

Review Article

Tracking Control Design in Nonlinear Multivariable Systems: Robotic Applications

Gustavo Scaglia ¹, Emanuel Serrano ¹, Andres Rosales,² and Pedro Albertos ³

¹National Council of Scientific and Technological Research (CONICET), Universidad Nacional de San Juan, San Juan, Argentina

²Escuela Politecnica Nacional, Quito, Ecuador

³Departamento de Ingeniería de Sistemas y Automática, Universidad Politécnica de Valencia, Valencia, Spain

Correspondence should be addressed to Gustavo Scaglia; gscaglia@unsj.edu.ar

Received 17 May 2019; Revised 14 July 2019; Accepted 24 July 2019; Published 19 August 2019

Academic Editor: Sajad Azizi

Copyright © 2019 Gustavo Scaglia et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this work, a controller design technique called linear algebra based controller (LABC) is presented. The controller is obtained following a systematic procedure that is summarized in this work. In addition, the influence of additive uncertainty on the tracking error is analyzed, and a solution using integrators is proposed. A mobile robot is used as a benchmark to test the performance of the proposed algorithms. In addition, implementation to other systems such as marine vessel is referenced. In this work, the design of controllers in continuous and discrete time is included and experimental and simulation results are shown in a Pioneer 3AT mobile robot. Comparisons are also shown with other controllers proposed in the literature.

1. Introduction

There are many control applications where the target is to follow a given trajectory for some of the process variables. This is the usual goal when dealing with robotic applications, where the robot itself or its end effector must follow a prescribed trajectory to perform some specific activities like, for instance, painting, welding, or cutting. But not for all the state variables, there is a defined trajectory, and for those remaining variables, there is some freedom to fix their time evolution.

Assuming an internal representation of the process, a nominal linear model could be expressed as

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t), \\ x(s) &= (sI - A)^{-1}Bu(s) = G(s)u(s),\end{aligned}\quad (1)$$

where $x \in \mathcal{R}^n$ is the vector of state variables, $u \in \mathcal{R}^m$ is the vector of control inputs, and A and B are matrices of appropriate dimensions. In general, $m < n$ and thus $G(s)$, the transfer matrix, is not square and the control action cannot

be directly derived from (1) even in the case that the desired trajectory of the full state vector, $x_d(s)$, is given ($u(s) = G^{-1}(s)x_d(s)$). Moreover, the model of the process could be nonlinear and in that case, the state-space representation should be expressed as

$$\dot{x}(t) = F(x(t), u(t), t). \quad (2)$$

The tracking control problem has deserved a lot of attention in the last years ([1–6]), and it has been usually tackled as a two-degrees-of-freedom problem. A model-based feedforward control is implemented in order to generate a feasible trajectory and a feedback control tries to reduce the tracking error. There is a lot of research dealing with this problem (see, for instance, [7]). Clearly, there is no general solution for the more general problem, where there is a reference signal $r(t)$ for some state variables, the process model is nonlinear, and there are uncertainties and external disturbances. Some approaches have been proposed to deal with specific problems.

In [8], a novel trajectory-tracking controller for a mobile robot was presented. A crucial assumption is that the robot model is affine in the control. The proposed technique is based on the calculation of control actions by solving a system of linear equations. The objective is to follow reference coordinates $(x_{\text{ref}}, y_{\text{ref}})$, and the unknowns are the linear and angular speed of the robot. A fundamental point is to find the conditions such that the system of linear equations has an exact solution. In this way, the desired value of the remaining state variables, that are named sacrificed variables, is determined. The condition that the system has an exact solution ensures that the reference trajectory together with the sacrificed variables forms an admissible trajectory of the system. This ensures that the tracking error tends to zero. LABC has been applied to several robotic systems. It has also been used in the formation of mobile robots where it is ensured that the formation error tends to zero. In [9], the proposed methodology was applied for trajectory tracking in a marine vessel. Simulations results demonstrate the good performance of the controller against environmental disturbances induced by wave, wind, and ocean current. In [10], a controller, based on the linear algebra methodology, allows a quad-rotor helicopter to track a desired trajectory in 3D space. This methodology has been accepted and applied by different authors (see, for instance [11, 12]), being referenced as an alternative to solve the path tracking problem (see, for instance [13–15]). In [16], the design in CT is presented using the kinematic model of the mobile robot. In that study, we have only one sacrificed variable, and different approximations were used. The higher the order of the approximations is, the quicker the error tends to zero. In [17], the design is made using the dynamic model of the mobile robot; in this case, there are three sacrificed variables. In this case, having three sacrificed variables, only a zero-order approximation is used to avoid increasing the calculation time.

Based on the previous results and applications, the aim of this paper is to provide a formal design methodology for tracking controllers in nonlinear multivariable plants with uncertainties in the model. In the next section, the controller design methodology is summarized. Then, the details to be applied in continuous time (CT) to a mobile by using either the kinematic or the dynamic model are presented. Clearer results are obtained if a discrete time (DT) model is considered. Different discretization options are used, leading to controllers with enhanced properties. Initially, a perfect model of the process is assumed. Model uncertainties as well as external disturbances are represented by including an additional term in the state equation. Based on the characteristics of this additional term, different kinds of uncertainties can be considered.

Most controllers are derived by using the discrete state-space model equations. The main advantage of this approach is the simplicity of the controllers, and the use of discrete time equations makes its implementation suitable for a computer-based control system.

2. Controller Design Methodology

In this section, the LABC methodology considered in this paper is presented. The procedure described below outlines the steps that must be followed to design a controller based on linear algebra for a multivariable nonlinear system affine in the control. Initially, the model is assumed to be perfect (no uncertainties or missing dynamics), and there are no external disturbances. The control is based on the system state which is assumed to be measured. The methodology can be applied either in CT or DT.

The design methodology can be summarized in five steps:

- (1) First, it is necessary to represent the mathematical model of the CT system in state form as a set of first-order differential equations, denoting $x \in \mathcal{R}^r$ as the state vector. The state functions are affine in the control, $u \in \mathcal{R}^m$. If the design is carried out in DT, the plant model must be discretized using a numerical method (for example, Euler, trapezoidal, or any other).
- (2) The state vector is split into two parts: the set of variables whose trajectory should be tracked ($z \in \mathcal{R}^m$) and the rest ($\bar{z} \in \mathcal{R}^{r-m}$), denoted as sacrificed variables. This name points out that their reference value is adjusted to allow an algebraic computation of the control action. The control problem is formulated as solving a system of equations linear in the control ($\mathbf{A}\mathbf{u} = \mathbf{b}$) where \mathbf{A} is a matrix of (possibly nonlinear) functions of the state variables and the system parameters, \mathbf{u} is a vector composed by the control actions, and \mathbf{b} is a vector of (possibly nonlinear) functions of the state variables and the system parameters.
- (3) The derivative of the state variables required to be tracked ($z(t)$) are replaced by that of the reference variables assuming a smooth approaching, for instance, proportional to the error. In continuous time, the replacement is done according to $\dot{z} = \dot{z}_{\text{ref}} + k_z(z_{\text{ref}} - z)$; in discrete time, the replacement is $z_{n+1} = z_{\text{ref},n+1} - k_z(z_{\text{ref},n} - z_n)$. In this setting, z represents a tracked state variable, z_n is its value at time instant n , k_z is a controller parameter associated to this variable, z_{ref} represents the desired value of this variable, and $z_{\text{ref},n}$ is its desired value in discrete time at time instant n . Next, the reference trajectory of the sacrificed state variables should be determined. The reference for these variables is defined with the subscript ez , and an approach proportional to the error is also assumed for their derivative.
- (4) The goal is to find the current value of the control actions such that the trajectory-tracking error tends to zero. To accomplish that, the system must have exact solution. Then, the vector \mathbf{b} must be contained in the space formed by the columns of \mathbf{A} , i.e., the vector \mathbf{b} must be a linear combination of the column vectors of matrix \mathbf{A} [18]. In this way, the conditions,

when the system $\mathbf{A}\mathbf{u} = \mathbf{b}$ has exact solution, are defined leading to the assumed references for the sacrificed state variables.

- (5) Finally, the calculation of the control actions forcing the tracking errors tend to zero is carried out. So, the control actions are obtained by solving the system $\mathbf{A}\mathbf{u} = \mathbf{b}$.

3. Controller Design in Continuous Time

In this section, the above methodology is applied to a CT affine in the control undisturbed nonlinear system.

Let us represent the plant as

$$\dot{x} = f(x) + g(x)u, \quad (3)$$

where the state is r -dimensional and there are m -inputs. Obviously, only the tracking of m state variables is possible. Let us split the state vector as

$$\dot{z} = \begin{pmatrix} \dot{z} \\ \dot{\bar{z}} \end{pmatrix} = \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix} + \begin{pmatrix} g_1(x) \\ g_2(x) \end{pmatrix} u, \quad (4)$$

where $z \in \mathcal{R}^m$ is the vector of state variables to be tracked, following the desired reference z_{ref} . f_i, g_i are matrices of appropriate dimension.

According to Step 2 exposed in the methodology summary in Section 2, the reference for the sacrificed state variables, \bar{z}_{ez} , will be computed to provide a solution of the set of equations:

$$\begin{pmatrix} \dot{z}_{\text{ref}} + \mathbf{K}_1(z_{\text{ref}} - z) - f_1(x) \\ \dot{\bar{z}}_{ez} + \mathbf{K}_2(\bar{z}_{ez} - \bar{z}) - f_2(x) \end{pmatrix} = \begin{pmatrix} g_1(x) \\ g_2(x) \end{pmatrix} \mathbf{u}; \quad (5)$$

$$\mathbf{b} = \mathbf{A}\mathbf{u},$$

where $\mathbf{K}_1, \mathbf{K}_2$ are diagonal matrices with diagonal elements determining the convergence rate for each variable. That is, \bar{z}_{ez} is defined in such a way that the vector \mathbf{b} is a linear combination of the column vectors of the matrix \mathbf{A} .

Once \bar{z}_{ez} is defined, its derivative can be computed if $z_{\text{ref}}(t)$ is continuous and derivable.

Finally, the control will be computed as

$$\mathbf{u} = \mathbf{A}^\dagger \mathbf{b}, \quad (6)$$

where \mathbf{A}^\dagger stands for the pseudoinverse matrix.

3.1. Kinematic Control of a Mobile Robot. The kinematic continuous time control of a mobile robot will be used to illustrate the procedure. The details of this application can be seen in [16]. A simple nonlinear kinematic model for a mobile robot is given by (7). This kinematic model has been used by other authors recently in the bibliography [16, 19].

$$\begin{cases} \dot{x} = V \cos \theta, \\ \dot{y} = V \sin \theta, \\ \dot{\theta} = W, \end{cases} \quad (7)$$

where V is the linear velocity of the mobile robot, W is its angular velocity, (x, y) is the Cartesian position, and θ is the mobile robot orientation.

The system model given by (7) has three state variables. Two of them ($z = [x \ y]'$) must follow a pre-established trajectory defined by $(x_{\text{ref}}(t), y_{\text{ref}}(t))$, whereas the orientation ($\bar{z} = \theta$) is going to be considered as a sacrificed state variable. This model has been used in several recent papers such as [2, 20–22]. Note that the dynamics of the mobile as well as that of the actuators are not considered.

The goal is to find the values of $V(t)$ and $W(t)$ so that the mobile robot may follow a given trajectory with an acceptable error.

As previously mentioned, the reference values for the sacrificed variables must be determined to provide an exact solution to a system of linear equations. This reference value will be denoted by θ_{ez} .

Now the design steps following the proposed methodology are described:

- (i) As the design is in continuous time, equation (7) represents the first step.
- (ii) Now, the system is described in matrix form

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V \\ W \end{bmatrix}, \quad (8)$$

which is similar to (3), being $f(x) = 0$ and affine in the control.

- (iii) In the third step, the derivative of the state variables to be tracked, in this case, the robot position (x, y) is replaced by the reference position assuming a smooth approaching. A similar expression is used for the derivative of the sacrificed variable, θ , also assuming an approach proportional to the error, as seen in (9) (see Step 2 in section 2):

$$\begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V \\ W \end{bmatrix} = \begin{bmatrix} \dot{x}_{\text{ref}} + k_x(x_{\text{ref}} - x) \\ \dot{y}_{\text{ref}} + k_y(y_{\text{ref}} - y) \\ \dot{\theta}_{ez} + k_\theta(\theta_{ez} - \theta) \end{bmatrix}. \quad (9)$$

The tracking error is defined as

$$e(t) = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} x_{\text{ref}} - x \\ y_{\text{ref}} - y \\ \theta_{ez} - \theta \end{bmatrix}. \quad (10)$$

In (9), k_x , k_y , and k_θ are positive constants allowing the performance adjustment of the proposed control system. They should be $k_x > 0$, $k_y > 0$, and $k_\psi > 0$ to ensure that the tracking errors (10) tend to zero (see [16]).

- (iv) In the four step, the reference for the sacrificed variables is defined to provide an exact solution to (9). Taking into account the first two rows of (9), θ_{ez} must satisfy

$$\frac{\sin \theta_{ez}}{\cos \theta_{ez}} = \frac{\dot{y}_{\text{ref}} + k_y(y_{\text{ref}} - y)}{\dot{x}_{\text{ref}} + k_x(x_{\text{ref}} - x)}. \quad (11)$$

The orientation θ_{ez} in (11) ensures that the system (9) has exact solution, and it is computed according to

$$\theta_{ez} = a \tan \frac{\dot{y}_{\text{ref}} + k_y(y_{\text{ref}} - y)}{\dot{x}_{\text{ref}} + k_x(x_{\text{ref}} - x)}. \quad (12)$$

- (v) Finally, the control law is found by solving the system (9) by least squares. Taking into account $\mathbf{A}'\mathbf{A} = \mathbf{I}$, $\mathbf{A}^\dagger = \mathbf{A}'$. Thus,

$$\begin{bmatrix} V \\ W \end{bmatrix} = \begin{bmatrix} [\dot{x}_{\text{ref}} + k_x e_x] \cos \theta + [\dot{y}_{\text{ref}} + k_y e_y] \sin \theta \\ \dot{\theta}_{ez} + k_\theta e_\theta \end{bmatrix}. \quad (13)$$

Providing the control actions, based on the system state and the references.

Remark 1. The value of $\dot{\theta}_{ez}$ is required to compute W but (12) allows to calculate θ_{ez} . However, θ_{ez} can be computed from (12), as far as the second derivative of the references is available. It can be also estimated as the past derivative. In practical applications, if the change in the robot orientation is slow, its derivative can be neglected.

3.2. Dynamic Control of a Mobile Robot. Let us consider the dynamic model of this mobile robot [17]. To perform tasks requiring high speed and/or transport of heavy loads, it is very important to consider the dynamics of the mobile robot because such tasks result in very large external forces on the robot and they will inevitably influence its

path and direction. Hence, a kinematic model is not sufficient.

The state-space dynamic model of the mobile robot is given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{V} \\ \dot{W} \end{bmatrix} = \begin{bmatrix} V \cos \theta \\ V \sin \theta \\ W \\ \frac{a_3}{a_1} W^2 - \frac{a_4}{a_1} V \\ -\frac{a_5}{a_2} VW - \frac{a_6}{a_2} W \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{a_1} & 0 \\ 0 & \frac{1}{a_2} \end{bmatrix} \begin{bmatrix} uc \\ \omega c \end{bmatrix}, \quad (14)$$

where the tracked state variables are as before ($z = [x \ y]'$), and the sacrificed variables are $\bar{z} = [\theta \ V \ W]'$. The parameters of the dynamic model (a_i for $i\{1, 2, 3, 4, 5, 6\}$) are described in ([17, 23]).

In this case, the system has 5 states. The aim is to find the combined control actions ($uc, \omega c$) such that the mobile robot reaches and follows a reference trajectory ($x_{\text{ref}}, y_{\text{ref}}$). The trajectory of the remaining state variables (the sacrificed variables) is not so relevant.

Next, the procedure to design the controller is applied.

- (i) The control problem is described in matrix form, and then, it is analyzed as a linear algebra problem:

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{a_1} & 0 \\ 0 & \frac{1}{a_2} \end{bmatrix} \begin{bmatrix} uc \\ \omega c \end{bmatrix} = \begin{bmatrix} \dot{x} - V \cos \theta \\ \dot{y} - V \sin \theta \\ \dot{\theta} - W \\ \dot{V} - \frac{a_3}{a_1} W^2 + \frac{a_4}{a_1} V \\ \dot{W} + \left(\frac{a_5}{a_2} VW + \frac{a_6}{a_2} W \right) \end{bmatrix}. \quad (15)$$

- (ii) The derivative of the robot position (x, y) is replaced by that of the reference position ($x_{\text{ref}}, y_{\text{ref}}$) assuming an approaching proportional to the error. A similar setting is used for the derivatives of the sacrificed variables whose reference trajectories are denoted as (θ_{ez}, V_{ez} , and W_{ez}), also assuming an approaching proportional to the error

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{a_1} & 0 \\ 0 & \frac{1}{a_2} \end{bmatrix} \begin{bmatrix} uc \\ \omega c \end{bmatrix} = \begin{bmatrix} \dot{x}_{ref} + k_x(x_{ref} - x) - V \cos \theta \\ \dot{y}_{ref} + k_y(y_{ref} - y) - V \sin \theta \\ \dot{\theta}_{ez} + k_\theta(\theta_{ez} - \theta) - W \\ \dot{V}_{ez} + k_V(V_{ez} - V) - \frac{a_3}{a_1}W^2 + \frac{a_4}{a_1}V \\ \dot{W}_{ez} + k_W(W_{ez} - W) + \left(\frac{a_5}{a_2}VW + \frac{a_6}{a_2}W\right) \end{bmatrix}. \quad (16)$$

(iii) In the next step, the conditions in order that system (16) has exact solution will determine the sacrificed variables reference, leading to

$$\begin{bmatrix} \theta_{ez} \\ V_{ez} \\ W_{ez} \end{bmatrix} = \begin{bmatrix} a \tan \frac{\dot{y}_{ref} + k_y e_y}{\dot{x}_{ref} + k_x e_x} \\ [\dot{x}_{ref} + k_x e_x] \cos \theta_{ez} + [\dot{y}_{ref} + k_y e_y] \sin \theta_{ez} \\ \dot{\theta}_{ez} + k_\theta e_\theta \end{bmatrix}. \quad (17)$$

(iv) In addition, the values of $\dot{\theta}_{ez}$, \dot{V}_{ez} , and \dot{W}_{ez} can be computed as before (see Remark 1).

(v) Finally, system (16) is solved by least squares, and the control action expression arises in a natural way:

$$\begin{bmatrix} uc \\ \omega c \end{bmatrix} = \begin{bmatrix} a_1(\dot{V}_{ez} + k_V(V_{ez} - V)) - a_3W^2 + a_4V \\ a_2(\dot{W}_{ez} + k_W(W_{ez} - W)) + a_5VW + a_6W \end{bmatrix}. \quad (18)$$

As before, the control action is based on the availability of the full state vector.

4. Controller Design in Discrete Time

In this section, the above methodology is applied to design the DT control for a CT nonlinear system, with sampling period T .

(1) If the Euler approximation method is used, (3) can be written as

$$\mathbf{x}_{n+1} = \mathbf{x}_n + T[f(\mathbf{x}_n) + g(\mathbf{x}_n)\mathbf{u}_n] = \bar{f}(\mathbf{x}_n) + \bar{g}(\mathbf{x}_n)\mathbf{u}_n, \quad (19)$$

(2) After discretization, the control problem is formulated as solving a system of equations like

$$[\mathbf{x}_{n+1} - \bar{f}(\mathbf{x}_n)] = [\bar{g}(\mathbf{x}_n)]\mathbf{u}_n. \quad (20)$$

(3) Next, the state is split as in (4) ($\mathbf{x}_n = [z'_n \bar{z}'_n]'$). The future value of the variables to be tracked (z_{n+1}) is replaced by the reference position ($z_{ref,n+1}$) assuming an approaching proportional to the error. In addition, an approach proportional to the error is also assumed for the required trajectory of the sacrificed variables ($\bar{z}_{ez,n+1}$), as can be seen in

$$\begin{bmatrix} \bar{g}_1(\mathbf{x}_n) \\ \bar{g}_2(\mathbf{x}_n) \end{bmatrix} \mathbf{u}_n = \begin{bmatrix} \frac{z_{ref,n+1} - z_n - k_1(z_{ref,n} - z_n)}{T} - \bar{f}_1(\mathbf{x}_n) \\ \frac{\bar{z}_{ez,n+1} - \bar{z}_n - k_2(\bar{z}_{ez,n} - \bar{z}_n)}{T} - \bar{f}_2(\mathbf{x}_n) \end{bmatrix}. \quad (21)$$

That is

$$\mathbf{A}_n \mathbf{u}_n = \mathbf{b}_n. \quad (22)$$

(4) The controller parameters are chosen such that $0 < k_1 < 1$ and $0 < k_2 < 1$ in order to force the tracking errors to tend to zero, see [24] for details.

(5) From (22), the sacrificed references $\bar{z}_{ez,n}$ are determined to ensure that \mathbf{b}_n is a linear combination of the column vectors of the matrix \mathbf{A}_n .

The next value of the sacrificed reference may be approximated by

$$\bar{z}_{ez,n+1} = \bar{z}_{ez,n} + (\bar{z}_{ez,n} - \bar{z}_{ez,n-1}). \quad (23)$$

(6) Equation (22) allows the calculation of the control actions, \mathbf{u}_n , making the tracking errors tend to zero at every sampling time. Then, the control action is obtained by solving (22) by means of $\mathbf{u}_n = \mathbf{A}_n^\dagger \cdot \mathbf{b}_n$.

4.1. Kinematic Controller. Again, the methodology is applied to design a discrete time controller for the mobile robot represented by (7).

Remark 2. For the discretized robot model, the values of $x(t)$, $y(t)$, $\theta(t)$, $V(t)$, and $W(t)$ at discrete time $t = nT$, where T is the sampling period, and $n \in \{0, 1, 2, \dots\}$ will be denoted as x_n , y_n , θ_n , V_n , and W_n , respectively.

According to Section 2, the steps to design linear algebra discrete controllers are

where the state is n -dimensional.

- (i) First, the system (7) is approximated using a numerical method. Initially, the simplest Euler method approximation is used:

$$\begin{aligned} x_{n+1} &= x_n + TV_n \cos \theta_n, \\ y_{n+1} &= y_n + TV_n \sin \theta_n, \\ \theta_{n+1} &= \theta_n + TW_n. \end{aligned} \quad (24)$$

- (ii) Now, the control problem is formulated as solving a system of linear equations ($\mathbf{A}\mathbf{u} = \mathbf{b}$).

$$\underbrace{\begin{bmatrix} \cos \theta_n & 0 \\ \sin \theta_n & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} V_n \\ W_n \end{bmatrix}}_{\mathbf{u}} = \underbrace{\begin{bmatrix} \frac{x_{n+1} - x_n}{T} \\ \frac{y_{n+1} - y_n}{T} \\ \frac{\theta_{n+1} - \theta_n}{T} \end{bmatrix}}_{\mathbf{b}} = \underbrace{\begin{bmatrix} \frac{\Delta x_n}{T} \\ \frac{\Delta y_n}{T} \\ \frac{\Delta \theta_n}{T} \end{bmatrix}}_{\mathbf{b}}. \quad (25)$$

- (iii) Next, the robot position (x_{n+1}, y_{n+1}) is replaced by the reference position $(x_{\text{ref}}, y_{\text{ref}})$ assuming an approaching proportional to the error, and the same for the desired trajectory of θ to be determined, as can be seen in

$$\begin{bmatrix} \cos \theta_{ez,n} & 0 \\ \sin \theta_{ez,n} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_n \\ W_n \end{bmatrix} = \begin{bmatrix} \frac{\Delta x_{\text{ref},n}}{T} \\ \frac{\Delta y_{\text{ref},n}}{T} \\ \frac{\Delta \theta_{ez,n}}{T} \end{bmatrix} \quad (26)$$

$$= \frac{1}{T} \begin{bmatrix} x_{\text{ref},n+1} - k_x(x_{\text{ref},n} - x_n) - x_n \\ y_{\text{ref},n+1} - k_y(y_{\text{ref},n} - y_n) - y_n \\ \theta_{ez,n+1} - k_\theta(\theta_{ez,n} - \theta_n) - \theta_n \end{bmatrix}.$$

- (iv) The controller parameters are chosen such that $0 < k_x < 1$, $0 < k_y < 1$, and $0 < k_\theta < 1$ in order to force the tracking errors to tend to zero.
- (v) The system (26) must have the exact solution. Then, the vector \mathbf{b} must be a linear combination of the column vectors of matrix \mathbf{A} [18]. So, the orientation $\theta_{ez,n}$ is computed according to

$$\tan \theta_{ez,n} = \frac{\Delta y_{\text{ref},n}}{\Delta x_{\text{ref},n}}. \quad (27)$$

- (vi) Equation (26) allows the calculation of the control actions, V_n and W_n , making the tracking errors tend to zero at every sampling time. Then, the control

action is obtained by solving (26) by means of least squares, leading to

$$\begin{bmatrix} V_n \\ W_n \end{bmatrix} = \begin{bmatrix} \frac{\Delta x_{\text{ref},n}}{T} \cos \theta_{ez,n} + \frac{\Delta y_{\text{ref},n}}{T} \sin \theta_{ez,n} \\ \frac{\Delta \theta_{ez,n}}{T} \end{bmatrix}. \quad (28)$$

Remark 3. The value of $\theta_{ez,n+1}$ is required to compute W_n but equation (27) allows to calculate $\theta_{ez,n}$. However, $\theta_{ez,n+1}$ can be estimated using the Taylor's formula (as suggested in (23))

$$\theta_{ez,n+1} = \theta_{ez,n} + \frac{d\theta_{ez}}{dt}T + \frac{d^2\theta_{ez}}{dt^2} \frac{T^2}{2} + \dots + \text{Comp}, \quad (29)$$

where Comp is the complementary term [25]. Then, if the sampling time is small, $\theta_{ez,n+1}$ can be approximated as follows:

$$\begin{aligned} \theta_{ez,n+1} &\approx \theta_{ez,n}, \\ \theta_{ez,n+1} &\approx \theta_{ez,n} + \frac{d\theta_{ez}}{dt}T \approx 2\theta_{ez,n} - \theta_{ez,n-1}, \\ \theta_{ez,n+1} &\approx \theta_{ez,n} + \frac{d\theta_{ez}}{dt}T + \frac{d^2\theta_{ez}}{dt^2} \frac{T^2}{2} \approx \dots \approx \theta_{ez,n} \\ &\quad + \theta_{ez,n} - \theta_{ez,n-1} + \frac{\theta_{ez,n} - 2\theta_{ez,n-1} + \theta_{ez,n-2}}{2}. \end{aligned} \quad (30)$$

In general, if the sampling period is small enough, the simplest approximation $\theta_{ez,n+1} \approx \theta_{ez,n}$ provides excellent results.

4.2. Controller Design Refinements. A different numerical method to approximate the mobile robot model can be used. For instance, if the trapezoidal approximation is adopted, a new control law is obtained by following the same steps described above.

- (i) The system (7) is now approximated as,

$$\begin{aligned} x_{n+1} &= x_n + \left(\frac{T}{2}\right)(V_n \cos \theta_n + V_{n+1} \cos \theta_{n+1}), \\ y_{n+1} &= y_n + \left(\frac{T}{2}\right)(V_n \sin \theta_n + V_{n+1} \sin \theta_{n+1}), \\ \theta_{n+1} &= \theta_n + \left(\frac{T}{2}\right)(W_n + W_{n+1}), \end{aligned} \quad (31)$$

and the aim is to calculate V_{n+1} and W_{n+1} such that the robot reaches and follows a preestablished trajectory by knowing the position and orientation of

the robot (x_n, y_n, θ_n) and its current velocities (V_n, W_n) .

- (ii) Again, the control problem is reformulated as a system of linear equations problem ($\mathbf{A}\mathbf{u} = \mathbf{b}$):

$$\underbrace{\begin{bmatrix} \cos \theta_{n+1} & 0 \\ \sin \theta_{n+1} & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} V_{n+1} \\ W_{n+1} \end{bmatrix}}_{\mathbf{u}} = \underbrace{\begin{bmatrix} \frac{2}{T}(x_{n+1} - x_n) - V_n \cos \theta_n \\ \frac{2}{T}(y_{n+1} - y_n) - V_n \sin \theta_n \\ \frac{2}{T}(\theta_{n+1} - \theta_n) - W_n \end{bmatrix}}_{\mathbf{b}}, \quad (32)$$

instead of (25).

- (iii) Next, the desired trajectory of the remaining state variable, $\theta_{ez,n+1}$, should be determined. Similar to (26), it follows

$$\underbrace{\begin{bmatrix} \cos \theta_{ez,n+1} & 0 \\ \sin \theta_{ez,n+1} & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} V_{n+1} \\ W_{n+1} \end{bmatrix}}_{\mathbf{u}} = \underbrace{\begin{bmatrix} \frac{\Delta x_{ref,n}}{T} \\ \frac{\Delta y_{ref,n}}{T} \\ \frac{\Delta \theta_{ez,n}}{T} \end{bmatrix}}_{\mathbf{b}}$$

$$= \underbrace{\begin{bmatrix} \frac{2}{T} [x_{ref,n+1} - k_x(x_{ref,n} - x_n) - x_n] - V_n \cos \theta_n \\ \frac{2}{T} [y_{ref,n+1} - k_y(y_{ref,n} - y_n) - y_n] - V_n \sin \theta_n \\ \frac{2}{T} [\theta_{ez,n+1} - k_\theta(\theta_{ez,n} - \theta_n) - \theta_n] - W_n \end{bmatrix}}_{\mathbf{b}}. \quad (33)$$

- (iv) Find the conditions in order that the system (33) has exact solution,

$$\tan \theta_{ez,n+1} = \frac{\Delta y_{ref,n}}{\Delta x_{ref,n}}. \quad (34)$$

- (v) The value of $\theta_{ez,n+1}$ is calculated from (34), and it represents the value that must be taken by the robot orientation in order that the tracking errors tend to zero.
- (vi) Finally, the control actions are calculated solving system (33) by least squares:

$$V_{n+1} = \frac{2\Delta x_{ref,n}}{T} \cos \theta_{ez,n+1} + \frac{2\Delta y_{ref,n}}{T} \sin \theta_{ez,n+1} - V_n \cos(\theta_{ez,n+1} - \theta_n), \quad (35)$$

$$W_{n+1} = \frac{2\Delta \theta_{ez,n}}{T} - W_n.$$

From (35), it can be seen that the control action now depends on the linear and angular velocity of the mobile robot, which leads to improved performance of the controller.

5. Controller Design under Uncertainties

The methodology for controller design based on linear algebra presented in Section 2 can be extended to deal with the problem of trajectory tracking under uncertainties [26].

Let us consider additive uncertainties in such a way that (19) is now written as

$$\mathbf{x}_{n+1} = \bar{\mathbf{h}}(\mathbf{x}_n, \mathbf{u}_n) + \mathbf{E}_n. \quad (36)$$

Notice that the additive uncertainty can be used to model perturbed systems as well as a wide class of model mismatches. Consider a real plant $\mathbf{x}_{n+1} = \mathbf{h}(\mathbf{x}_n, \mathbf{u}_n)$, the additive uncertainty can be expressed by $\mathbf{E}_n = \mathbf{h}(\mathbf{x}_n, \mathbf{u}_n) - \bar{\mathbf{h}}(\mathbf{x}_n, \mathbf{u}_n)$, where $\bar{\mathbf{h}}(\mathbf{x}_n, \mathbf{u}_n)$ is the discrete time nonlinear model of the system. Note that if, as it will be assumed, \mathbf{s} and \mathbf{u} are bounded and \mathbf{h} is Lipschitz, then \mathbf{E}_n can be modeled as a bounded uncertainty ([27]).

In the simplest case, the uncertainty will be constant. Thus, the procedure previously mentioned is modified to incorporate new terms considering the error integral.

In particular, in the Step 3 outlined in Section 2 and considering a DT control, the tracked state variables are replaced by the reference variables assuming a smooth approaching including an error integral. Thus, these variables are replaced by $z_{n+1} = z_{ref,n+1} - k_z(z_{ref,n} - z_n) + \mathbf{K}_{z,1} \mathbf{U}_{z,n}$. In this approaching, z_n is a tracked state variable at instant n , $\mathbf{U}_{z,1}$ are integrative variables, k_z and $\mathbf{K}_{z,1}$ are controller parameters associated to the variable z , and $z_{ref,n}$ is its desired value in DT at time instant n .

The integrative variables are defined by

$$\mathbf{U}_{z,n+1} = \mathbf{U}_{z,n} + \int_{nT}^{(n+1)T} e_z(t) dt \cong \mathbf{U}_{z,n} + T e_{z,n}, \quad (37)$$

where $e_z(t)$ is as defined in (10). The goal of the integrative terms is to ensure a null steady-state error, if there is a first-order model mismatch.

If the uncertainty dynamics (\mathbf{E}_n) is more complicated but it can be modeled by means of a polynomial term, additional integral actions can be added to cancel its steady-state effect. The rest of the steps are the same.

Now, the procedure is applied to the mobile robot kinematic model (7), where an additional uncertainty is introduced.

- (i) First, the system is discretized. In this case, the used approximation is based on a linear interpolation, assuming constant the control actions in the sampling interval:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \\ \theta_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \\ \theta_n \end{bmatrix} + \begin{bmatrix} \frac{1}{W_n} (\sin \theta_{n+1} - \sin \theta_n) & 0 \\ \frac{1}{W_n} (\cos \theta_n - \cos \theta_{n+1}) & 0 \\ 0 & T \end{bmatrix} \cdot \begin{bmatrix} V_n \\ W_n \end{bmatrix} + \mathbf{E}_n. \quad (38)$$

- (ii) Second, the control problem is formulated as solving a system of linear equations, without uncertainties:

$$\begin{bmatrix} \frac{1}{W_n} (\sin \theta_{n+1} - \sin \theta_n) & 0 \\ \frac{1}{W_n} (\cos \theta_n - \cos \theta_{n+1}) & 0 \\ 0 & T \end{bmatrix} \begin{bmatrix} V_n \\ W_n \end{bmatrix} = \begin{bmatrix} x_{n+1} - x_n \\ y_{n+1} - y_n \\ \theta_{n+1} - \theta_n \end{bmatrix}. \quad (39)$$

- (iii) Now, the tracked state variables required are replaced by the reference variables assuming a smooth approaching that, in this case, is assumed to be proportional to the error and the error integral in order to cope with the model constant uncertainties. A similar approaching is assumed for the sacrificed variable:

$$\begin{bmatrix} \frac{1}{W_n} (\sin \theta_{n+1} - \sin \theta_n) & 0 \\ \frac{1}{W_n} (\cos \theta_n - \cos \theta_{n+1}) & 0 \\ 0 & T \end{bmatrix} \begin{bmatrix} V_n \\ W_n \end{bmatrix} = \begin{bmatrix} x_{\text{ref},n+1} - x_n - k_x(x_{\text{ref},n} - x_n) + \mathbf{K}_{x,1} \mathbf{U}_{x,1,n+1} \\ y_{\text{ref},n+1} - y_n - k_y(y_{\text{ref},n} - y_n) + \mathbf{K}_{y,1} \mathbf{U}_{y,1,n+1} \\ \theta_{ez,n+1} - \theta_n - k_\theta(\theta_{ez,n} - \theta_n) + \mathbf{K}_{\theta,1} \mathbf{U}_{\theta,1,n+1} \end{bmatrix}. \quad (40)$$

- (iv) Next, conditions such that system (40) has exact solution and the errors tend to zero are defined:

$$\theta_{ez,n+1} = a \tan \frac{\sin \theta_{ez,n+1}}{\cos \theta_{ez,n+1}} = \frac{-(W_n/V_n)\Delta y_n + \cos \theta_n}{(W_n/V_n)\Delta x_n + \sin \theta_n}, \quad (41)$$

where

$$\begin{aligned} \Delta x_n &= x_{\text{ref},n+1} - x_n - k_x(x_{\text{ref},n} - x_n) + \mathbf{K}_{x,1} \mathbf{U}_{x,1,n+1}, \\ \Delta y_n &= y_{\text{ref},n+1} - y_n - k_y(y_{\text{ref},n} - y_n) + \mathbf{K}_{y,1} \mathbf{U}_{y,1,n+1}, \\ \Delta \theta_n &= \theta_{ez,n+1} - \theta_n - k_\theta(\theta_{ez,n} - \theta_n) + \mathbf{K}_{\theta,1} \mathbf{U}_{\theta,1,n+1}. \end{aligned} \quad (42)$$

- (v) The orientation θ in (40) ensures that the system (39) has exact solution and is computed according to

$$\theta_{ez,n+1} = a \tan 2 \left(\frac{W_n}{V_n} \Delta x_n + \sin \theta_n, \frac{W_n}{V_n} \Delta y_n + \cos \theta_n \right). \quad (43)$$

- (vi) Finally, the control actions are obtained by solving the system (39) by means of $\mathbf{u} = \mathbf{A}^+ \mathbf{b}$:

$$\begin{aligned} W_n &= \frac{\Delta \theta_n}{T}, \\ V_n &= \frac{\Delta x_n^2 + \Delta y_n^2}{2(\Delta y_n \cos \theta_n - \Delta x_n \sin \theta_n)} W_n. \end{aligned} \quad (44)$$

6. Experimental and Simulation Results

In this section, several tests are carried out in order to evaluate the controller's performance. The experiments were carried out by using a PIONEER 3AT mobile robot. This robot (Figure 1) includes an estimation system based on an odometric-based positioning system. Measurement updating through external sensors is necessary. This problem is separated from the strategy of trajectory tracking and it is not considered in this paper [28, 29]. The PIONEER 3AT has a PID speed controller used to maintain the velocity of the mobile robot at the desired value. The structure of the implemented control system is given in Figure 2, where the trajectory-tracking controller implements equations described in (13), (18), (28), (35), or (44).

In order to compare the controller's performance, the tracking errors are considered. An idea widely used in the literature is to consider the cost incurred by the error [26]. Then, the cost function attached to the trajectory Φ can be represented by the combination of the quadratic error in the x -coordinate and the y -coordinate as shown in the following equation:

$$C^\Phi = \sum_{i=0}^{\#\Phi} \frac{1}{2} \left((x_{\text{ref},i} - x_i)^2 + (y_{\text{ref},i} - y_i)^2 \right). \quad (45)$$

In the results presentation, the controller based on kinematic model developed in continuous time will be called C1 (Section 3); the controller developed in discrete time and Euler approximation will be called C2 (Section 4.1); the controller based on trapezoidal approximation will be noted as C3 (Section 4.2); the controller based on



FIGURE 1: The pionner 3AT mobile robot.

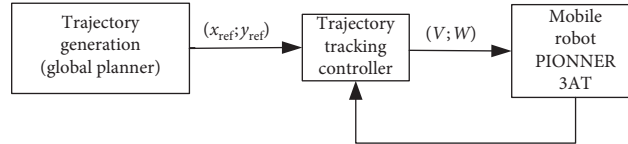


FIGURE 2: Structure of control system.

dynamic model and Euler approximation will be called C4 (Section 3.2); and the controller developed considering uncertainties (Section 5) will be called C5. The controller and system parameters used for the experimentation are obtained from [8, 16, 21, 24, 26], respectively.

Remark 4. In this work, only a few results are shown. However, exhaustive tests of the controllers against different trajectories, disturbance in the control actions, as well as parametric uncertainty, can be found in [8, 16, 21, 24, 26].

To evaluate the performance of the designed controllers (C1, C2, C3, C4, C5), they are compared with controllers previously published by other authors in [2, 27, 30] that will be named (C6, C7, C8), respectively (see Table 1). Table 1 summarizes the formulation of the controllers implemented for comparison (C6, C7, and C8). The designing details of these controllers can be found in their respective references ([2, 27, 30]).

The simulation experiments have been carried out in two different scenarios. First, all controllers are evaluated trying to follow an eight-shaped trajectory. It is well known that following an eight-way trajectory is a complicated task due to its geometrical shape and directions changes. Then, the controller's performance is evaluated following a sinusoidal trajectory in presence of disturbances in the control actions.

6.1. Eight-Shaped Trajectory Tracking. In this subsection, the robot should follow an eight-shaped trajectory following the guidelines previously published in [16]. The initial position of the robot is at the system origin and the trajectory begins in the position $(x_{\text{ref}}(0), y_{\text{ref}}(0)) = (1 \text{ m}, -1 \text{ m})$. The sampling time used for the discrete controllers is $T = 0.1 \text{ sec}$.

Figure 3 shows the results of the implementation. By inspection of this figure, all controllers reach and follow the reference trajectory. However, C5 shows the lowest

TABLE 1: Trajectory tracking controllers.

Controller	Formulation
[2]	$C = \begin{bmatrix} \cos \psi & -a \sin \psi \\ \sin \psi & a \cos \psi \end{bmatrix}$ $\begin{bmatrix} v \\ w \end{bmatrix} = C^{-1} \left(\begin{bmatrix} \dot{x}_{\text{ref}} \\ \dot{y}_{\text{ref}} \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \end{bmatrix} \right)$
[27]	$h_x = k_p e_x + \dot{x}$ $h_y = k_p e_y + \dot{y}$ $v = h_x \cos \psi + h_y \sin \psi$ $w = k_\psi (\psi_{\text{ref}} - \psi) + \dot{\psi}_{\text{ref}}$
[30]	$v = u_{\text{ref}} \cos \psi_e + k_x x_e$ $w = r_{\text{ref}} + u_{\text{ref}} (k_y y_e + k_\psi \sin \psi_e)$

error cost (as defined in (45)) when compared to the rest of the controllers (Figures 4 and 5). Good results were expected in the tracking costs because the controller C5 is designed taking into account modeling errors. The performance of each implemented controller can be analyzed by inspection of Figures 6 and 7, where the tracking errors in x -coordinate and y -coordinate are shown, respectively. Figure 5 shows that the controller C5 improves the tracking cost by up to 25% when compared with C6, C7, and C8.

6.2. Experimental Results with Disturbance in the Control Actions. Now a second task is carried out in order to test the limits of our formulation. Thus, the controller's performance when disturbance in the control actions are introduced is evaluated. The disturbance is introduced according to the diagram in Figure 8.

The disturbance in the linear velocity at time n is represented by $V = V + \delta$ and disturbance in the angular velocity at time n is represented by $W = W + \delta$, where δ represents the disturbance introduced in each velocity and is given in Figure 9.

For this task, a trajectory reference is generated with a constant linear velocity $V = 0.5 \text{ m/s}$. The initial position of

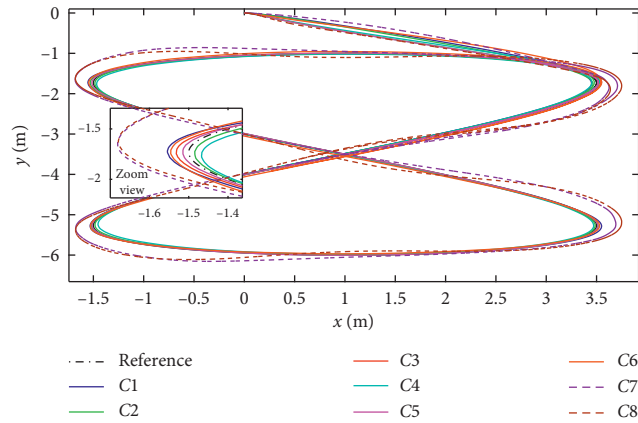


FIGURE 3: Robot position for each controller.

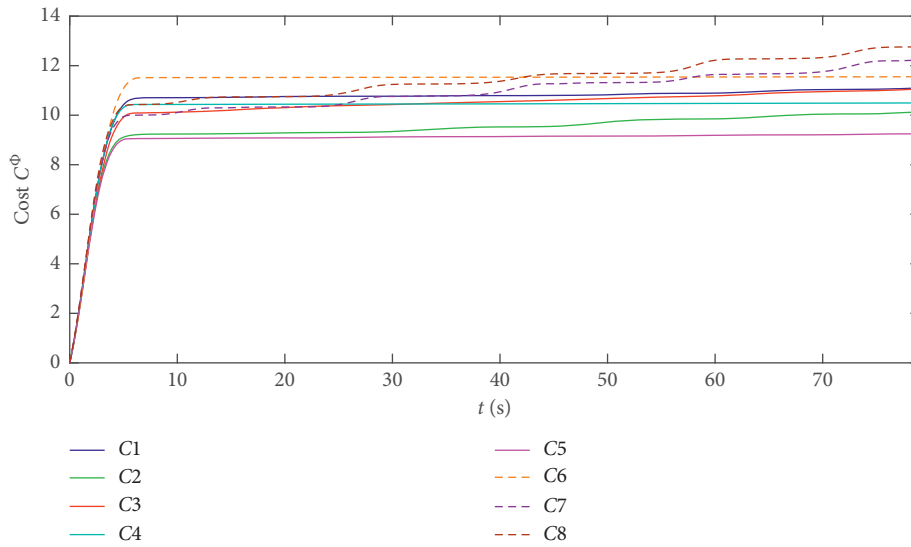


FIGURE 4: Tracking cost of controllers.

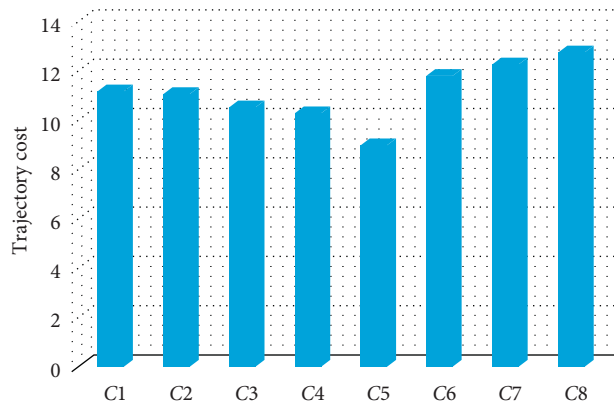


FIGURE 5: The pionner 3AT mobile robot.

the robot is at the system origin, and the trajectory begins in the position $(x_{ref}(0), y_{ref}(0)) = (1\text{ m}, 1\text{ m})$. The sample time used is $T = 0.1\text{ s}$.

The results of the implementation with disturbance in the control action are shown in Figure 10. Figure 10 shows the trajectory reference and the position of the mobile

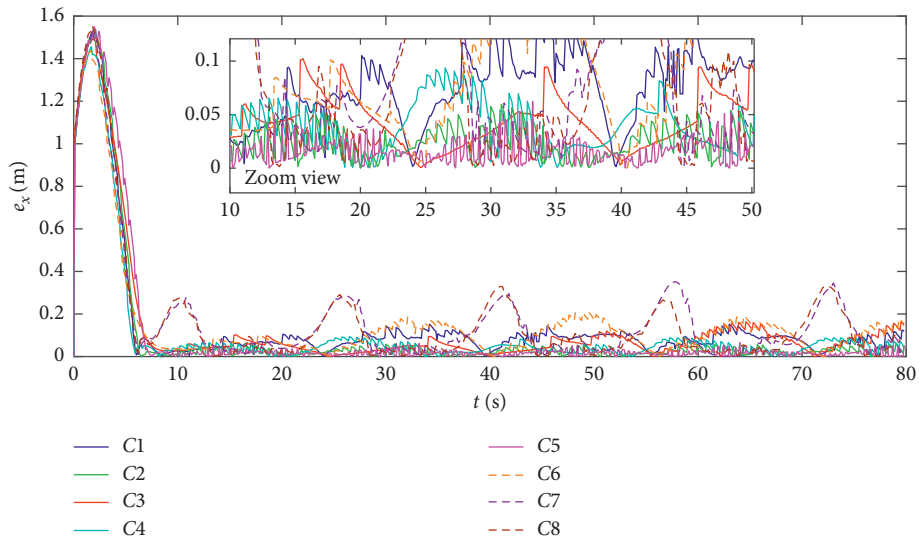


FIGURE 6: Error in x variable for each controller.

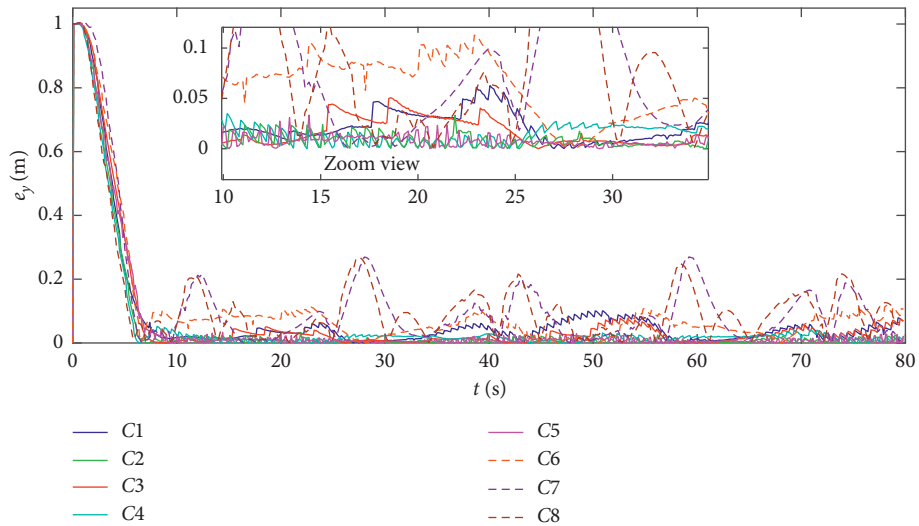


FIGURE 7: Error in y variable for each controller.

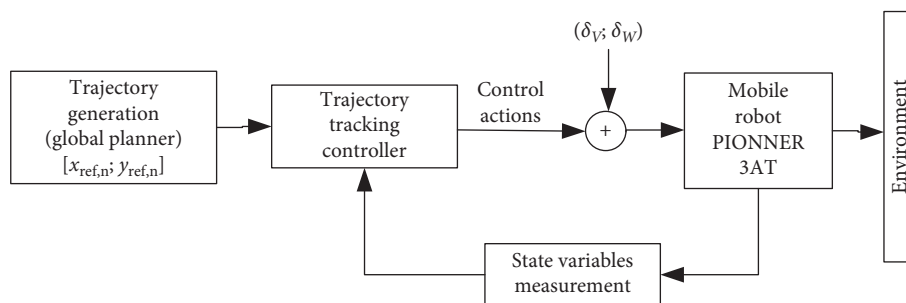


FIGURE 8: Architecture of the trajectory-tracking controllers.

robot for each controller. As it can be seen, the controllers C4 and C5 present the lowest tracking error. Figures 11 and 12 show the tracking errors in x -coordinate and y -

coordinate, separately. From Figure 13, it is observed that the controllers designed with the proposed methodology (C1, C2, C3, C4, and C5) exhibit better performance than

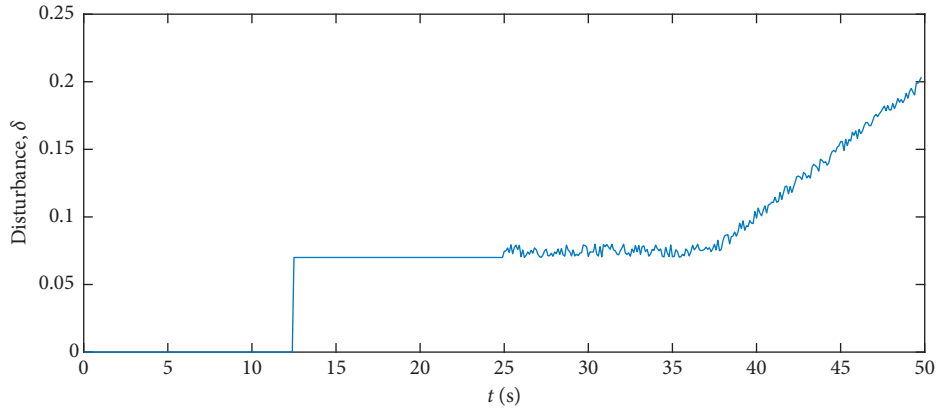


FIGURE 9: Disturbance introduced in the robot velocities.

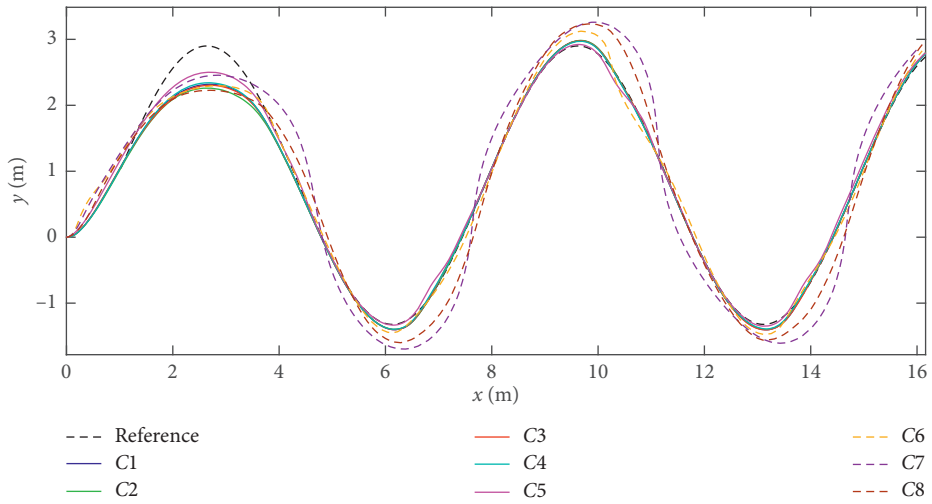


FIGURE 10: Position of the mobile robot for each experimentation.

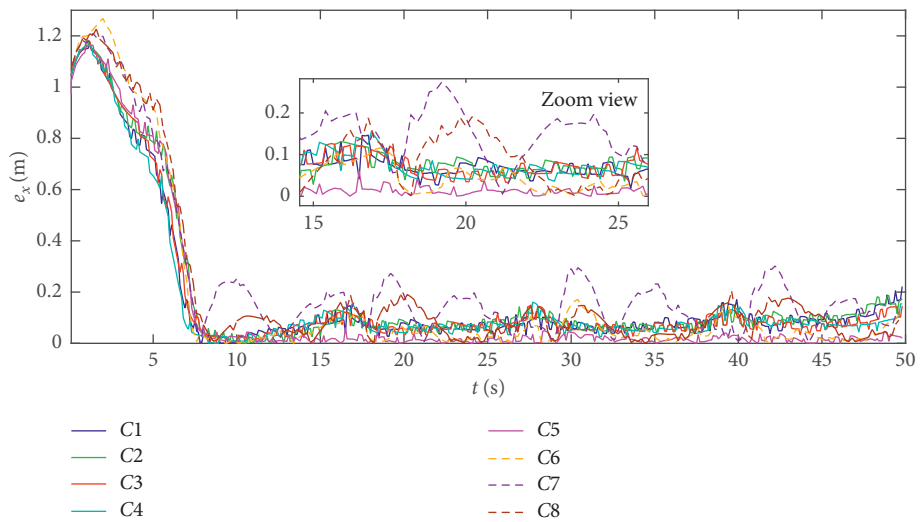


FIGURE 11: Tracking error in x variable: e_x .

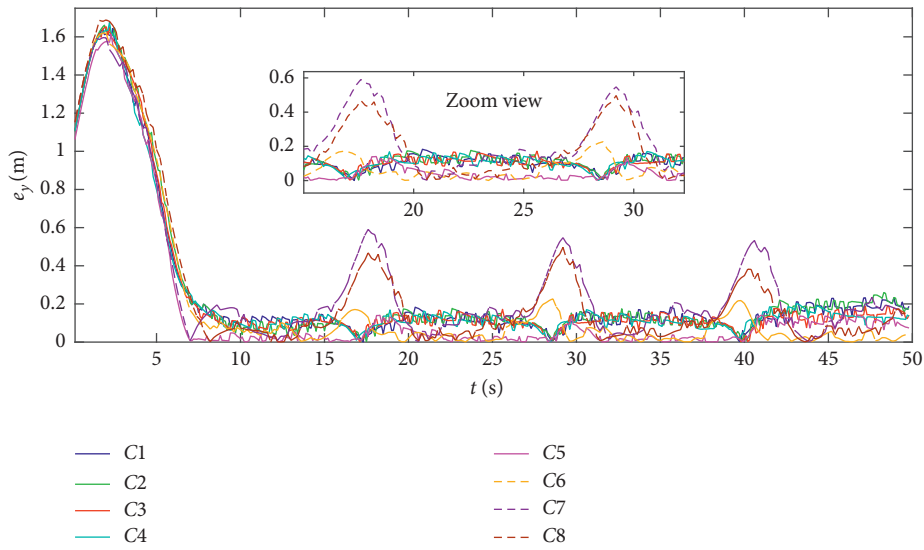


FIGURE 12: Tracking error in y variable: e_y .

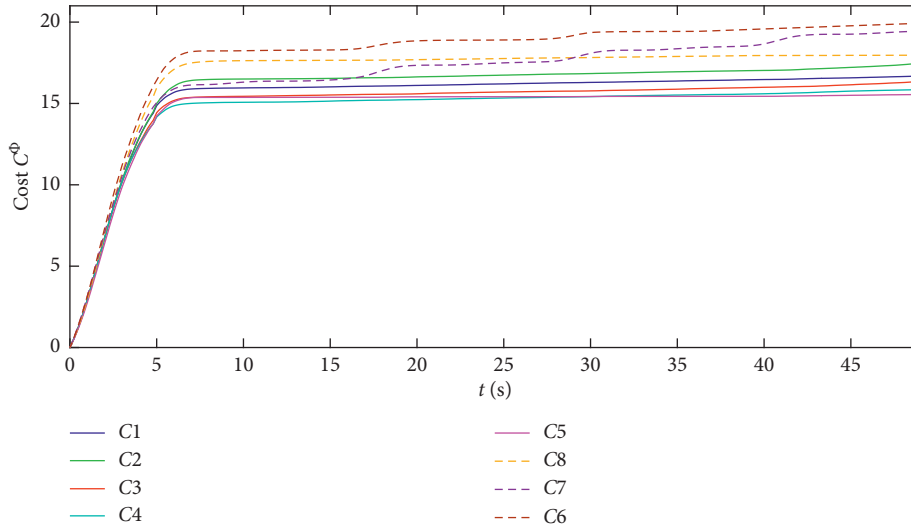


FIGURE 13: Tracking cost obtained in the experimentation.

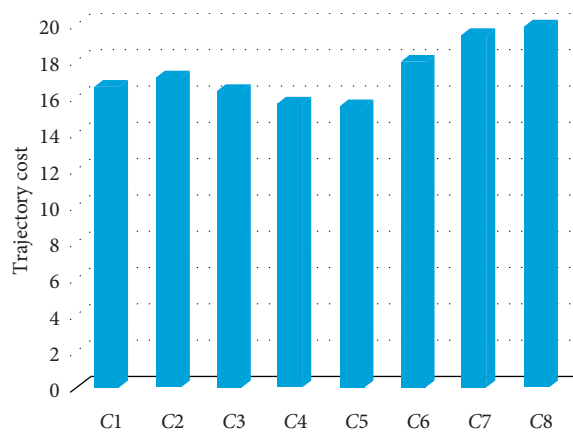


FIGURE 14: The pionner 3AT mobile robot.

controllers C_6 , C_7 , and C_8 . Furthermore, the trajectory cost is shown in Figure 14, showing that C_5 has the lowest tracking cost.

7. Conclusion

A well-known and accepted trajectory-tracking methodology based on linear algebra was formalized. Two main limitations are assumed: the state-space process model is affined in the control and the state variables are measured. The design steps are summarized and illustrated by some examples in both continuous and discrete time. Furthermore, the procedure to deal with model uncertainties is presented. The main characteristics of the proposed controllers are that they present low calculation complexity and that they can be easily implemented. This is because for the calculation of the control action, it uses the system model and not complex coordinate transformations.

Experimental results in a mobile robot illustrate the good performance of the designed controllers. Furthermore, the experimental results considering disturbance in the control actions illustrate that our methodology is effective and robust to external disturbances. The simulations results are satisfactory and suggest that the methodology can be applied to nonlinear multivariable systems of different nature.

Currently, the inclusion and design of observers to evaluate the full state of the plant, if it is not accessible, are under consideration.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially funded by the Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, National Council for Scientific Research), Argentina. The authors thank the Institute of Chemical Engineering of the National University of San Juan, Argentine, the Escuela Politécnica Nacional del Ecuador (Facultad de Ingeniería Eléctrica y Electrónica), and Instituto de Automática e Informática Industrial, Universidad Politécnica de Valencia.

References

- [1] J. M. Toibero, F. Roberti, R. Carelli, and P. Fiorini, "Switching control approach for stable navigation of mobile robots in unknown environments," *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 3, pp. 558–568, 2011.
- [2] C. Z. Resende, R. Carelli, and M. Sarcinelli-Filho, "A nonlinear trajectory tracking controller for mobile robots with velocity limitation via fuzzy gains," *Control Engineering Practice*, vol. 21, no. 10, pp. 1302–1309, 2013.
- [3] K. Wang, "Near-optimal tracking control of a nonholonomic mobile robot with uncertainties," *International Journal of Advanced Robotic Systems*, vol. 9, no. 3, 66 pages, 2012.
- [4] J. Yuan, F. Sun, and Y. Huang, "Trajectory generation and tracking control for double-steering tractor trailer mobile robots with on-axle hitching," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7665–7677, 2015.
- [5] D. Zhang and B. Wei, "A review on model reference adaptive control of robotic manipulators," *Annual Reviews in Control*, vol. 43, pp. 188–198, 2017.
- [6] P. K. Menon and S. G. Park, "Dynamics and control technologies in air traffic management," *Annual Reviews in Control*, vol. 42, pp. 271–284, 2016.
- [7] R. M. Murray, *Optimization-Based Control*, California Institute of Technology, Pasadena, CA, USA, 2009.
- [8] G. Scaglia, L. Q. Montoya, V. Mut, and F. Di Sciascio, "Numerical methods based controller design for mobile robots," *Robotica*, vol. 27, no. 2, pp. 269–279, 2009.
- [9] M. E. Serrano, G. J. E. Scaglia, S. A. Godoy, V. Mut, and O. A. Ortiz, "Trajectory tracking of underactuated surface vessels: a linear algebra approach," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 3, pp. 1103–1111, 2014.
- [10] C. Rosales, D. Gandolfo, G. Scaglia, M. Jordan, and R. Carelli, "Trajectory tracking of a mini four-rotor helicopter in dynamic environments—a linear algebra approach," *Robotica*, vol. 33, no. 8, pp. 1638–1652, 2014.
- [11] V. H. Andaluz, E. R. Sásig, W. D. Chicaiza, and P. M. Velasco, "Linear algebra applied to kinematic control of mobile manipulators," in *Proceedings of the International Conference On Information Theoretic Security*, pp. 297–306, Springer, Hong Kong, China, November 2017.
- [12] F. A. Cheein, S. Blazic, and M. Torres-Torriti, "Computational approaches for improving the performance of path tracking controllers for mobile robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6495–6500, IEEE, Hamburg, Germany, September 2015.
- [13] N. H. Amer, H. Zamzuri, K. Hudha, and Z. A. Kadir, "Modelling and control strategies in path tracking control for autonomous ground vehicles: a review of state of the art and challenges," *Journal of Intelligent & Robotic Systems*, vol. 86, no. 2, pp. 225–254, 2017.
- [14] L. Pan, A. Song, G. Xu, H. Li, and B. Xu, "Novel dynamic interpolation strategy for upper-limb rehabilitation robot," *Journal of Rehabilitation Robotics*, vol. 1, pp. 19–27, 2013.
- [15] C. B. Low, "Experimental implementation of a novel trajectory tracking control design on a full-sized nonholonomic tracked mobile robot with low-level velocities control systems," in *Proceedings of the IEEE Conference on Control Applications (CCA)*, pp. 1318–1323, IEEE, Juan Les Antibes, France, October 2014.
- [16] M. E. Serrano, S. A. Godoy, L. Quintero, and G. J. Scaglia, "Interpolation based controller for trajectory tracking in mobile robots," *Journal of Intelligent & Robotic Systems*, vol. 86, no. 3-4, pp. 569–581, 2017.
- [17] A. Rosales, G. Scaglia, V. A. Mut, and F. Di Sciascio, "Trajectory tracking of mobile robots in dynamic environments—a linear algebra approach," *Robotica*, vol. 27, no. 7, pp. 981–997, 2009.
- [18] G. Williams, *Linear Algebra with Applications*, Jones & Bartlett Publishers, Burlington, MA, USA, 2012.
- [19] P. Paniagua-Contro, E. G. Hernandez-Martinez, O. González-Medina et al., "Extension of leader-follower behaviours for wheeled mobile robots in multirobot coordination," *Mathematical Problems in Engineering*, vol. 2019, Article ID 4957259, 16 pages, 2019.
- [20] F. G. Rossomando, C. Soria, and R. Carelli, "Adaptive neural dynamic compensator for mobile robots in trajectory tracking

- control,” *IEEE Latin America Transactions*, vol. 9, no. 5, pp. 593–602, 2011.
- [21] A. Rosales, G. Scaglia, V. Mut, and F. Di Sciascio, “Formation control and trajectory tracking of mobile robotic systems—a linear algebra approach,” *Robotica*, vol. 29, no. 3, pp. 335–349, 2011.
- [22] S. Blažič, “A novel trajectory-tracking control law for wheeled mobile robots,” *Robotics and Autonomous Systems*, vol. 59, no. 11, pp. 1001–1007, 2011.
- [23] C. De La Cruz and R. Carelli, “Dynamic model based formation control and obstacle avoidance of multi-robot systems,” *Robotica*, vol. 26, no. 3, pp. 345–356, 2008.
- [24] G. Scaglia, A. Rosales, L. Quintero, V. Mut, and R. Agarwal, “A linear-interpolation-based controller design for trajectory tracking of mobile robots,” *Control Engineering Practice*, vol. 18, no. 3, pp. 318–329, 2010.
- [25] B. F. Hildebrand, *Introduction to Numerical Analysis*, Dover Publications, Inc., New York, NY, USA, 2nd edition, 1987.
- [26] G. Scaglia, E. Serrano, A. Rosales, and P. Albertos, “Linear interpolation based controller design for trajectory tracking under uncertainties: application to mobile robots,” *Control Engineering Practice*, vol. 45, pp. 123–132, 2015.
- [27] M. Michałek and K. Kozłowski, “Feedback control framework for car-like robots using the unicycle controllers,” *Robotica*, vol. 30, no. 4, pp. 517–535, 2012.
- [28] J. E. Normey-Rico, J. Gómez-Ortega, and E. F. Camacho, “A Smith-predictor-based generalised predictive controller for mobile robot path-tracking,” *Control Engineering Practice*, vol. 7, no. 6, pp. 729–740, 1999.
- [29] J. E. Normey-Rico, I. Alcalá, J. Gómez-Ortega, and E. F. Camacho, “Mobile robot path tracking using a robust PID controller,” *Control Engineering Practice*, vol. 9, no. 11, pp. 1209–1214, 2001.
- [30] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, “A stable tracking control method for an autonomous mobile robot,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 384–389, IEEE, Cincinnati, OH, USA, May 1990.

Copyright © 2019 Gustavo Scaglia et al. This is an open access article distributed under the Creative Commons Attribution License (the “License”), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License. <http://creativecommons.org/licenses/by/4.0/>