A linear algebra controller based on reduced order models applied to trajectory tracking for mobile robots: an experimental validation

Leonardo Guevara, Oscar Camacho*,
Andrés Rosales and Javier Guevara

Departamento de Automatización y Control Industrial,
Escuela Politécnica Nacional,
Quito, 170517, Ecuador
Email: cesar.guevara01@epn.edu.ec
Email: oscar.camacho@epn.edu.ec
Email: andres.rosales@epn.edu.ec
Email: dario.guevara@epn.edu.ec
*Corresponding author

Gustavo Scaglia

Facultad de Ingeniería,
Instituto de Ingeniería Química-CONICET,
Universidad Nacional de San Juan,
Av. San Martin (Oeste) 1109,
San Juan, Argentina
Email: gscaglia@unsj.edu.ar

Abstract: A linear algebra controller (LACr) based on an empirical linear model of the system is presented in this paper. The controller design is based on a first order plus dead time (FOPDT) model and can be tuned using the characteristic parameters obtained from the reaction curve. In previous studies, the versatility of this proposed controller was tested by simulations, proving be an alternative to control many kinds of processes. In this paper, the proposed controller is implemented for trajectory tracking using a real mobile robot platform. The performance results are compared against a PI controller using the ISE performance index to measure it.

Keywords: numerical method controller; linear algebra; reduced order models; characteristic parameters; pioneer 3-DX; mobile robot; trajectory tracking.

Reference to this paper should be made as follows: Guevara, L., Camacho, O., Rosales, A., Guevara, J. and Scaglia, G. (2019) 'A linear algebra controller based on reduced order models applied to trajectory tracking for mobile robots: an experimental validation', *Int. J. Automation and Control*, Vol. 13, No. 2, pp.176–196.

Biographical notes: Leonardo Guevara received his degree in Electronic and Control Engineering from the Escuela Politécnica Nacional, Quito, Ecuador in 2016, where he was also Research Assistant and member of the Editorial Support Committee of Revista Politécnica Magazine. He is currently a PhD student in Electronic Engineering at the Universidad Técnica Federico Santa María, Valparaíso, Chile. His research interests include robotics, control and modelling.

Oscar Camacho received his BSc in EE from the Universidad de Los Andes (ULA), Venezuela in 1984, MSc in Control Engineering from the ULA in 1992 and ME and PhD in Chemical Engineering from the University of South Florida, Tampa, Fl. in 1994 and 1996, respectively. He developed post-doctoral activities at USF in 2001. He has held teaching and researching in the ULA, USF and Escuela Politécnica Nacional (EPN) in Ecuador. His current research interests include sliding mode control, long delay systems, and chemical process control. He has authored more than 150 publications in journal and conference proceedings.

Andrés Rosales is a full time Professor at the Escuela Politécnica Nacional (EPN), Quito. He received his Bachelor in Electronics and Control Engineering (2001) from the EPN. He graduated (2009) as a PhD in Control Systems Engineering at the Universidad Nacional de San Juan, Argentina. He made a research residency at the University of Hannover, Germany, as a DAAD scholarship holder. Actually, he is the Research Head at the EPN. His research interests include control systems, robotics, industrial process, and education.

Javier Guevara is a PhD student at the Universidad Tecnica Federico Santa Maria since 2017. He received his degree in Control Electronics Engineering in the Escuela Politecnica Nacional in 2016. He had worked at the Escuela Politecnica Nacional since May of 2015 until October 2016 as a Research Assistant. His main interests are: robots in agriculture, autonomous cars, computer vision, and deep learning.

Gustavo Scaglia is a Professor at the Universidad Nacional de San Juan, San Juan, Argentina. He received his MSc in Control Systems in 2003 and in 2006, he graduated as a PhD in Control Systems at the Universidad Nacional de San Juan. Currently, he develops investigation and education projects at the Institute of Chemistry Engineering at Universidad Nacional de San Juan and National Council for Scientific and Technical Research (CONICET), Argentina.

1 Introduction

Scaglia et al. (2009) proposed an easy and simple technique that allows controlling a diversity of nonlinear systems based on linear algebra and numerical methods concepts to design control algorithms. The methodology consists of searching the conditions under which a system of linear equations has an exact solution and establishing the desired control values for tracking the error to zero.

Several works use the above mentioned methodology to design control algorithms for diverse applications such as chemical processes (Quintero et al., 2009; Suvire et al., 2013; Scaglia et al., 2014; Rómoli et al., 2015) or Robotics (Scaglia et al., 2009, 2015; Rosales et al., 2010; Serrano et al., 2015; Capito et al., 2016.) For all of these previous applications, controllers were designed based on the complete model of the process. Besides, there are two problems with the use of a model as far as industrial processes are concerned. First, the development of a complete model is difficult due to the complexity of the process and to the lack of knowledge about process parameters. Second, most process models relating the controlled and the manipulated variables are of a higher-order. Therefore, the traditional numerical method procedure can produce a more complex controller. For each process, a different control law is synthesised, resulting in a designed controller for the process under study.

An efficient alternative modelling method for a process control is the use of an empirical model with low order linear models with dead time (Smith and Corripio, 1997). Usually, first-order-plus dead time (FOPDT) models are adequate for process control analysis and design. In many cases, mobile robotics as well as chemical processes, can be represented by FOPDT models (Capito et al., 2016).

With this background in mind, Guevara et al. (2016a) proposed the design of a controller based on systems with an open loop behaviour such as a FOPDT model. The resulting controller presented a general and fixed structure with tuning parameters as a function of the characteristic parameters of the process. Therefore, that controller is easy to implement in a programmable platform and can be applied to different kind of processes if they have similar comportment like a FOPDT model. The proposed controller has been tested for higher order linear systems and then on nonlinear chemical processes (Guevara et al., 2016b) obtaining good performance results. All the tests were performed on the controller using simulations.

The aim of this work is to test the proposed controller on a real process. This work shows the application of the suggested controller for tracking the trajectory of a mobile robot. The objective is to find the control actions that order the mobile robot to reach the Cartesian position (x, y) with an adequate orientation for each sampling period. In previous works, several control strategies have been proposed for trajectory tracking, designed considering the kinematic model (Hedjar et al., 2005; Kühne et al., 2005) or dynamic robot models (Brennan and Alleyine, 2002; Shuli, 2005). In this work, an approximated model of the robot is used and the controller tests are performed on different types of trajectories maintaining a set point of linear speed. The results are evaluated using a performance index and compared to the results of a PID controller.

This paper is organised as follows: Section 2 briefly presents the basic concepts of controller design and its application. Section 3 shows the methodology for designing controllers using Linear Algebra and developing the proposed controller. In Section 4, the simulation results compare the proposed controller to the conventional design. Then, the real time experiments are presented where the proposed controller is compared to a PID controller. Finally, in Section 5 conclusions and future works are discussed.

2 Background

This section provides an overview of basic conceptual issues needed for the development of the proposed controller and its application within the field of mobile robotics.

2.1 Unicycle robot

Unicycle is a mobile robot whose structure consists of two fixed wheels on the same axis, controlled independently and to provide greater stability, an additional wheel with no controls. The traction-steering system works by the difference in speed between the fixed wheels, allowing independent linear and angular velocity control.

The advantages of the unicycle type mobile robot is high mobility, high traction and simplicity in its configuration. It also implies restriction when moving in lateral directions. Like a car, it is only allowed to move forward or backward.

The Pioneer 3DX (Figure 1) is a unicycle mobile robot created by adept mobile robots, the reference platform for robotic research thanks to its versatility, reliability, and durability.

Figure 1 Pioneer 3DX (see online version for colours)

The Pioneer 3DX has an internal microcontroller that manages low-level actions, including acquiring sensor data such as sonars, encoders or additional accessories such as clamps. Its purpose is to maintain the desired course and speed of the platform using a low level PID that controls PWM signals sent to the motors.

To carry out more advanced and high-level control actions requires the connection of a PC in a client-server using serial communication. Actions such as: avoidance of obstacles, route planning, pattern recognition, location-navigation and more are possible.

2.2 Numerical methods

A numerical method is a procedure by which an approximate solution to a problem is obtained by performing elementary arithmetic operations or purely logical calculations. In general, these methods apply when a numerical value is needed as a solution to a mathematical problem, and analytical procedures (algebraic manipulations, differential equation theory or integration methods) are very complex or incapable of giving a response. Due to this, numerical methods are procedures frequently used by physicists and engineers, whose development has been favoured by the need to obtain approximate solutions, although the precision is not complete.

The efficiency or loss of information during the calculation of the approximation often depends on the ease of implementation of the algorithm and the special characteristics and limitations of the calculation instruments (computers).

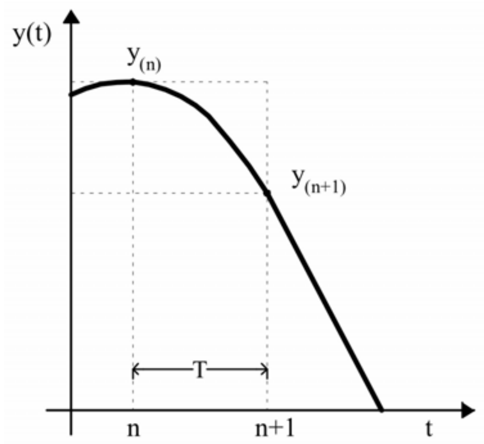
2.2.1 Euler approximation

This approximation (Figure 2) is used for discretising the derivative term, replacing it by a finite difference. The required precision by the numerical method used here is smaller than the one required simulating a system. This is so because, when state feedback is used at each sampling time, any shift between the approximation and the real system can then be corrected so that no accumulative errors exist. The approximation is only used to find the best manner to go from the current state to the following one, and not to duplicate the entire system evolution. Euler's method allows achieve very good simulated and practical results, as can be seen on the following works as well (Rómoli et al., 2017; Serrano et al., 2017; Pantano et al., 2017). Therefore, the Euler approximation method can be chosen instead of other methods and it can be concluded that Euler's method election is a proper choice.

Hence, if the slope value is known in a period, the value of the next state could be estimated using a linear approximation (Hildebrand, 1987). The discrete approximation of a derivative is represented by:

$$\dot{y}_{(n)} \approx \frac{y_{(n+1)} - y_{(n)}}{T} \tag{1}$$

Figure 2 Euler approximation



2.3 *First order plus dead time model*

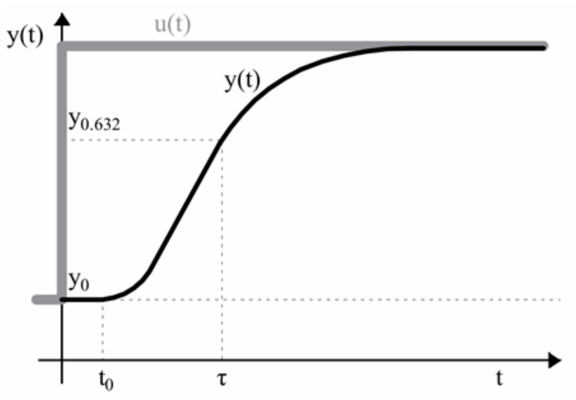
In industrial processes, the reaction curve is an often-used method for identifying dynamic models (Camacho and Smith, 2000). It is simple to use and provides adequate models for many applications. The curve is obtained by introducing a step change in the controller output and recording the transmitter output. The curve allows to obtain the characteristics model parameters. The FOPDT model approximates the actual higher-order process. Therefore, a first order plus time delay model (2), can adequately represent industrial process dynamics over a range of frequencies (Smith and Corripio, 1997). Figure 3 shows the reaction curve procedure and the response is called a FOPDT.

The transfer function of a FOPDT system is represented by:

$$G(s) = \frac{y(s)}{u(s)} = \frac{ke^{-t_0 s}}{\tau s + 1} \tag{2}$$

where k is the gain, τ is the time constant and t_0 is the dead time or delay.

Figure 3 Typical response of a FOPDT system



3 Controller design

This section is divided in two parts. The first part presents the general methodology for designing the LACr. The second part presents the proposed self-regulating processes, using a FOPDT process model.

3.1 General methodology

The controller proposed in this work is based on linear algebra and numerical methods (Scaglia et al., 2009). Consider the following first-order differential equation:

$$\frac{dy}{dt} = y = f(y, t, u) \quad y(0) = y_0 \tag{3}$$

where y represents the output of the system to be controlled, u is the control action, and t is the time. The values of $y(t)$ at discrete time $t = nT$ will be denoted as $y_{(n)}$, where T is the sampling period and $n \in \{0, 1, 2, \ldots\}$. Thus, when computing $y_{(n+1)}$ by knowing $y_{(n)}$, (3) should be integrated over the time interval $nT \le t \le (n + 1)T$ as follows:

$$y_{(n+1)} = y_{(n)} + \int_{nT}^{(n+1)T} f(y, t, u)dt \tag{4}$$

where u remains constant during the time interval. Therefore, if the reference trajectory $y_{ref(t)}$ to be followed by $y(t)$ is known beforehand, then $y_{(n+1)}$ can be substituted by $y_{ref(n+1)}$ into (4). Then, it is possible to calculate $u(n)$ that represents the control action required to go from the current state to the desired one.

The Euler approximation is a numerical integration method that can be used to calculate the integral in (4). The use of this kind of numerical method is based on the possibility of determining the state of the system at the next instant from state, the control action, and other variables. Then, can be substituted by a function of reference trajectory and the control action to make the output system evolve from the current value to the desired one can be calculated. To accomplish this, a system of linear equations for each sampling period must be solved.

3.2 Proposed approach

This section presents the development of a general linear algebra controller (LACr), for self-regulating processes, using a FOPDT process model. The FOPDT model approximates the actual mobile robot. The development of this controller significantly simplifies the application of linear algebra control theory to mobile robots. Thus, to synthesise the controller two hypotheses are considered.

3.2.1 First hypothesis

The model of the mobile robot is given by:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v_c \cos(\theta) \\ v_c \sin(\theta) \\ w_c \end{bmatrix} \tag{5}$$

where, the vector (x, y, θ) defines the posture of the vehicle; (x, y) are the coordinates of the middle point between the driving wheels and θ denotes the heading of the vehicle relative to the x-axis of the world coordinate system, v_c and w_c are the control signals. That means the linear speed (v_c) and the angular speed (w_c) are the control signals (Scaglia et al., 2009; Capito et al., 2016).

To obtain the controller using the linear algebra methodology, we begin with the approximation of the state equations using some numerical method, in this case the Euler approximation is used:

$$\begin{bmatrix} x_{(n+1)} \\ y_{(n+1)} \\ \theta_{(n+1)} \end{bmatrix} = \begin{bmatrix} x_{(n)} \\ y_{(n)} \\ \theta_{(n)} \end{bmatrix} + T \begin{bmatrix} v_{c(n)} \cos\left(\theta_{(n)}\right) \\ v_{c(n)} \sin\left(\theta_{(n)}\right) \\ w_{c(n)} \end{bmatrix} \tag{6}$$

Then, from (6), we must consider the control problem as solving a system of linear equations, only the terms that depend on the control actions must be moved to the first member.

$$\begin{bmatrix} \cos\left(\theta_{(n)}\right) & 0 \\ \sin\left(\theta_{(n)}\right) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{c(n)} \\ w_{c(n)} \end{bmatrix} = \frac{1}{T} \begin{bmatrix} x_{(n+1)} - x_{(n)} \\ y_{(n+1)} - y_{(n)} \\ \theta_{(n+1)} - \theta_{(n)} \end{bmatrix} = \frac{1}{T} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} \tag{7}$$

A system of three equations with two variables that generally have no solution is presented in (7). The conditions for the system to have exact solution must be determined.

First, an adjustment parameter is added to reduce the error to zero in a regulated and smooth form using the following expressions:

$$x_{(n+1)} = x_{ref(n+1)} - k_x \left(x_{ref(n)} - x_{(n)} \right) = x_{ref(n+1)} - k_x e_{x(n)} \tag{8}$$

$$y_{(n+1)} = y_{ref(n+1)} - k_y \left(y_{ref(n)} - y_{(n)} \right) = y_{ref(n+1)} - k_y e_{y(n)} \tag{9}$$

where (k_x, k_y) are the adjustment parameters which depend on the error value between the reference value and the current value of each state variable. In addition, (k_x, k_y) are positive constants, $0 < (k_x, k_y) < 1$, which allow for adjusting the performance of the proposed control system, reducing the variations in the state variables.

Then, to have an exact solution for the system of equations the vectors that multiply the second member, of the previous equations, must be parallel to each other and have a $\theta ez_{(n)}$ orientation, as follows:

$$\tan\left(\theta ez_{(n)}\right) = \frac{\sin\left(\theta ez_{(n)}\right)}{\cos\left(\theta ez_{(n)}\right)} = \frac{\Delta y}{\Delta x} \tag{10}$$

$$\tan\left(\theta ez_{(n)}\right) = \frac{y_{ref(n+1)} - k_y e_{y(n)} - y_{(n)}}{x_{ref(n+1)} - k_x e_{x(n)} - x_{(n)}} \tag{11}$$

Like (8) and (9), an adjustment parameter k_θ is added to regulate the error variation for the θ case as follows:

$$\theta_{(n+1)} = \theta ez_{(n+1)} - k_\theta \left(\theta ez_{(n)} - \theta_{(n)} \right) = \theta ez_{(n+1)} - k_\theta e_{\theta(n)} \tag{12}$$

To solve the system of equations, the normal equations $A^T Ax = A^T B$ (Strang, 1980) so only the control variables are in the first member.

$$\begin{bmatrix} v_{c(n)} \\ w_{c(n)} \end{bmatrix} = \frac{1}{T} \begin{bmatrix} \cos\left(\theta_{(n)}\right) \sin\left(\theta_{(n)}\right) \\ 0 \quad\quad 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} \tag{13}$$

$$\begin{bmatrix} v_{c(n)} \\ w_{c(n)} \end{bmatrix} = \frac{1}{T} \begin{bmatrix} \frac{x_{ref(n+1)} - k_x e_{x(n)} - x_{(n)}}{T} \cos\left(\theta ez_{(n)}\right) + \frac{y_{ref(n+1)} - k_y e_{y(n)} - y_{(n)}}{T} \sin\left(\theta_{(n)}\right) \\ \frac{\theta ez_{(n+1)} - k_\theta e_{\theta(n)} - \theta_{(n)}}{T} \end{bmatrix} \tag{14}$$

Remark 1: To compute $\omega_{c(n)}$, the value of $\theta ez_{(n+1)}$ is required but what (14) allows to calculate is $\theta ez_{(n)}$. However, $\theta ez_{(n+1)}$ can be estimated using the Taylor's formula:

$$\theta ez_{(n+1)} = \theta ez_{(n)} + \frac{d\theta ez}{dt} T + \frac{d^2 \theta ez}{dt^2} \frac{T^2}{2} + \cdots + C \tag{15}$$

where C is the complementary term (Hildebrand, 1987). So, if the sampling time is small, $\theta ez_{(n+1)}$ can be estimated in one of following ways:

$$\theta ez_{(n+1)} \approx \theta ez_{(n)} \tag{16}$$

$$\theta ez_{(n+1)} \approx \theta ez_{(n)} + \frac{d\theta ez}{dt} T \approx 2\theta ez_{(n)} - \theta ez_{(n-1)} \tag{17}$$

$$\theta ez_{(n+1)} \approx \theta ez_{(n)} + \frac{\theta ez_{(n)} - \theta ez_{(n-1)}}{T} T + \frac{\theta ez_{(n)} - 2\theta ez_{(n-1)} - \theta ez_{(n-2)}}{T} \frac{T^2}{2} \tag{18}$$

In general, the approximation in (16) provides excellent results.

3.2.2 Second hypothesis

The linear and angular speeds are not the control signals, but they are related through an unknown model, such as a FOPDT model, as follows:

$$G(s) = \frac{y(s)}{u(s)} = \frac{ke^{-t_0 s}}{\tau s + 1} \tag{19}$$

For each case:

$$G_v(s) = \frac{v(s)}{v_c(s)} = \frac{k_v e^{-t_{0v} s}}{\tau_v s + 1} \tag{20}$$

$$G_w(s) = \frac{w(s)}{w_c(s)} = \frac{k_w e^{-t_{0w} s}}{\tau_w s + 1} \tag{21}$$

The dead time term is replaced in the transfer function of linear speed (Camacho et al., 1997) as follows:

$$G_v(s) = \frac{v(s)}{v_c(s)} \approx \frac{k_v}{(\tau_v s + 1)(t_{0v} s + 1)} \tag{22}$$

To simplify, two parameters are chosen: K_{Av} and K_{Bv} include the characteristic parameters (t_{0v}, τ_v) as follows:

$$K_{Av} = \frac{t_{0v} + \tau_v}{t_0 \tau} [=][\text{time}]^{-1} \tag{23}$$

$$K_{Bv} = \frac{1}{t_{0v} \tau_v} [=][\text{time}]^{-2} \tag{24}$$

Therefore, (22) can be rewritten as:

$$G_v(s) = \frac{v(s)}{v_c(s)} = \frac{k_v K_{Bv}}{s^2 + K_{Av} s + K_{Bv}} \tag{25}$$

Then, (25) can be represented in the differential equation form as follows:

$$\ddot{v} + K_{Av} \dot{v} + K_{Bv} v = k_v K_{Bv} v_c \tag{26}$$

Similarly, for the angular speed, the following is obtained:

$$\ddot{w} + K_{Aw} \dot{w} + K_{Bw} w = k_w K_{Bw} w_c \tag{27}$$

Consequently, the model is: $x_1 = v$; $x_2 = \dot{v}$; $x_3 = \dot{w}$.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_1 \cos\theta \\ x_1 \sin\theta \\ x_3 \\ x_2 \\ k_v K_{Bv} v_c - K_{Av} x_2 - K_{Bv} x_1 \\ x_4 \\ k_w K_{Bw} w_c - K_{Aw} x_4 - K_{Bw} x_3 \end{bmatrix} \tag{28}$$

Applying the procedure described in the previous subsection, the Euler approximation is used to clear the discretised control action.

$$\begin{bmatrix} x_{(n+1)} \\ y_{(n+1)} \\ \theta_{(n+1)} \\ x_{1(n+1)} \\ x_{2(n+1)} \\ x_{3(n+1)} \\ x_{4(n+1)} \end{bmatrix} = \begin{bmatrix} x_{(n)} \\ y_{(n)} \\ \theta_{(n)} \\ x_{1(n)} \\ x_{2(n)} \\ x_{3(n)} \\ x_{4(n)} \end{bmatrix} + T \begin{bmatrix} x_{1(n)} \cos\left(\theta_{(n)}\right) \\ x_{1(n)} \sin\left(\theta_{(n)}\right) \\ x_{3(n)} \\ x_{2(n)} \\ k_v K_{Bv} v_{c(n)} - K_{Av} x_{2(n)} - K_{Bv} x_{1(n)} \\ x_4 \\ k_w K_{Bw} w_{c(n)} - K_{Aw} x_{4(n)} - K_{Bw} x_{3(n)} \end{bmatrix} \tag{29}$$

A system of linear equations where the unknown variables are the control actions can now be solved:

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ k_v K_{Bv} & 0 \\ 0 & k_w K_{Bw} \end{bmatrix} = \begin{bmatrix} v_{c(n)} \\ w_{c(n)} \end{bmatrix} \begin{bmatrix} x_{(n+1)} - x_{(n)} - T x_{1(n)} \cos\left(\theta_{(n)}\right) \\ y_{(n+1)} - y_{(n)} - T x_{1(n)} \sin\left(\theta_{(n)}\right) \\ \theta_{(n+1)} - \theta_{(n)} - T x_{3(n)} \\ x_{1(n+1)} - x_{1(n)} - T x_{2(n)} \\ x_{3(n+1)} - x_{3(n)} - T x_{4(n)} \\ \dfrac{x_{2(n+1)} - x_{2(n)}}{T} + K_{Av} x_{2(n)} - K_{Bv} x_{1(n)} \\ \dfrac{x_{4(n+1)} - x_{4(n)}}{T} - K_{Av} x_{4(n)} - K_{Bv} x_{3(n)} \end{bmatrix} \tag{30}$$

In order for the system to have an exact solution (Strang, 1980), the first five terms should be zero. Therefore, the three first terms are:

$$x_{(n+1)} = x_{(n)} + T x_{1(n)} \cos\left(\theta_{(n)}\right) \tag{31}$$

$$y_{(n+1)} = y_{(n)} + T y_{1(n)} \sin\left(\theta_{(n)}\right) \tag{32}$$

$$\theta_{(n+1)} = \theta_{(n)} + T x_{3(n)} \tag{33}$$

where the unknown variables are $x_{1(n)}$ and $x_{3(n)}$, i.e., the linear and angular velocities as in the first hypothesis. Therefore, the solution will be the same as (14), except in this case

$x_{1ez(n)}$ and $x_{3ez(n)}$ represent the speed references that will be used for the calculation of the control actions.

$$\begin{bmatrix} x_{1ez(n)} \\ x_{3ez(n)} \end{bmatrix} = \begin{bmatrix} \frac{\Delta x}{T}\cos\left(\theta ez_{(n)}\right) + \frac{\Delta y}{T}\cos\left(\theta ez_{(n)}\right) \\ \frac{\Delta\theta}{T} \end{bmatrix} \tag{34}$$

where $x_{1ez(n)}$ and $x_{3ez(n)}$ are the references for the state variables x_1 and x_3 (see Remark 1).

The fourth and fifth terms also should be zero, therefore, it is necessary to satisfy:

$$x_{1(n+1)} - x_{1(n)} + Tx_{2(n)} = 0 \tag{35}$$

$$x_{3(n+1)} - x_{3(n)} + Tx_{4(n)} = 0 \tag{36}$$

Because, the reference signals for x_1 and x_3 were obtained from (34) and it is important that linear and angular speeds gradually approach their reference values, $x_{1(n+1)}$ is replaced for $x_{1ez(n+1)} - k_1e_{1(n)}$ and $x_{3(n+1)}$ for $x_{3ez(n+1)} - k_3e_{3(n)}$, therefore:

$$x_{2ez(n)} = \frac{x_{1ez(n+1)} - k_1e_{1(n)} - x_{1(n)}}{T} \tag{37}$$

$$x_{4ez(n)} = \frac{x_{3ez(n+1)} - k_3e_{3(n)} - x_{3(n)}}{T} \tag{38}$$

where $x_{2ez(n)}$ and $x_{4ez(n)}$ are the references for the state variables x_2 and x_4, respectively (see Remark 1). And, (k_1, k_2) are positive constants, which adjust the performance of the proposed control system; they satisfy $0 < (k_1, k_2) < 1$, reducing the variations in the state variables.

Finally, from (30), (34), (37) and (38):

$$v_{c(n)} = \left(\frac{x_{2ez(n+1)} - k_2e_{2(n)} - x_{2(n)}}{T} + K_{Av}x_{2(n)} + K_{Bv}x_{1(n)}\right) \Big/ K_v k_{Bv} \tag{39}$$

$$w_{c(n)} = \left(\frac{x_{4ez(n+1)} - k_4e_{4(n)} - x_{4(n)}}{T} + K_{Aw}x_{4(n)} + K_{Bw}x_{3(n)}\right) \Big/ K_w k_{Bv} \tag{40}$$

4 Results

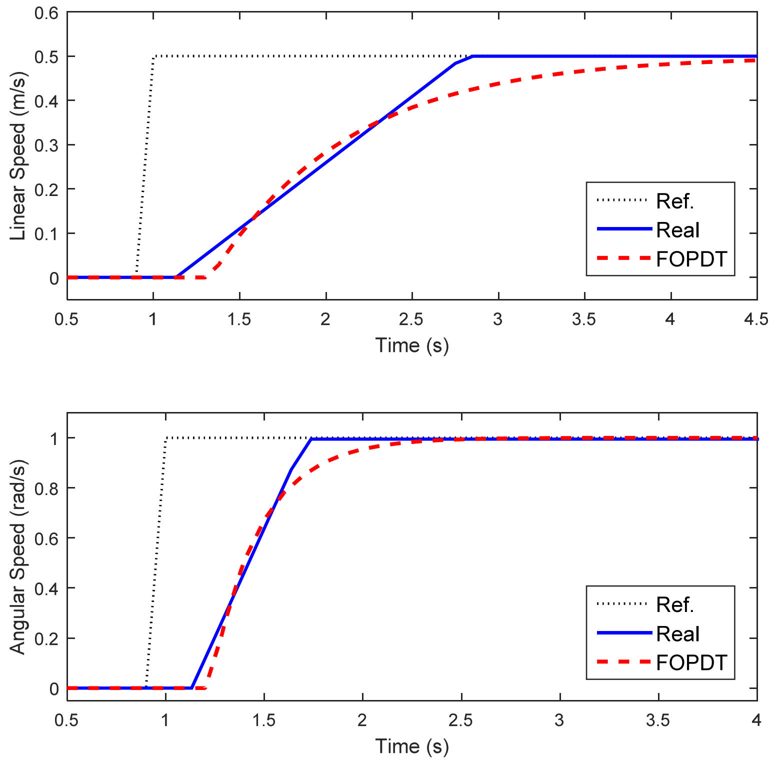
In this section the proposed approach is tested. In the first section part, a comparison by simulations is done and then a realistic experiment is developed. For both cases the FOPDT model of the robot, used to design the proposed controller, is obtained. For the simulation experiment, the approach based on the reduced order model of the robot is compared against a conventional linear algebra controller designed using the complete model of the robot. For experimental results, the Pioneer 3DX was used. It is compared the performance of the LACr controller against a PI controller. The realistic results for tracking two trajectories (a squared one and a circular one) are presented. The integral of squared error (ISE) index is used to measure the performance for the controllers.

4.1 Empirical model of the robot from reaction curve method

Considering that the Pioneer 3DX is a black box process, only the input and output are known. Then, using an identification procedure, the characteristic parameters (k, τ, t_0) required to represent the robot as a FOPDT model are obtained. Finally, the mobile robot behaviour is represented by two transfer functions for linear and angular speeds.

The real response of the robot and the approximate response to a step input for each speed are presented in Figure 4.

Figure 4 Linear and angular speeds step responses (see online version for colours)



The transfer functions which represent the approximate linear speed u and angular speed ω are represented by (38) and (39), respectively. The transfer function of the angular speed is considered to have a significant controllability relationship (t_0/τ).

$$G_u(s) = \frac{1}{0.5s+1}e^{-0.22s} \tag{41}$$

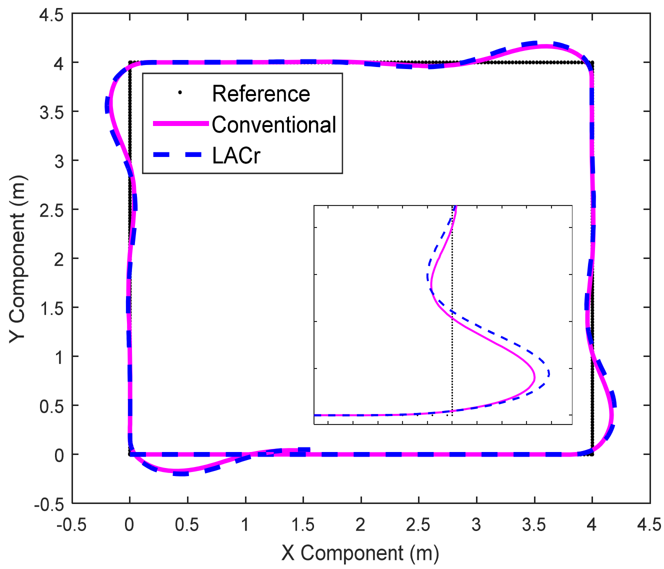
$$G_\omega(s) = \frac{1}{0.15s+1}e^{-0.32s} \tag{42}$$

4.2 Linear algebra controllers comparison by simulation

In this subsection, the proposed controller based on an approximate model is compared with the conventional linear algebra controller based on the robot's kinematic model. The conventional controller applied to trajectory tracking with the Pioneer 3DX robot is taken from Capito et al. (2016).

Figures 5 and 6 show the response of both controllers for a square trajectory. A square trajectory is tested, setting a linear speed set point of 0.3 m/s.

Figure 5 XY graph of square trajectory tracking (see online version for colours)



Similar results shown in the graph prove that despite the use of a linear approximate model in the proposed controller, the robot's response is comparable to the conventional controller.

4.3 Realistic results

In this subsection, proposed controller LACr is compared to a traditional PI. The two different tracking trajectories are tested: a square with each side of 4 m and a circle with a radius of 4 m. In both trajectories, a set point of linear speed of 0.3 m/s is applied. Figure 7 shows the robot's positions during the experiments.

4.3.1 Square trajectory

Figure 8 illustrates the simultaneously behaviour of the two controllers on the reference square trajectory. Figure 9 shows the variation of linear and angular speeds during trajectory tracking. Figure 10 shows the position error in each component and error norm during trajectory tracking.

Figure 6 Linear and angular speed vs. time (see online version for colours)

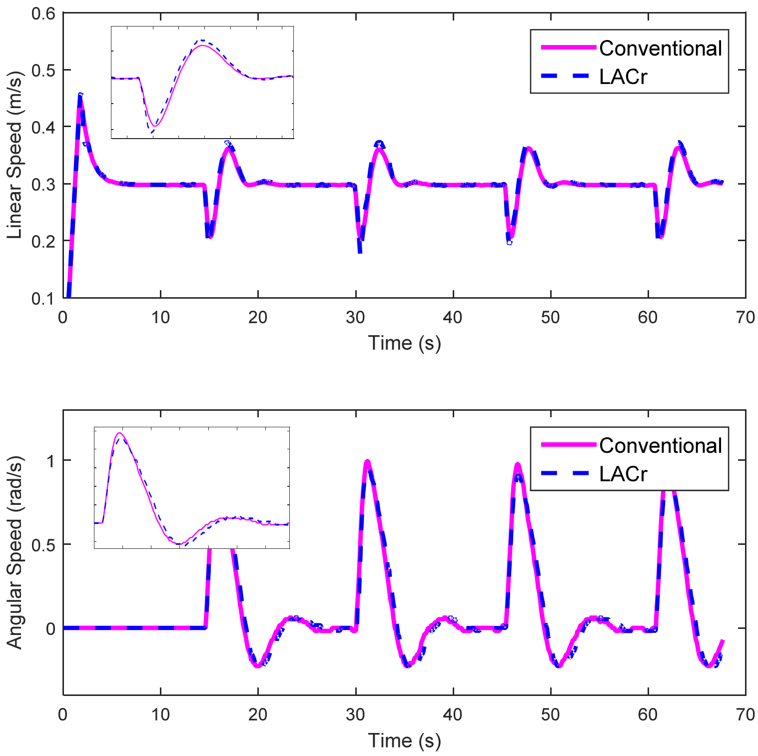
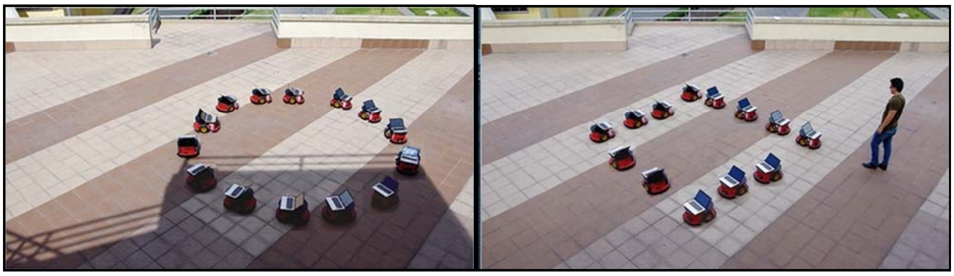


Figure 7 Robot position changes during trajectory tracking (see online version for colours)



As observed in the previous figure, the PI controller presents a better response for the linear speed case. But, for the angular speed case, which has a controllability relationship close to two, the proposed approach is faster with lower overshoot when compared to PI.

Figure 8 XY graph of square trajectory tracking (see online version for colours)

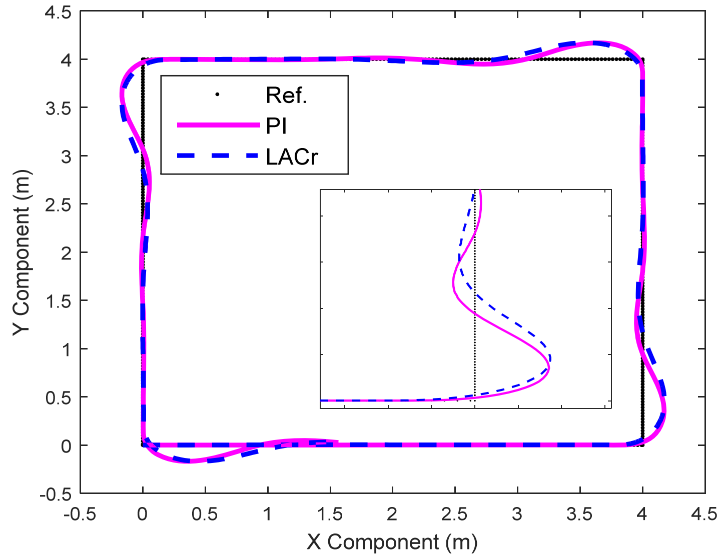


Figure 9 Linear and angular speed vs. time (see online version for colours)

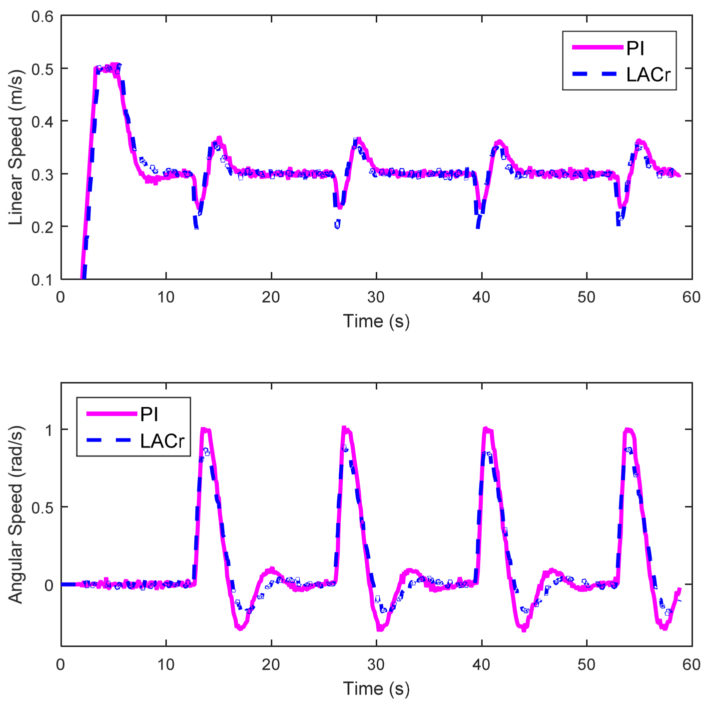
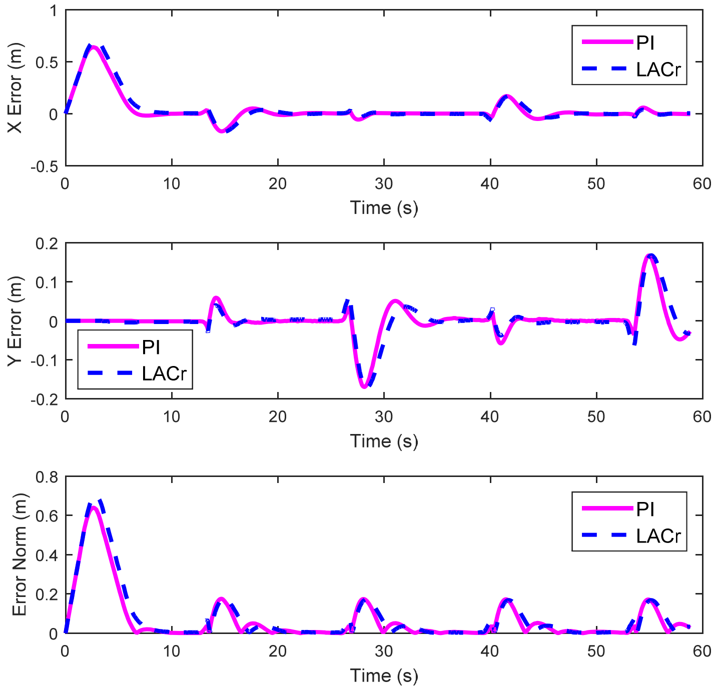


Figure 10 Position error vs. time (see online version for colours)



4.3.2 Circular trajectory

In a similar way, as presented for the square trajectory tracking, Figures 11 to 13 are graphs of trajectory tracking.

Like Figure 9, Figure 12 shows that the PI controller responds better for the linear speed case. For the angular speed case, which has a controllability relationship close to two, however, the proposed approach is faster with lower overshoot when compared to PI.

4.4 Performance evaluation

Here, a performance index compares and evaluates the proposed approach against a conventional one, the simulation case, and the experimental part when a PI controller is used.

Table 1 ISE comparison of linear algebra controllers by simulations

Trajectory	*Component*	*ISE*	
		LACr	*Conventional LACr*
Square	e_x	0.1157	0.1297
	e_y	0.09316	0.0933
	e_v	0.0993	0.1021
	e_w	0.2551	0.2523

Figure 11 XY graph of square trajectory tracking (see online version for colours)

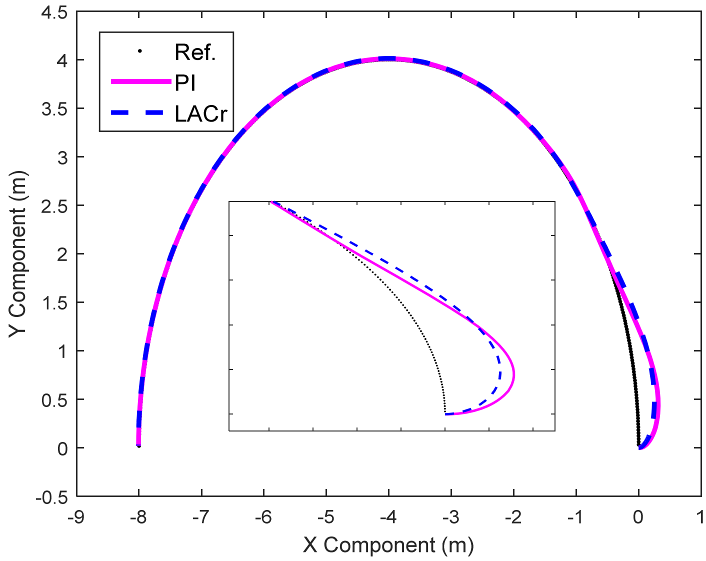


Figure 12 Linear speed and angular speed vs. time (see online version for colours)

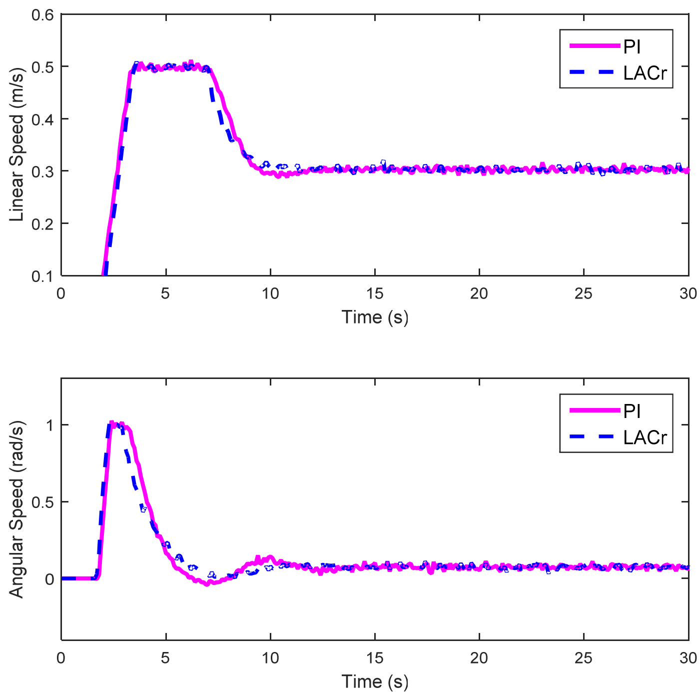


Figure 13 Position error vs. time (see online version for colours)

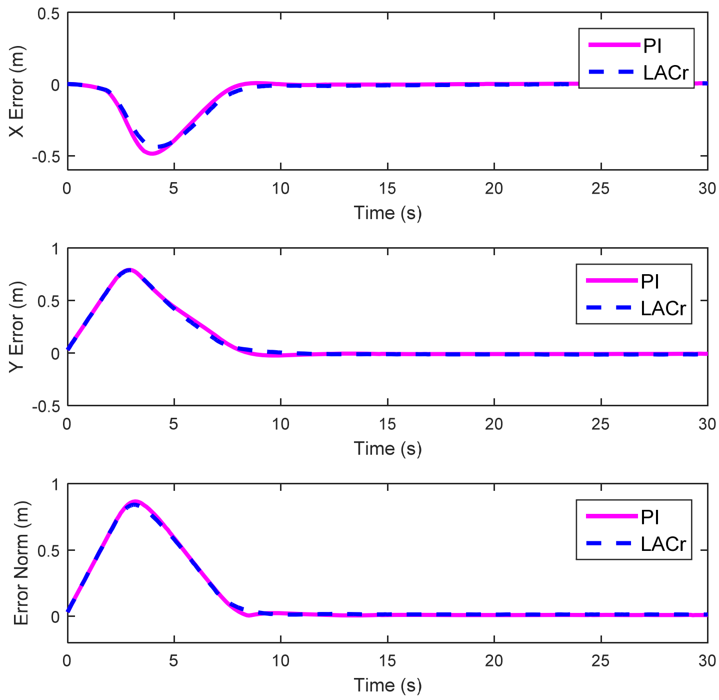


Table 2 ISE comparison of realistic trajectory tracking

Trajectory	*Component*	*ISE*	
		PI	*LACr*
Square	e_x	1.1445	1.4014
	e_y	0.0926	0.081
	e_v	1.3743	1.3701
	e_w	0.4573	0.1164
Circle	e_x	0.5829	0.5336
	e_y	1.8225	1.7816
	e_v	2.2751	2.2417
	e_w	5.5697	5.1885

The ISE index is calculated from the error in each component separately (e_x, e_y) and is defined as follows:

$$\overline{ISE} = \frac{\int_0^T e^2(t)dt}{T} \tag{43}$$

Table 1 shows the results of the performance, by simulations, between the proposed controller and the conventional one. For this, the results are very close.

Table 2 summarises the results for the real experiments performed for two trajectories. Experiments have shown that both controllers produce similar trajectory tracking quality but using the performance index the LACr produces lower ISE values. Also, it is important to mention that the proposed approach presented a better result for the angular speed when the controllability relationship was greater than two.

5 Conclusions

A controller using linear algebra methodology was presented in this work. For designing purposes, a first order plus dead time model of the robot was used instead of the complete model of the mobile robot.

The comparison, between the conventional controllers based on the complete model to the proposed controller based on a linear approximate model, showed that despite the empirical modelling procedure, responses were very close.

It is not difficult to tune the proposed controller, since its parameters come from the FOPDT process model. It is also simple to implement with low computational requirements making it especially appropriate for fast, real-time trajectory tracking. Experimental results have demonstrated that the proposed controller is an option to track trajectories in a faster and simpler manner.

The proposed controller works for tracking and regulation tasks for different processes in diverse fields such as robotics and chemical processes, if the systems can be approximated for an open-loop dynamic like a FOPDT response. The proposed approach presented a better result for the angular speed case when the controllability relationship is greater than 2. Intelligent computational procedures are considered for determining optimal controller tuning parameters will be studied.

Acknowledgements

OC thanks to PROMETEO Project of SENESCYT, Republic of Ecuador, for its support for the realisation of this work. This work was partially funded by the Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina. The authors would like to thank the Universidad de Las Fuerzas Armadas – ESPE, Ecuador, for providing them a robot Pioneer 3DX to conduct the experimental tests included in this paper

References

Brennan, S. and Alleyne, A. (2002) 'H-infinity vehicle control using non-dimensional perturbation measures', *American Control Conference*, pp.2534–2539.

Camacho, O. and Smith, C.A. (2000) 'Sliding mode control: An approach to regulate nonlinear chemical process', *ISA Transactions*, Vol. 39, No. 2, pp.205–218.

Camacho, O., Smith, C. and Chacón, E. (1997) 'Toward an implementation of sliding mode control to chemical processes', *IEEE International Symposium on Industrial Electronics*, pp.1101–1105.

Capito, L., Proaño, P., Camacho, O., Rosales, A. and Scaglia, G. (2016) 'Experimental comparison of control strategies for trajectory tracking for mobile robots', *International Journal of Control and Automation*, Vol. 10, No. 3, pp.308–327.

Guevara, L., Guevara, J., Scaglia, G., Camacho, O. and Rosales, A. (2016a) 'A New Approach of a Numerical Methods Controller for Self-Regulating Processes', IEEE Biennial Congress of Argentina (ARGENCON). DOI: 10.1109/ARGENCON.2016.7585282

Guevara, J., Guevara, L., Camacho, O., Scaglia, G. and Rosales, A. (2016b) 'An approach of a numerical methods controller for nonlinear chemical processes', *Impact and Advances of Automatic Control in Latinoamerica*, Artes y Letras S.A.S, Colombia, Chapter 1, pp.52–58.

Hedjar, R., Toumi, R. and Boucher, P. (2005) 'Finite horizon nonlinear predictive control by the Taylor approximation', *International Journal of Applied Mathematics and Computer Science*, Vol. 15, No. 4, pp.527–540.

Hildebrand, F.B. (1987) *Introduction of Numerical Analysis*, 2nd ed., Dover Publications, Inc., New York.

Kühne, F., Gomes, J. and Fetter, W. (2005) 'Mobile robot trajectory tracking using model predictive control', *II IEEE LARS*, São Luís, pp.1–7.

Pantano, M., Serrano, M., Fernández, M., Rossomando, F., Ortiz, O. and Scaglia, G.J. (2017) 'Multivariable control for tracking optimal profiles in a nonlinear fed-batch bioprocess integrated with state estimation', *Industrial and Engineering Chemistry Research*, Vol. 56, No. 20, pp.6043–6056, ISSN: 0888-5885.

Quintero, O.L., Amicarelli, A.A., Scaglia, G. and Di Sciascio, F. (2009) 'Control based on numerical methods and recursive Bayesian estimation in a continuous alcoholic fermentation process', *BioResources*, Vol. 4, No. 4, pp.1372–1395.

Rómoli, S., Serrano, M., Rossomando, F., Vega, J., Ortiz, O. and Scaglia, G. (2017) 'Neural network-based state estimation for a closed-loop control strategy applied to a fed-batch bioreactor', *Complexity*, John Wiley and Sons Inc., ISSN: 1076-2787, vol. 2017, Article ID 9391879, 16 pp., DOI: 10.1155/2017/9391879 [online] https://doi.org/10.1155/2017/9391879 (accessed 9 October 2017).

Rómoli, S., Serrano, E., Ortiz, O., Vega, J. and Scaglia, G. (2015) 'Tracking control of concentration profiles in a fed-batch bioreactor using a linear algebra methodology', *ISA Transactions*, Vol. 57, pp.162–171 [online] http://dx.doi.org/10.1016/j.isatra.2015.01.002.

Rosales, A., Scaglia, G., Mut, V. and Di Sciascio, F. (2010) 'Formation control and trajectory tracking of mobile robotic systems – a linear algebra approach', *Robotica*, Vol. 29, No. 3, pp.335–349.

Scaglia, G., Aballay, P., Serrano, E., Ortiz, O., Jordan, M. and Vallejo, M. (2014) 'Linear algebra based controller design applied to a bench-scale oenological alcoholic fermentation', *Control Engineering Practice*, Vol. 25, pp.66–74 [online] http://dx.doi.org/10.1016/j.conengprac.2014.01.002.

Scaglia, G., Quintero, L., Mut, V. and Di Sciascio, F. (2009) 'Numerical methods based controller design for mobile robots', *Robotica*, Vol. 27, No. 2, pp.269–279.

Scaglia, G., Serrano, E., Rosales, A., and Albertos, P. (2015) 'Linear interpolation based controller design for trajectory tracking under uncertainties: application to mobile robots', *Control Engineering Practice*, Vol. 45, pp.123–132 [online] http://dx.doi.org/10.1016/j.conengprac.2015.09.010.

Serrano, E., Scaglia, G., Auat, F., Mut, V. and Ortiz, O. (2015) 'Trajectory-tracking controller design with constraints in the control signals: a case study in mobile robots', *Robotica*, Vol. 33, No. 10, pp.2186–2203.

Serrano, M., Godoy, S., Rómoli, S. and Scaglia, G. (2017) 'A numerical approximation-based controller for mobile robots with velocity limitation', *Asian Journal of Control. Version of Record Online*, 11 April 2017, DOI: 10.1002/asjc.1522, ISSN: 1934-6093.

Shuli, S. (2005) 'Designing approach on trajectory tracking control of mobile robots', *Robotics and Computer Integrated Manufacturing*, Vol. 21, No. 1, pp.81–85.

Smith, C. and Corripio, A.B. (1997) *A Principles and Practice of Automatic Process Control*, John Wiley and Sons, Inc., New York.

Strang, G. (1980) *Linear Algebra and its Applications*, Academic Press, New York.

Suvire, R., Scaglia, G., Serrano, E., Vega, J. and Ortiz, O. (2013) 'Trajectory tracking in nonlinear CSTR, controller design based on linear algebra approach', *IEEE Biennial Congress of Argentina (ARGENCON)*, pp.67–72, DOI: 10.1109/ARGENCON.2014.6868473.