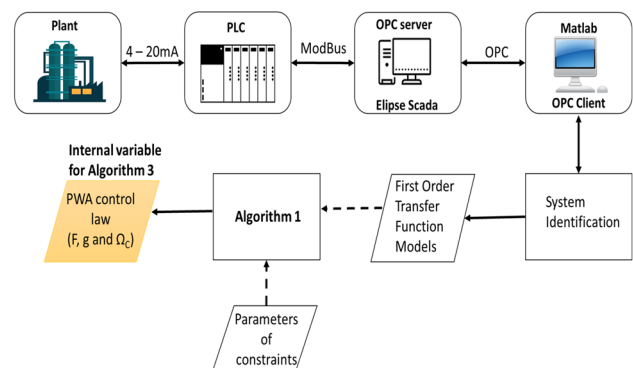


Fig. 2 Overview of the control system and the PWA control law computed by Algorithm 1

approach. Note that both F_i and g_i values, calculated in Algorithm 1, shall be manually inserted as **internal variable** parameters (Fig. 2). These parameters do not change as long as the application remains the same. The control action calculation routine is quite simple. At first, we calculate the error and integral of the error values. Then, both values are used to find the polyhedral region R_j to which the errors belong. As described in Sect. 2, the polyhedral region R_j is associated with F_j and g_j parameters which correspond to those needed to calculate the control action. Because of its importance, the algorithm to find the value of j , used to find both polyhedral region and associated parameters, is described as follows:

Algorithm 4 isInside Routine

Input: state vector ($x(k)$);

Internal Variables: Ω_c (λ -contractive controlled invariant set);

Output: isInside; Region in which the states are (-1 is none of them);

```

1: isInside := -1;
2: For  $j = 1:Regions\_Quantity$ 
3:   if state vector is contained inside the region;
4:     isInside := j;

```

Algorithm 4 presents the steps needed to find the active polyhedral region in state space and then find its associated

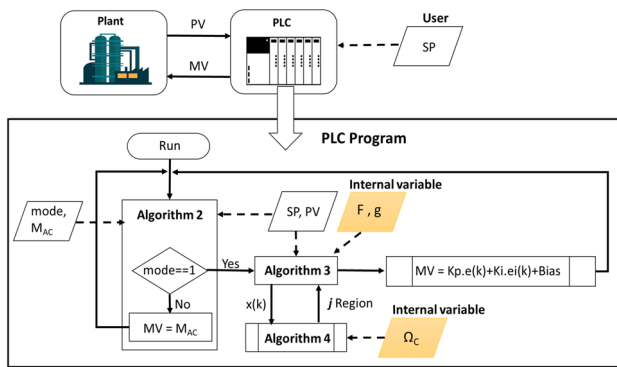


Fig. 3 Flowchart showing Algorithms 2, 3 and 4 inside PLC program

parameters. It is important to note that multiparametric programming technique returns piecewise affine parameters F_j and g_j of the PWA law associated with polyhedral regions from the invariant set Ω_c . In other words, it is related to the PWA control. This means that the index j is important to find the PWA control law ($u(k) = F_j x(k) + g_j$) used to calculate the control action, which is assured to constrain the states inside the invariant set and, consequently, inside the constraint set. Figure 3 depicts how Algorithms 2, 3 and 4 are combined in the PLC.

As shown in Fig. 3, the PLC collects the PV data and sends the MV values to the plant, where the MV was derived from the control law (line 10 of Algorithm 3). Also, the PLC executes a set of routines, namely Algorithms 2, 3 and 4, to compute MV value. Moreover, Algorithm 2 decides whether Algorithm 3 will be executed or not based on operator decision. If the operator decides set variable *mode* equal to one (Automatic mode), Algorithm 3 will be executed. Furthermore, Algorithm 3, when executed, receives SP and PV parameters from the plant, receives j from Algorithm 4, using j index to obtain F_j and g_j internal variable indexes. Note that Algorithm 4 is executed from Algorithm 3, receiving x at sample k from Algorithm 3 to find j index and returning it back to Algorithm 3. It is also important to note that all the developed algorithms presented in Fig. 3 are implemented in structured text language (Pinto 2018).

4 Applications of the PWA-PI Controller

To implement PWA-PI code in industrial PLCs, we use the Altus' Training Box Duo (Altus 2018). The Training Box Duo, illustrated in Fig. 4, is a complete didactic solution, with features that allow us to perform simulations of the elements present in industrial systems. This equipment is useful for commercial research laboratories and educational institutions because it can easily be linked to an industrial plant. It also communicates with several SCADA softwares via Mod-Bus RTU.



Fig. 4 Altus' Training Box Duo (Altus 2018)

The Training Box Duo TB 131 consists of a programmable controller (PLC) DU351 Series (PLC with integrated HMI) with a 32-bit processor and counts with a central unit that has graphical display 3.2" which allows visualization of programmed elements and the interaction with the process. The PLC has integrated I/O, digital inputs for simulation of field signals and the digital outputs to drive devices. In addition, analog input and output current and voltage are available, and at TB131 model, fast digital outputs of PWM and PTO type for DC motor drive and encoder's reading inputs allowing the realization of positioning axes. Finally, it has a dedicated circuit to perform control simulations and disturbance in PID loops and has the capacity to link up to 16 PLCs Duo network MODBUS RTU (Altus 2018).

The Training Box Duo uses MasterTool IEC programming software, which is free and offers five languages of programming (LD, FBD, ST, IL and SFC), described by IEC 61131-3 and an additional language (CFC).

The main features for this equipment are:

- Terminal blocks for digital inputs with simulation keys and status indication done by LEDs
- Terminal blocks for digital relay outputs with status indication done by LEDs
- Terminal blocks for analog input connections in voltage and current scale and potentiometer for simulation of voltage inputs
- Terminal block for analog output connection
- Digital voltage meter for analog output status indication
- Terminal block for fast input connection
- Stepper motor for usage with fast outputs
- Encoder for use with fast counter
- Connectors for RS-232 and RS-485 serial ports link

- Power plug for laptops
- Disturbance simulation for PID control study

The Training Box Duo, as other PLC, has important issues that should be highlighted. In this PLC, for example, some function blocks have update period different from the PLC duty cycle, which can influence the controller calculation. Moreover, numerical problems (integration for example) can affect the calculation of the control law-associated region. At last, there are other problems usually associated with hardware, such as I/O modules or field equipment, interference and memory limitation.

In this study, the Training Box Duo is used to operate two different systems using OPC communication to connect Elipse SCADA with MATLAB, where Elipse SCADA is the OPC server and MATLAB the client. MATLAB is used to analyze the data obtained from the plant, to identify their first-order function models and to perform Algorithm 1.

Note that all of the applications used the ideal PID equation, that is, the Altus PID block which includes anti-reset windup.

In the first application, we use the PWA-PI to control an available RLC network in the Training Box Duo, which simulates a temperature plant.

PWA-PI is also used in a level plant, as shown in Fig. 5. This plant has an 80-cm-high vessel with 100 cm diameter. The level data are obtained by a Smar LD301 pressure trans-

mitter. The pump has is 0.5 HP, and the frequency inverter is a WEG CFW-07.

Both RLC circuit (plant) and level plant use the same PLC.

To identify both systems, we assume that the models can be represented by first-order transfer functions. To estimate these first-order transfer functions, as represented by Eq. (2), we used the step response tests by the Smith (1985) method and step response by the Sundaresan and Krishnaswamy (1978) method, as well as the Åström and Hägglund (1995) method based on relay.

The methodology can be summarized as follows: From the step response, or relay feedback excitation, the input and output data of the plants are collected and the models are identified. The controllers are tuned in sequence. Finally, the closed-loop behavior with the PWA-PI is verified. The following examples aim to illustrate the controlled invariant concept to make a comparison with other tuning methods and verify the performance.

4.1 RLC Network

In the first example, we identified three first-order transfer function models to compare the tuned PWA-PI performance and its adherence to different models of the same control system. Models (8), (9) and (10) were obtained by the Smith, Sundaresan and Krishnaswamy and Åström and Hägglund methods.

$$M_1(s) = \frac{0.22}{s + 0.2222} \quad (8)$$

$$M_2(s) = \frac{0.2463}{s + 0.2487} \quad (9)$$

$$M_3(s) = \frac{0.285}{s + 0.2903} \quad (10)$$

For each identified first-order model, five tests were performed with different parameters, as shown in Table 1. The constraints on the error were defined in the -50°C and 50°C interval and the operating point at 50°C . These assumptions make the plant temperature vary in the interval from 0 to 100°C . Note the λ is the rate of contractivity of the invariant set, which may interfere in the settling time, and the constraint in the integral of the error is the parameter that influences the integrative gain of the PWA-PI control law. We assume that the integral of the error can affect the performance of the plant, such as settling time and overshoot, besides eliminating the same steady-state errors for common PI controllers.

Table 2 shows the performance evaluation results using both indexes the mean square error (MSE) and Goodhart et al. (1994). Note that the upper output limit was set at 100°C and the upper limit of the control action was 100%.



Fig. 5 Level plant used in the PWA-PI tests

Table 1 Test conditions using identified models

Model	Tests	λ	Constraints of the error	Constraint in the integral of the error
$M_1(s)$	Test 1	0.9	[50 -50]	[150 -150]
	Test 2	0.9	[50–50]	[200–200]
	Test 3	0.9	[50–50]	[300–300]
	Test 4	0.8	[50–50]	[150–150]
	Test 5	0.77	[50–50]	[250–250]
$M_2(s)$	Test 1	0.9	[50–50]	[150–150]
	Test 2	0.9	[50–50]	[200–200]
	Test 3	0.9	[50–50]	[300–300]
	Test 4	0.8	[50–50]	[150–150]
	Test 5	0.77	[50–50]	[250–250]
$M_3(s)$	Test 1	0.9	[50–50]	[150–150]
	Test 2	0.9	[50–50]	[200–200]
	Test 3	0.9	[50–50]	[300–300]
	Test 4	0.8	[50–50]	[150–150]
	Test 5	0.77	[50–50]	[250–250]

Table 2 Performance evaluation analysis of the PWA-PI with different constraint parameters

Model	Tests	MSE	Goodhart
$M_1(s)$	Test 1	360.78	351.35
	Test 2	270.59	273.96
	Test 3	259.09	252.43
	Test 4	355.58	359.86
	Test 5	307.05	310.81
$M_2(s)$	Test 1	347.95	352.17
	Test 2	283.43	287.81
	Test 3	234.73	228.36
	Test 4	295.23	298.68
	Test 5	300.64	304.90
$M_3(s)$	Test 1	290.30	284.22
	Test 2	274.88	269.15
	Test 3	233.59	227.24
	Test 4	282.92	275.15
	Test 5	235.88	230.77

The Goodhart is a control-loop performance assessment index which uses a combination of $p1$ (Eq. 11), $p2$ (Eq. 12) and $p3$ (Eq. 13), as follows:

1. The control action average performed to achieve a given response, as

$$p1 = \frac{\sum |u(k)|}{N}, \quad (11)$$

where N is the number of samples and $u(k)$ is the control signal.

2. The control action variance around the average, given by

$$p2 = \frac{\sum [u(k) - p1]^2}{N}, \quad (12)$$

3. The total deviation of the output from the reference value, given by

$$p3 = \frac{\sum [r(k) - y(k)]^2}{N}, \quad (13)$$

where $r(k)$ is the reference value and $y(k)$ is the output process.

The Goodhart index is an affine combination of $p1$, $p2$ and $p3$

$$\text{Goodhart} = \alpha_1 p1 + \alpha_2 p2 + \alpha_3 p3, \quad (14)$$

where α_1 , α_2 , α_3 are assigned weights whose sum must be one.

Test 3's parameters resulted in the best performance for the controller using the model $M_3(s)$. Therefore, we will use these parameters to show the PWA-PI control results in the remainder of this section. Additionally, we conclude that a numerical increase in the integral of the error parameters improved the plant performance in this case.

In brief, we defined the operating point at 50 °C. The process output (process variable—PV) constraint parameters were: 0° for lower limit and 100 °C for upper limit. The constraint parameters used for the control action (manipulated variable—MV) were: 0% for lower limit and 100% for upper limit. The constraint of the error parameters used in the three models were: – 50 and 50 for the error, and the integral of the error at – 300 and 300. In addition, we defined λ as 0.9. From these parameters, we obtained a PWA control law with 20 polyhedral regions for model (10), according to Fig. 6.

The PWA-PI parameters were obtained for models (8), (9) and (10). The algorithm returned F_j , g_j and Ω_c for all these models. The next step is to configure these data in the PLC and test the PWA-PI loop algorithm. The test considered initial condition equal to zero, which is equivalent to 50 °C from the operation point as shown in Fig. 7.

Figure 7 shows the plant output using the PWA control law with 20 polyhedral regions. Note that the control action is saturated, which indicates that this constraint was respected. The overshoot was 58%, and the settling time (5% criterion) was 15 s. In Fig. 8, the plant setpoint was changed from 50 to 90 °C, being the setpoint close to the maximum output constraint. Note in this situation, the process variable and control action constraints are also respected and in this region the overshoot is 5.5%.

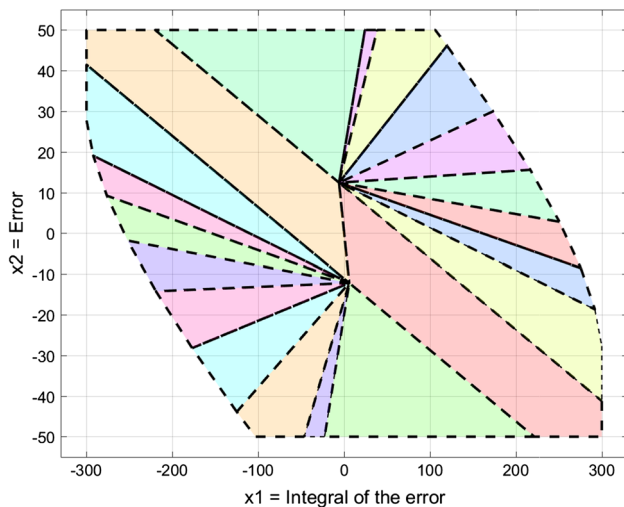


Fig. 6 Invariant set with 20 polyhedral regions (model $M_3(s)$)

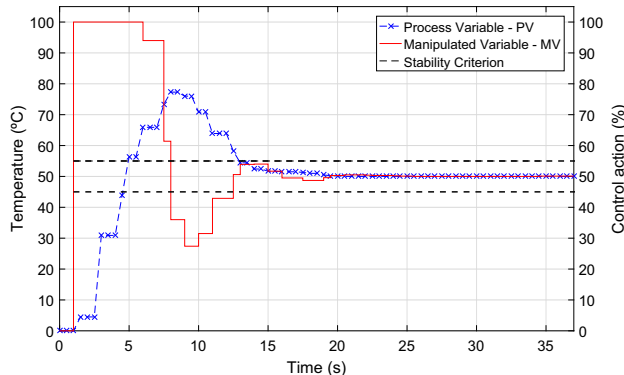


Fig. 7 Controlled plant output by PWA-PI

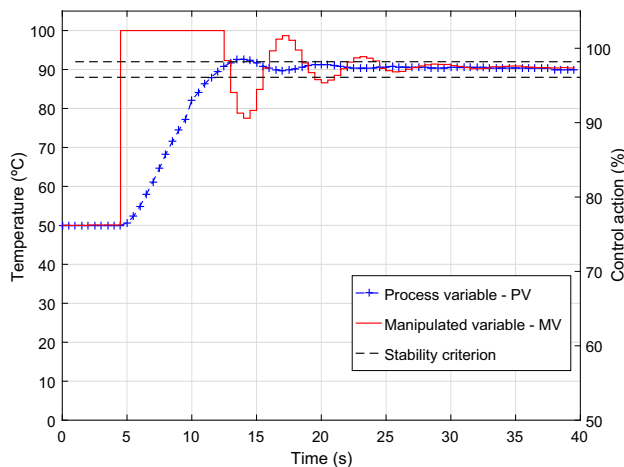


Fig. 8 Plant output close to the maximum output constraint. Overshoot 5.5%

Figure 9 shows the state evolution from inside the invariant set. From the initial condition to the operating point of 50 °C. PWA-PI respected the previously defined constraints

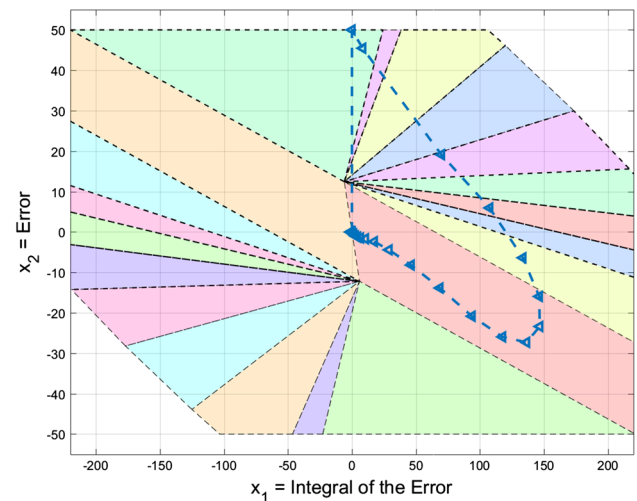


Fig. 9 State trajectory inside controlled invariant set

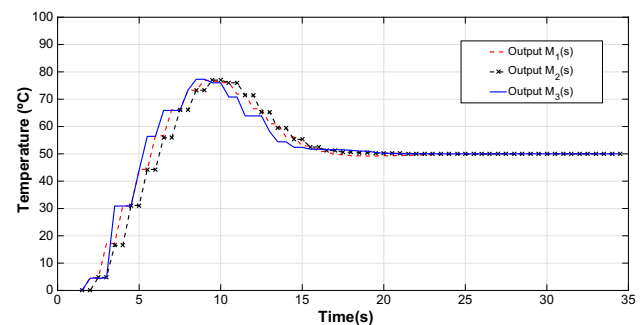


Fig. 10 Models $M_1(s)$, $M_2(s)$ and $M_3(s)$ output controlled by plant

in output and control action, keeping the error within the invariant set.

Figure 10 shows the plant output obtained using the parameters of Test 3, but generated from models $M_1(s)$, $M_2(s)$ and $M_3(s)$.

Small differences between the outputs due to different models are shown in Fig. 10. However, PWA-PI remains with similar performance.

To verify the efficiency of the PWA-PI, we compared the closed-loop control with both PWA-PI and PI with Ziegler–Nichols tuning method (Ziegler and Nichols 1943). The tuning PI was 1.3 for proportional gain and 8.0 for integrative time was found for this plant using a PI controller with Z–N method. Figure 11 shows a comparison of both controllers.

Generally, the Z–N method results in a more aggressive response when systems are well represented by a first-order transfer function. However, in this case, the Z–N method presented non-aggressive response as shown in Fig. 11, because of the estimation of the identified system transport delay (Åström and Hägglund 1995).

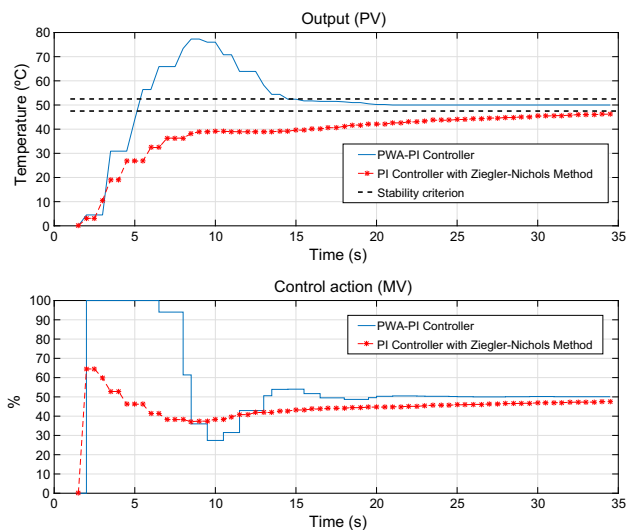


Fig. 11 Comparison of the plant output with PWA-PI controller and PI controller with Z–N method

Table 3 Comparison of the performance evaluation of the level plant controlled by PWA-PI and PI with Z–N method

Controller	MSE	Goodhart	$T_{s5\%}$ (s)
PWA-PI	233.60	227.23	15
PI with Z–N method	240.87	244.85	35

Considering the 5% settling time criterion ($T_{s5\%}$), the output controlled by PWA-PI stabilized in 15 s, while the PI controller with Z–N rules stabilized in 35 s.

Table 3 uses the previously defined MSE, Goodhart indexes and $T_{s5\%}$, and analyzes the performance of both controllers.

According to the results, the PWA-PI controller presented a higher overshoot. However, the overshoot can be lower by changing the constraints on the error and on the integral of the error, showing the proposed controller is flexible to attend several scenarios. A study of this flexibility is shown in Sect. 4.2.

4.2 Industrial Plant Case

The industrial plant presented in Fig. 5 was used to test the proposed controller. The pressure in the vessel was maintained constant, ensuring only PWA-PI control actions in the level control loop. Both the industrial plant vessel level and control action are expressed as a percentage. The chosen output operating point was 40%.

For the industrial plant level control loop, the model presented in Eq. 15 was identified using the Åström and Hägglund relay method.

$$M_4(s) = \frac{1.0008}{s}. \quad (15)$$

The PWA-PI controller was evaluated by performing three tests in this industrial plant changing the setpoint by 20% from the 40% operating point. In the first two tests, the PI-PWA was used with parameters according to Table 4. The third test used the PI controller with Z–N method, similar to the one used in the previous subsection. Table 4 shows the parameters used to generate the PWA control law from the defined constraints and the model. The maximum level value and control action maximum limit were set at 100%.

Figures 12 and 13 show Tests 1 and 2 polyhedral regions, respectively. Both presented PWA control laws with seven polyhedral regions. The main difference is in the integral of the error parameters. In the tests of Sect. 4.1, the variation constraints on the integral of the error directly affected the industrial plant performance and indexes such as settling time and overshoot. Therefore, a larger numerical difference in bounds of the integral of the error between the two tests was used.

Figure 14 shows the comparison of the outputs controlled by PWA-PI and by a PI controller tuned with Z–N method, using Tests 1 and 2. For the PI controller, the proportional gain was 0.7 and the integrative time was 16.8.

According to Table 5, PWA-PI presented the best results. In Fig. 15, the state evolution from the invariant set perspective with the parameters defined in Test 2 is only presented for the PWA-PI controller, where the PWA nature of the control laws is evident.

Table 4 Parameters used in two tests to obtain PWA control law

Tests	λ	Constraint of the error	Constraint integral of the error
Test 1	0.9	[40–60]	[500–500]
Test 2	0.9	[40–60]	[1500–1500]

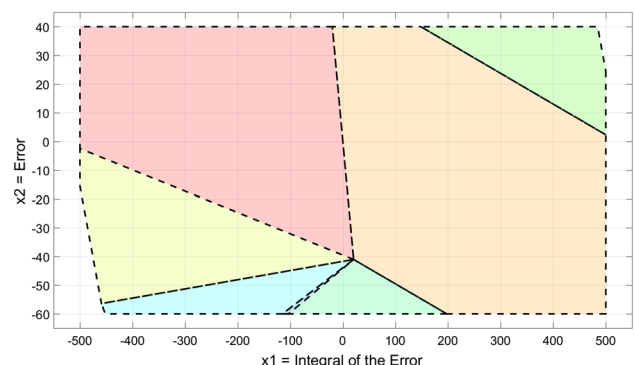


Fig. 12 Invariant set with seven polyhedral regions with Test 1 parameters

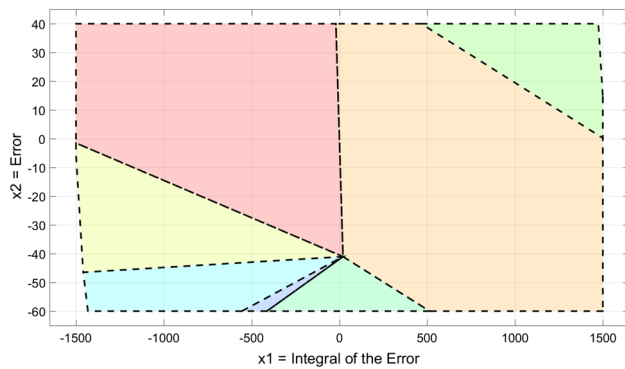


Fig. 13 Invariant set with seven polyhedral regions with Test 2 parameters

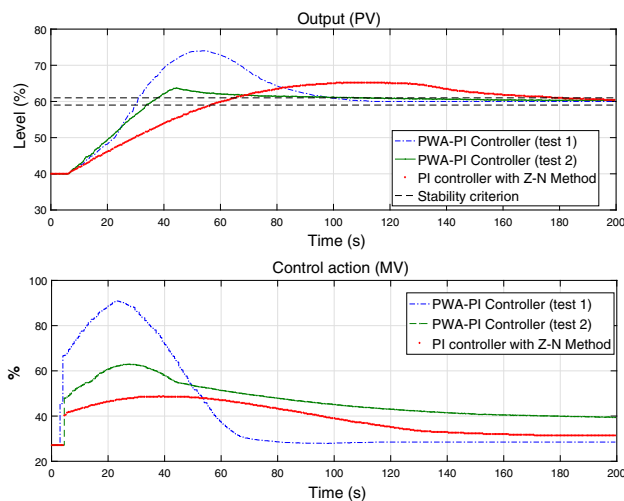


Fig. 14 Comparison of the plant output by PWA-PI controller and PI controller with Z–N method

Table 5 Comparison performance evaluation between PWA-PI controller and PI controller with Z–N method

Controller	MSE	Goodhart	$T_{S5\%}$ (s)
PWA-PI (Test 1)	233.61	224.99	98
PWA-PI (Test 2)	109.58	110.17	102
PI with Z–N method	131.89	132.47	179

Considering the 5% settling time criterion ($T_{S5\%}$), the industrial plant output over PWA-PI controller with the parameters defined in Test 1 presented a 98s settling time and with the parameters defined in Test 2 it presented a 102s settling time, whereas the PI controller with Z–N method presented a 179s settling time.

Table 5 shows the performance evaluation of analysis both controllers using the MSE, Goodhart indexes and $T_{S5\%}$.

In the performed tests, the PWA-PI controller satisfied the defined control constraints without the need for auxiliary techniques, such as anti-reset windup and bumpless. Addi-

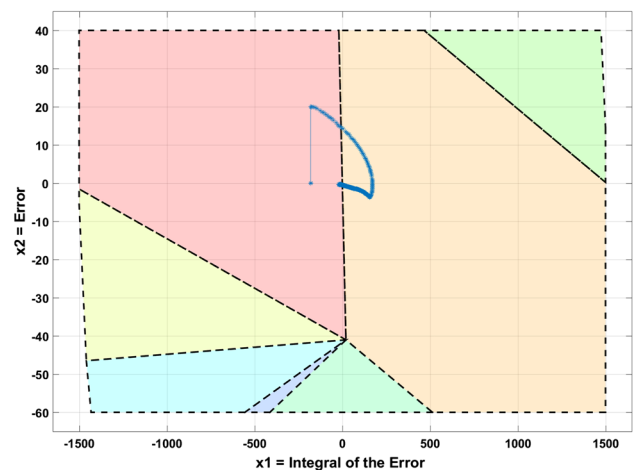


Fig. 15 Trajectory contained inside controlled invariant set for Test 2

tionally, PWA-PI control laws with better results are achieved by changing up the integral of the error constraint.

5 Conclusions

In this paper, a PWA-PI controller algorithm based on invariant set and multiparametric programming for constrained systems was implemented in an industrial PLC and submitted to several tests. These tests compared the PWA-PI with PI controller tuned by Z–N method and it presented better results in terms of settling time, MSE and Goodhart indexes. Despite presenting higher overshoot in the first test results, the analyzed tuning method (mp-LP) was adjusted to diminish it by changing the constraint set. In addition, the outputs were contained inside the constraint set in all cases. Additionally, the PWA-PI presented 98s settling time, whereas the PI controller with Z–N method presented 179s settling time. In conclusion, we believe that the PWA-PI controller is a reliable solution to control constrained systems because the parameters are obtained by an optimization problem that can be adjusted to best performance indexes. Additionally, the implemented method is easy to code in ST programming language without needing additional hardware.

In the test cases, the PWA-PI controller presented lower overshoot when the reference is defined near the constraint. This behavior should be better investigated in future works.

Acknowledgements The authors would like to thank the UFRN (Federal University of Rio Grande do Norte), UnP (Potiguar University), LAUT (Laboratory in Oil Automation), PPgEEc-UFRN (Postgraduate Program in Electrical and Computation Engineering of UFRN) and CAPES (Coordination for the Improvement of Higher Education Personnel).

References

- Allad, M., & Acheli, D. (2018). Liquid level cascade control using various combinations of controllers. *Reveu Roumaine des Sciences Techniques-serie Electrotechnique et Energetique*, 63(1), 94–99.
- Altus. (2018). Training box duo. <http://www.altus.com.br/produto/15/training-box-duo>. Accessed 18 July 2018
- Åström, K. J., & Hägglund, T. (1995). *PID controllers: Theory, design, and tuning* (Vol. 2). Research Triangle Park, NC: Instrument society of America.
- Athanasopoulos, N., & Bitsoris, G. (2009). A novel approach to the computation of the maximal controlled invariant set for constrained linear systems. In *2009 European control conference (ECC)* (pp 3124–3129). <https://doi.org/10.23919/ECC.2009.7074885>
- Bemporad, A., Morari, M., Dua, V., & Pistikopoulos, E. N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.
- Blanchini, F. (1994). Ultimate boundedness control for uncertain discrete-time systems via set-induced Lyapunov functions. *IEEE Transactions on Automatic Control*, 39(2), 428–433.
- Blanchini, F. (1999). Set invariance in control. *Revista Automática*, 35, 1747–1767.
- Blanchini, F., & Miani, S. (2008). *Set-theoretic methods in control* (1st ed.). Boston: Birkhauser.
- Castelan, E. B., & Hennes, J. C. (1993). On invariant polyhedra of continuous-time linear systems. *IEEE Transactions on Automatic Control*, 38, 1680–1685. <https://doi.org/10.1109/9.262058>.
- Cohen, G. H., & Coon, G. A. (1953). Theoretical considerations of retarded control. *Transactions ASME*, 75, 827–837.
- Dantas, A. D. O. S., Dantas, A. F. O. A., Campos, J. T. L. S., de Almeida Neto, D. L., & Dórea, C. E. T. (2018a). PID control for electric vehicles subject to control and speed signal constraints. *Journal of Control Science and Engineering*, 2018, 6259049. <https://doi.org/10.1155/2018/6259049>.
- Dantas, A. D. O. S., Dantas, A. F. O. A., & Dórea, C. E. T. (2018b). Static output feedback control design for constrained linear discrete-time systems using data cluster analysis. In *IET control theory and applications*. <https://doi.org/10.1049/iet-cta.2018.5744>.
- Dórea, C. E. T., & Hennes, J. C. (1999). (a, b)-invariant polyhedral sets of linear discrete-time systems. *Journal of Optimization Theory and Applications*, 103(3), 521–542.
- Gersnoviez, A., Brox, M., & Baturone, I. (2017). High-speed and low-cost implementation of explicit model predictive controllers. *IEEE Transactions on Control Systems Technology*, 27, 647–662.
- Goodhart, S. G., Burnham, K. J., & James, D. J. G. (1994). Bilinear self-tuning control of a high temperature heat treatment plant. *IEE Proceedings Control Theory and Applications*, 141(1), 12–18. <https://doi.org/10.1049/ip-cta:19949748>.
- Herceg, M., Kvasnica, M., Jones, C. N., & Morari, M. (2013). Multi-parametric toolbox 3.0. In *Control conference (ECC), 2013 European* (pp 502–510). IEEE
- Homod, R. Z. (2018). Analysis and optimization of HVAC control systems based on energy and performance considerations for smart buildings. *Renewable Energy*, 126, 49–64.
- Huyett, L., Dassau, E., Zisser, H., & Doyle, F. (2015). Design and evaluation of a robust PID controller for a fully implantable artificial pancreas. *Industrial and Engineering Chemistry Research*, 54(42), 10311–10321.
- Jiao, Z. Q., Qu, B. D., & Xu, B. G. (2008). Decoupling design for multivariate PID controller based on RBF neural network. *Xitong Fangzhen Xuebao/Journal of System Simulation*, 20(3), 627–634.
- Khan, M., Choudhry, M., Zeeshan, M., & Ali, A. (2015). Adaptive fuzzy multivariable controller design based on genetic algorithm for an air handling unit. *Energy*, 81, 477–488.
- Li, Y., & Liu, J. (2016). Computing maximal invariant sets for switched nonlinear systems. In *2016 IEEE conference on computer aided control system design (CACSD)* (pp 862–867). <https://doi.org/10.1109/CACSD.2016.7602547>
- Monteiro, J. R., Pinheiro, C. A. M., & Lopes, B. I. L. (2017). Implementation of a rough controller with proportional-integral action to control a nonlinear system. *Journal of Control, Automation and Electrical Systems*, 28(3), 337–348. <https://doi.org/10.1007/s40313-017-0302-6>.
- Nie, X., & Zheng, W. X. (2015). Complete stability of neural networks with nonmonotonic piecewise linear activation functions. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 62(10), 1002–1006. <https://doi.org/10.1109/TCSII.2015.2436131>.
- O'Dwyer, A. (2006). *PI and PID controller tuning rules*. Covent Garden: Imperial College Press.
- Pinto. (2018). ST code implementation in PLC. <https://goo.gl/Wm64uR>. Accessed 11 December 2018
- Puangdownreong, D. (2018). Optimal PID controller design for DC motor speed control system with tracking and regulating constrained optimization via cuckoo search. *Journal of Electrical Engineering and Technology*, 13(1), 460–467.
- Rai, P. K., Dhanya, C. T., & Chahar, B. R. (2017). Flood control in an urban drainage system using a linear controller. *Water Practice and Technology*, 12(4), 942–952. <https://doi.org/10.2166/wpt.2017.102>.
- Schulze Darup, M., & Mönnigmann, M. (2018). Optimization-free robust MPC around the terminal region. *Automatica*, 95, 229–235. <https://doi.org/10.1016/j.automatica.2018.05.025>.
- Sendoya-Losada, D. F., & Quintero-Polanco, J. D. (2018). PID controller applied to an unmanned aerial vehicle. *ARN Journal of Engineering and Applied Sciences*, 12(1), 325–334.
- Short, M., & Abugchem, F. (2017). A microcontroller-based adaptive model predictive control platform for process control applications. *Electronics*, 6(4), 88. <https://doi.org/10.3390/electronics6040088>.
- Shorten, R., Wirth, F., Mason, O., Wulff, K., & King, C. (2007). Stability criteria for switched and hybrid systems. *SIAM Review*, 49(4), 545–592.
- Smith, G. D. (1985). *Numerical solution of partial differential equations: Finite difference methods*. Oxford: Oxford University Press.
- Sundaresan, K., & Krishnaswamy, P. (1978). Estimation of time delay time constant parameters in time, frequency, and laplace domains. *The Canadian Journal of Chemical Engineering*, 56(2), 257–262.
- Tchamna, R., & Lee, M. (2018). Analytical design of an industrial two-term controller for optimal regulatory control of open-loop unstable processes under operational constraints. *ISA Transactions*, 72, 66–76.
- Vassilaki, M., Hennes, J. C., & Bitsoris, G. (1988). Feedback control of linear discrete-time systems under state and control constraints. *International Journal of control*, 47(6), 1727–1735.
- Weber, F. (2015). Robust force tracking control scheme for mr dampers. *Structural Control and Health Monitoring*, 22(12), 1373–1395. <https://doi.org/10.1002/stc.1750>.
- Wei, Y., Park, J. H., Qiu, J., & Jung, H. (2018). Reliable output feedback control for piecewise affine systems with Markov-type sensor failure. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 65(7), 913–917. <https://doi.org/10.1109/TCSII.2017.2725981>.
- Weinkeller, G. P., Salles, J. L., & Teodiano Filho, F. (2012). Predictive control via multi-parametric programming applied to the dynamic model of a robotic wheelchair. In *Robotics symposium and Latin American robotics symposium (SBR-LARS), 2012 Brazilian* (pp 179–184). IEEE.

- Xiu, X., & Zhang, J. (2018). Grid Kd tree approach for point location in polyhedral data sets-application to explicit MPC. *International Journal of Control*, 1, 1–14.
- Zelentsov, D., & Denysiuk, O. (2019). Neural network algorithm for accuracy control in modelling of structures with changing characteristics. In Z. Hu, S. Petoukhov, I. Dychka, & M. He (Eds.), *Advances in Computer Science for Engineering and Education* (pp. 507–516). Cham: Springer.
- Zhang, J., Xiu, X., Xie, Z., & Hu, B. (2016). Using a two-level structure to manage the point location problem in explicit model predictive control. *Asian Journal of Control*, 18(3), 1075–1086.
- Ziegler, J., & Nichols, N. (1942). Optimum settings for automatic controllers. *Transactions ASME*, 64(1), 759–768 .
- Ziegler, J., & Nichols, N. (1943). Process lags in automatic control circuits. *Transactions ASME*, 65(5), 433–443.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.