Control Engineering Practice 45 (2015) 123–132

Contents lists available at ScienceDirect

Control Engineering Practice

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

Linear interpolation based controller design for trajectory tracking under uncertainties: Application to mobile robots

Gustavo Scaglia [a,b,*], Emanuel Serrano [a], Andrés Rosales [b], Pedro Albertos [b,c]

[a] Instituto de Ingeniería Quím ica, Universidad Nacional de San Juan, Av. Libertador San Martín Oeste 1109, San Juan J5400ARL, Argentina
[b] Departamento de Automatización y Control Industrial, Facultad de Ingeniería Eléctrica y Electrónica, Escuela Politécnica Nacional, Quito, Ecuador
[c] Departamento de Ingeniería de Sistemas y Automática, Universidad Politécnica de Valencia, España

ARTICLE INFO

Article history:
Received 9 April 2015
Received in revised form
31 August 2015
Accepted 20 September 2015

Keywords:
Tracking control
Difference equations
Uncertainties
Non-linear multivariable control
Mobile robot

ABSTRACT

The problem of trajectory tracking control in mobile robots under uncertainties is addressed in this paper. Following the results of mobile robots trajectory tracking reported in (Scaglia et al., 2010), the problem of model errors is focused and the zero convergence of tracking errors under polynomial uncertainties is demonstrated. A simple scheme is obtained, which can be easily implemented. Simulation and experimental results are presented and discussed, showing the good performance of the controller.

1. Introduction

This paper addresses the problem of trajectory tracking control in mobile robots under uncertainties. The use of trajectory tracking for mobile robots is justified in structured working spaces as well as in partially structured workspaces, where unexpected obstacles can be found during the navigation. In the first case, the reference trajectory can be set from a global trajectory planner. In the second case, the algorithms used to avoid obstacles usually re-plan the trajectory in order to avoid a collision generating a new reference trajectory from this point on. In general, the objective is to find the control actions that make the mobile robot to reach a Cartesian position (x_{ref}, y_{ref}) with a pre-established orientation θ at each sampling period. These combined actions result in tracking the desired trajectory of the mobile robot.

Several tracking controllers designed by using a linearized model have been reported. In Oriolo, Luca & Vendittelli (2002), a dynamic feedback linearization technique is used to control a mobile robot platform. Usually, the desired trajectory is obtained by using a reference virtual robot; therefore, all the kinematic constraints are implicitly considered by the desired trajectory. The control inputs are mostly obtained by a combination of feedforward inputs, calculated from the desired trajectory, and the feedback control law. In Klancar & Skrjanc (2007), a model-predictive trajectory-tracking control applied to a mobile robot is presented. Linearized tracking-error dynamics is used to predict the system future behavior and a control law is derived from a quadratic cost function penalizing the system tracking error and the control effort. In Blažič (2011), a kinematic model is proposed where the transformation between the robot posture and the system state is bijective. The authors show that the global asymptotic stability of the system is achieved if the reference velocities satisfy the condition of persistent excitation.

In Scaglia, Quintero, Mut & Di Sciascio (2009), a novel trajectory-tracking controller has been presented. The originality of this control approach is based on the application of linear algebra for trajectory tracking, where the control actions are obtained by solving a system of linear equations. The methodology developed for tracking the desired trajectory (x_{ref} , y_{ref}) is based on determining the trajectories of the remaining state variables, thus the tracking error tends to zero. These states are determined by analyzing the conditions so that the system of linear equations has exact solution. In order to achieve this objective, only two control variables are available: the linear velocity (V) and the angular velocity (W) of the robot, Fig. 1. This design technique has been applied successfully in several systems (Scaglia et al., 2009; Rosales, Scaglia, Mut & Di Sciascio, 2011; Rosales, Scaglia, Mut & Di Sciascio, 2009; Serrano, Scaglia, Godoy, Mut & Ortiz, 2013).

The authors in Scaglia, Rosales, Quintero, Mut & Agarwal (2010), present a novel linear interpolation based methodology to design control algorithms for trajectory tracking of mobile vehicles. In that work, it is assumed that the evolution of the system

* Corresponding author at: Instituto de Ingeniería Quím ica, Universidad Nacional de San Juan, Av. Libertador San Martín Oeste 1109, San Juan J5400ARL, Argentina.

http://dx.doi.org/10.1016/j.conengprac.2015.09.010
0967-0661/

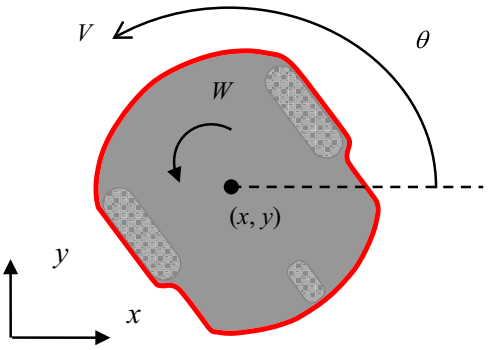

Fig. 1. Geometric description of the mobile robot.

can be approximated by a linear interpolation at each sampling time. A novel approach for trajectory tracking of a mobile-robots formation by using linear algebra theory and numerical methods is presented in Rosales et al. (2011). Using this strategy, the formation of mobile robots is able to change its configuration (shape and size) and follow different trajectories in a precise way, minimizing the tracking and formation errors. To face up to sudden velocity changes and to improve the performance of the system, Rosales et al. (2009) proposed working with the dynamic model of the mobile robot.

This paper provides a positive answer to the challenging problem of designing controllers for trajectory tracking in multivariable nonlinear systems with additive uncertainty. This problem has attracted the interest of many researchers and several solutions are reported in the literature (see for instance, Lee, Lin, Lim & Lee (2009), Farooq, Hasan, Hanif, Amer & Asad (2014), Wang (2012), Scaglia et al. (2009), and many others). The proposed controllers are rather complicated and require high computation cost.

The controller is derived by using the discrete state space model equations. This simple approach suggests that knowing the value of the desired state, a value for the control action forcing the system to move from its current state to the desired one can be computed. A system of linear equations needs to be solved to compute the control actions to carry the current outputs to the desired values. The main contribution of this work is the extension of the proposed methodology, based upon easily understandable concepts and without requiring complex calculations to attain the control signal, to design the tracking control of a mobile with uncertainties in the model. The methodology proposed here yields trajectory tracking controllers with low computational cost, small tracking error and low control effort. Additionally, our controller shows to be robust under disturbances in the control actions. Due to its mathematical formulation, our approach can also be implemented embedded (it does not compute higher order derivatives, exponentiation or complex trigonometric functions). Another contribution of this paper is the application of Monte Carlo (MC) based sampling experiment in the simulations. The controller parameters can be computed to minimize a cost index, here being determined by the Monte Carlo (MC) experiment, and the theoretical results are validated by simulations and experimentation.

It is noteworthy that due to the above mentioned characteristics, the computing power required to perform the mathematical operations is low. Thence it is possible to implement the algorithm in any controller with low computing capacity. Furthermore, the developed algorithm is easier to implement in a real system because the use of discrete equations allows direct adaptation to any computer system or programmable device running sequential instructions at a programmable clock speed. Thus, other than the simplicity of the controller, one great advantage of this approach is the use of discrete-time equations, simplifying its implementation on a computer system. The proof of the zero-convergence of the tracking error under uncertainty is another main contribution of this work.

The paper is organized as follows: in the next sections, some results from previous works are summarized to make this paper self-contained. In Section 2, the kinematical model of the mobile robot is presented, and the controller design methodology is shown in Section 3. In Section 4, the controller parameters are tuned by the MC experiment and theoretical results are validated with simulation results of the control algorithm. Three experimental results using a mobile robot Pioneer 3AT are presented in Section 5. Finally, the conclusions and some topics that will be addressed in future contributions are outlined in Section 6.

2. Kinematic model of the mobile robot and controller design

2.1. Kinematic model of the mobile robot

A simple nonlinear kinematic model for a mobile robot shown in Fig. 1, represented by (1), will be used

$$\begin{cases} \dot{x} = V\cos\theta \\ \dot{y} = V\sin\theta \\ \dot{\theta} = W \end{cases} \tag{1}$$

where, V: linear velocity of the mobile robot, W: angular velocity of the mobile robot, (x, y): Cartesian position, θ: mobile robot orientation. This model has been used in several recent papers such as Blažič (2011), Rosales et al., (2011), Rossomando, Soria & Carelli (2011) and Resende, Carelli & Sarcinelli-Filho (2013). Note that the dynamics of the mobile as well as those from the actuators are not initially considered. They will be taken into account later on as uncertainties in the model (1).

2.2. Controller design

The goal is to find the values of V and W so that the mobile robot may follow a pre-established trajectory (x_{ref}, y_{ref}) with a minimum error. The values of $x(t)$, $y(t)$, $\theta(t)$, $V(t)$ and $W(t)$ at discrete time $t = nT_s$, where T_s is the sampling time, and $n \in \{0, 1, 2, \cdots\}$ will be denoted as x_n, y_n, θ_n, V_n and W_n, respectively.

From (1), it follows,

$$\begin{cases} x_{n+1} = x_n + \int_{nT_s}^{(n+1)T_s} V\cos\theta dt \\ y_{n+1} = y_n + \int_{nT_s}^{(n+1)T_s} V\sin\theta dt \\ \theta_{n+1} = \theta_n + \int_{nT_s}^{(n+1)T_s} W dt \end{cases} \tag{2}$$

where the control actions V and W remain constant in the interval $nT_s \le t < (n+1)T_s$ and equal to V_n and W_n. Then, for $nT_s \le t < (n+1)T_s$,

$$\begin{cases} \theta(t) = \theta_n + W_n(t - nT_s) \\ \theta_{n+1} = \theta_n + W_n T_s \end{cases} \tag{3}$$

Then, from (2) and (3), it follows

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \\ \theta_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \\ \theta_n \end{bmatrix} + \begin{bmatrix} \frac{1}{W_n}\{\sin\theta_{n+1} - \sin\theta_n\} & 0 \\ \frac{1}{W_n}\{\cos\theta_n - \cos\theta_{n+1}\} & 0 \\ 0 & T_s \end{bmatrix} \begin{bmatrix} V_n \\ W_n \end{bmatrix} \tag{4}$$

Denote by $\mathbf{z}_n$ the state vector and $\Delta\mathbf{z}_n$ its increment, that is

$$\mathbf{z}_n = \begin{bmatrix} x_n \\ y_n \\ \theta_n \end{bmatrix}; \quad \Delta\mathbf{z}_n = \begin{bmatrix} \Delta x_n \\ \Delta y_n \\ \Delta\theta_n \end{bmatrix} = \begin{bmatrix} x_{n+1} - x_n \\ y_{n+1} - y_n \\ \theta_{n+1} - \theta_n \end{bmatrix}; \tag{5}$$

and

$$B_n(W_n, \theta_n, \theta_{n+1}) = \begin{bmatrix} \frac{1}{W_n}\{\sin\theta_{n+1} - \sin\theta_n\} & 0 \\ \frac{1}{W_n}\{\cos\theta_n - \cos\theta_{n+1}\} & 0 \\ 0 & T_s \end{bmatrix}; \quad u_n = \begin{bmatrix} V_n \\ W_n \end{bmatrix} \tag{6}$$

are the input matrix and the control action, respectively. Then (4) can be rewritten

$$z_{n+1} = z_n + B_n(W_n, \theta_n, \theta_{n+1})u_n \tag{7}$$

From (4), the control to move from (x_n, y_n) to (x_{n+1}, y_{n+1}) can be derived:

1. From the last row in (4), the control input to go from $\mathbf{z}_n$ to $\mathbf{z}_{n+1}$ will require

$$W_n = \frac{\theta_{n+1} - \theta_n}{T_s} = \frac{\Delta\theta_n}{T_s} \tag{8}$$

1. Note that the first two rows in (4) involve a unique variable V_n. So, in order to follow an exact trajectory, z_n to z_{n+1}, and being (x_{n+1}, y_{n+1}) given, the direction θ_{n+1} must be (Scaglia et al., 2010)

$$\theta_{ref,n+1} = a\tan\frac{\sin\theta_{ref,n+1}}{\cos\theta_{ref,n+1}} \tag{9}$$

where,

$$\sin\theta_{ref,n+1} = \frac{W_n}{V_n}\Delta x_n + \sin\theta_n$$

$$\cos\theta_{ref,n+1} = -\frac{W_n}{V_n}\Delta y_n + \cos\theta_n$$

The orientation $\theta_{ref,n+1}$ will be taken as the reference value for θ in the next period, and the ratio between the control actions is given by

$$\frac{W_n}{V_n} = \frac{2(\Delta y_n \cos\theta_n - \Delta x_n \sin\theta_n)}{\Delta x_n^2 + \Delta y_n^2} \tag{10}$$

The value of θ that satisfies (9) will be called $\theta_{ref,n+1}$ (this is the necessary orientation to force the tracking error to zero). This orientation ensures that the robot tends to the required trajectory and it will be used as the reference for the orientation. Now, let us assume that the goal is to track a trajectory given by $\mathbf{z}_{ref}$, where (x_{ref}, y_{ref}) is known for any time instant in the future and, as already mentioned, θ_{ref} will be given by (9), to solve the control law. At time n, the mobile being at $\mathbf{z}_n$, in order to match the required trajectory, the control actions should be computed by (8), (10), leading to a sharp reply with strong control actions.

2.2.1. Ideal (hard) controller

1. Given $(x_{ref,n+i}, y_{ref,n+i})$ for $i = 1,2,\ldots$, and some initial conditions $\mathbf{z}_n$.
2. Compute the position increments $\Delta x_n = x_{ref,n+1} - x_n$; $\Delta y_n = y_{ref,n+1} - y_n$.
3. Compute the control actions rate, W_n/V_n , by (10).
4. Derive the next orientation angle $\theta_{ref,n+1}$ by (9).
5. According to the selected sampling period, T_s, compute the required angular speed (8)

$$W_n = \frac{\theta_{ref,n+1} - \theta_n}{T_s}$$

6. Compute the required linear speed V_n from (10).

This control will force to follow the required trajectory since the first control action, leading to unacceptable control magnitudes and involving some discrete approximation errors in the model (2) if there are step changes in the reference path.

2.2.2. Proportional controller

To get a smooth trajectory, the desirable next state, $\mathbf{z}_{d,n+1}$, is not assumed to be the new reference state value. This state vector $(\mathbf{z}_{d,n+1})$, assuming an approaching proportional to the error, is given by

$$\mathbf{z}_{d,n+1} = \begin{bmatrix} x_{d,n+1} \\ y_{d,n+1} \\ \theta_{d,n+1} \end{bmatrix} = \underbrace{\begin{bmatrix} x_{ref,n+1} \\ y_{ref,n+1} \\ \theta_{ref,n+1} \end{bmatrix}}_{\mathbf{z}_{ref,n+1}} - k\underbrace{\begin{bmatrix} x_{ref,n} - x_n \\ y_{ref,n} - y_n \\ \theta_{ref,n} - \theta_n \end{bmatrix}}_{\mathbf{e}_n = \mathbf{z}_{ref,n} - \mathbf{z}_n}, \quad \mathbf{e}_n = \begin{bmatrix} e_{x,n} \\ e_{y,n} \\ e_{\theta,n} \end{bmatrix}$$

$$= \begin{bmatrix} x_{ref,n} - x_n \\ y_{ref,n} - y_n \\ \theta_{ref,n} - \theta_n \end{bmatrix} \tag{11}$$

where k is a design positive parameter $(0 < k < 1)$. Note that:

1. if $k = 0$, $(\mathbf{z}_{d,n+1} = \mathbf{z}_{ref,n+1})$, the goal is to reach the reference trajectory in one step.
2. if $k = 1$, the error will remain constant, $(\mathbf{z}_{d,n+1} - \mathbf{z}_n = \mathbf{z}_{ref,n+1} - \mathbf{z}_{ref,n})$.

Replacing $\mathbf{z}_{n+1}$ by $\mathbf{z}_{d,n+1}$ in the first two components, the proposed controller will be

1. Given $(x_{ref,n+i}, y_{ref,n+i})$ for $i = 1,2,\ldots$, and some initial conditions $\mathbf{z}_n$.
2. Compute the new targeted point, (11)
   $x_{d,n+1} = x_{ref,n+1} - k(x_{ref,n} - x_n)$; $y_{d,n+1} = y_{ref,n+1} - k(y_{ref,n} - y_n)$
3. Compute the position increments $\Delta x_n = x_{d,n+1} - x_n$; $\Delta y_n = y_{d,n+1} - y_n$.
4. Compute the control actions rate, W_n/V_n , by (10).
5. Derive the next reference for the orientation angle $\theta_{ref,n+1}$ by (9).
6. The desired orientation angle will be: $\theta_{d,n+1} = \theta_{ref,n+1} - k(\theta_{ref,n} - \theta_n)$.
7. According to the selected sampling period, T_s, compute the required angular speed (8)

$$W_n = \frac{\theta_{d,n+1} - \theta_n}{T_s} \tag{12}$$

8. Compute the required linear speed V_n from (10).

Remark 1. In order to calculate the control actions, x_{n+1}, y_{n+1} and θ_{n+1} are replaced by $x_{d,n+1}, y_{d,n+1}$ and $\theta_{d,n+1}$ according to (11) in Δx_n, Δy_n and $\Delta\theta_n$.

Remark 2. The selection of the sampling period, T_s, will determine the magnitude of the control actions (last two steps of the control algorithm), as well as the validity of the discrete time model approximation.

Theorem 1. *The mobile robot described by (1), assuming an appropriate sampling period T_s, controlled by the proportional controller described above, leads to an asymptotically stable system.*

Proof. The state evolution, from (4) to (12), can be expressed by

$$x_{n+1} = x_n + T_s V_n \underbrace{\frac{\sin\theta_{n+1} - \sin\theta_n}{\theta_{n+1} - \theta_n}}_{f(\theta_{n+1})} = x_n + T_s V_n f(\theta_{n+1}) y_{n+1}$$

$$= y_n + T_s V_n \underbrace{\frac{\cos\theta_n - \cos\theta_{n+1}}{\theta_{n+1} - \theta_n}}_{g(\theta_{n+1})} = y_n + T_s V_n g(\theta_{n+1}) \theta_{n+1}$$

$$= \theta_n + T_s W_n \tag{13}$$

as a function of the control actions (12),

$$\theta_{n+1} = \theta_n + T_s W_n$$

$$\theta_{n+1} = \theta_n + T_s \frac{\theta_{ref,n+1} - k(\theta_{ref,n} - \theta_n) - \theta_n}{T_s}$$

$$e_{\theta,n+1} = k\ e_{\theta,n},\ 0 < k < 1 \Rightarrow e_{\theta,n+1} \to 0,\ n \to \infty \tag{14}$$

By using the Taylor interpolation rule, the functions $f(\theta_{n+1})$, $g(\theta_{n+1})$ can be expressed as

$$f(\theta_{n+1}) = \frac{\sin\theta_{n+1} - \sin\theta_n}{\theta_{n+1} - \theta_n}$$

$$= f(\theta_{ref,n+1}) + \underbrace{\left.\frac{df}{d\theta_{n+1}}\right|_{\theta_{n+1}=\psi_{\xi,n+1}}}_{f'(\psi_{\xi,n+1})} (\theta_{n+1} - \theta_{ref,n+1})$$

where $\psi_{\xi,n+1}$ is an interpolation point between θ_{n+1} and $\theta_{ref,n+1}$,

$$\psi_{\xi,n+1} = \theta_{ref,n+1} + \xi(\theta_{n+1} - \theta_{ref,n+1});\quad 0 < \xi < 1$$

Thus, (13) will be

$$x_{n+1} = x_n + T_s V_n [f(\theta_{ref,n+1}) + f'(\psi_{\xi,n+1})(\theta_{n+1} - \theta_{ref,n+1})]$$

From (10) and considering (11), (12) ,

$$V_n = \frac{\Delta x_n^2 + \Delta y_n^2}{2(\Delta y_n \cos\theta_n - \Delta x_n \sin\theta_n)} \frac{\theta_{ref,n+1} - k e_{\theta,n} - \theta_n}{T_s}$$

Thence,

$$V_n = \frac{\Delta x_n^2 + \Delta y_n^2}{2(\Delta y_n \cos\theta_n - \Delta x_n \sin\theta_n)} \frac{\theta_{ref,n+1} - \theta_n}{T_s} - \frac{\Delta x_n^2 + \Delta y_n^2}{2(\Delta y_n \cos\theta_n - \Delta x_n \sin\theta_n)} \frac{k e_{\theta,n}}{T_s} \tag{15}$$

Thus, introducing V_n in the x_{n+1} equation,

$$x_{n+1} = x_n + T_s V_n f(\theta_{n+1}) = x_n + T_s \left(\frac{\Delta x_n^2 + \Delta y_n^2}{2(\Delta y_n \cos\theta_n - \Delta x_n \sin\theta_n)} \frac{\theta_{ref,n+1} - \theta_n}{T_s} - - \frac{\Delta x_n^2 + \Delta y_n^2}{2(\Delta y_n \cos\theta_n - \Delta x_n \sin\theta_n)} \frac{k e_{\theta,n}}{T_s} \right)$$

$$(f(\theta_{ref,n+1}) + f'(\psi_{\xi,n+1})(\theta_{n+1} - \theta_{ref,n+1}))$$

Considering (9) and operating,

$$x_{n+1} = x_n + \Delta x_n + T_s \left(\frac{\Delta x_n^2 + \Delta y_n^2}{2(\Delta y_n \cos\theta_n - \Delta x_n \sin\theta_n)} \frac{\theta_{ref,n+1} - \theta_n}{T_s} \right) f'(\psi_{\xi,n+1})$$

$$(\theta_{n+1} - \theta_{ref,n+1}) - - T_s \left(\frac{\Delta x_n^2 + \Delta y_n^2}{2(\Delta y_n \cos\theta_n - \Delta x_n \sin\theta_n)} \frac{k e_{\theta,n}}{T_s} \right)$$

$$(f(\theta_{ref,n+1}) + f'(\psi_{\xi,n+1})(\theta_{n+1} - \theta_{ref,n+1}))$$

With,

$$T_s \frac{\Delta x_n^2 + \Delta y_n^2}{2(\Delta y_n \cos\theta_n - \Delta x_n \sin\theta_n)} \frac{\theta_{ref,n+1} - \theta_n}{T_s} \underbrace{\frac{\sin\theta_{ref,n+1} - \sin\theta_n}{\theta_{ref,n+1} - \theta_n}}_{f(\theta_{ref,n+1})} = \Delta x_n$$

Then we define,

$$h_{1,n} = T_s \left(\frac{\Delta x_n^2 + \Delta y_n^2}{2(\Delta y_n \cos\theta_n - \Delta x_n \sin\theta_n)} \frac{\theta_{ref,n+1} - \theta_n}{T_s} \right) f'(\psi_{\xi,n+1})(\theta_{n+1} - \theta_{ref,n+1}) - - \left(\frac{\Delta x_n^2 + \Delta y_n^2}{2(\Delta y_n \cos\theta_n - \Delta x_n \sin\theta_n)} \frac{k e_{\theta,n}}{T_s} \right)$$

$$(f(\theta_{ref,n+1}) + f'(\psi_{\xi,n+1})(\theta_{n+1} - \theta_{ref,n+1}))$$

Thence,

$$x_{n+1} = x_n + \Delta x_n + h_{1,n}$$

According to (11),

$$\Delta x_n = x_{d,n+1} - x_n = \underbrace{x_{ref,n+1} - k(x_{ref,n} - x_n)}_{x_{d,n+1}} - x_n$$

leading to

$$e_{x,n+1} = k\ e_{x,n} - h_{1,n};\quad e_{x,n} = x_{ref,n} - x_n \tag{16}$$

Applying the same reasoning to the y-coordinate, and taking into account that from (4), (11), (12) and Remark 1,

it yields, in compact form:

$$\begin{bmatrix} e_{x,n+1} \\ e_{y,n+1} \end{bmatrix} = \begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix} \begin{bmatrix} e_{x,n} \\ e_{y,n} \end{bmatrix} - \begin{bmatrix} h_{1,n} \\ h_{2,n} \end{bmatrix} \tag{17}$$

where

$$h_{2,n} = T_s \left(\frac{\Delta x_n^2 + \Delta y_n^2}{2(\Delta y_n \cos\theta_n - \Delta x_n \sin\theta_n)} \frac{\theta_{ref,n+1} - \theta_n}{T_s} \right) g'(\psi_{\varepsilon,n+1})(\theta_{n+1} - \theta_{ref,n+1}) -$$

$$- T_s \left(\frac{\Delta x_n^2 + \Delta y_n^2}{2(\Delta y_n \cos\theta_n - \Delta x_n \sin\theta_n)} \frac{k e_{\theta,n}}{T_s} \right) (g(\theta_{ref,n+1}) + g'(\psi_{\varepsilon,n+1})(\theta_{n+1} - \theta_{ref,n+1}))$$

$$\psi_{\varepsilon,n+1} = \theta_{ref,n+1} + \varepsilon(\theta_{n+1} - \theta_{ref,n+1});\quad 0 < \varepsilon < 1$$

is another intermediate point for $g(\cdot)$.

Taking account (14), $(h_{1,n}, h_{2,n}) \to 0$, $n \to \infty$. Thus, for $|k| < 1$, the error (17) tends to zero, and without oscillations, if $k > 0$.

2.3. Algorithm for controller implementation

Based on the theoretical results in the previous sections, assuming a sampling period T_s at each sampling time the following operations to calculate the control actions are performed.

Step 1: Get the state variables (x_n, y_n, θ_n)

Step 2: Calculate $\Delta x_n = x_{d,n+1} - x_n = x_{ref,n+1} - k(x_{ref,n} - x_n) - x_n$

$\Delta y_n = y_{d,n+1} - y_n = y_{ref,n+1} - k(y_{ref,n} - y_n) - y_n$

Step 3: Calculate $\frac{W_n}{V_n} = \frac{2(\Delta y_n \cos\theta_n - \Delta x_n \sin\theta_n)}{\Delta x_n^2 + \Delta y_n^2}$

Step 4: Calculate $\theta_{ref,n+1} = \text{atan2}\left(\frac{W_n}{V_n}\Delta x_n + \sin\theta_n, -\frac{W_n}{V_n}\Delta y_n + \cos\theta_n\right)$

Step 5: Calculate $\Delta\theta_n = \theta_{ref,n+1} - k(\theta_{ref,n} - \theta_n) - \theta_n$

Step 6: Calculate the control actions.

$$W_n = \frac{\Delta\theta_n}{T_s}$$

$$V_n = \frac{\Delta x_n^2 + \Delta y_n^2}{2(\Delta y_n \cos\theta_n - \Delta x_n \sin\theta_n)} W_n$$

If there is saturation in the control actions, the sampling period should be reduced.

3. Controller design under uncertainty

Now, an additive uncertainty is incorporated into the model of the system and an approach to eliminate its influence on the tracking error is proposed. Then, considering (7), the following mobile robot model is assumed

$$z_{n+1} = z_n + B_n(W_n, \theta_n, \theta_{n+1}) u_n + \mathbf{E}_n \tag{18}$$

where $\mathbf{E}_n$ is the additive uncertainty. Notice that the additive uncertainty can be used to model perturbed systems as well as a wide class of model mismatches.

Taking into account that the mismatch might depend on the state and on the input of the system, consider a real plant $\mathbf{z}_{n+1} = \mathbf{f}(\mathbf{z}_n, \mathbf{u}_n)$, the additive uncertainty can be expressed by $\mathbf{E}_n = \mathbf{f}(\mathbf{z}_n, \mathbf{u}_n) - \hat{\mathbf{f}}(\mathbf{z}_n, \mathbf{u}_n)$, where $\hat{\mathbf{f}}(\mathbf{z}_n, \mathbf{u}_n)$ is the discrete-time nonlinear model of the system. Note that if, as it will be assumed, $\mathbf{z}$ and $\mathbf{u}$ are bounded and $\mathbf{f}$ is Lipschitz, then $\mathbf{E}_n$ can be modeled as a bounded uncertainty (Michalska & Mayne 1993, and Mayne, Rawlings, Rao & Scokaert, 2000).

Replacing the control actions given in Step 6 Section 2.3 into the model (18), the tracking error evolution will be

$$\mathbf{e}_{n+1} = \begin{bmatrix} e_{x,n+1} \\ e_{y,n+1} \\ e_{\theta,n+1} \end{bmatrix} = k\ \mathbf{e}_n - \begin{bmatrix} h_{1,n} \\ h_{2,n} \end{bmatrix} - \mathbf{E}_n \tag{19}$$

Now, looking at (19), a direct effect of the additive uncertainty on the tracking error can be seen.

3.1. Integral action

In order to reduce the effect of $\mathbf{E}_n$, some integrators of the tracking errors in the system state variables will be introduced, depending on the time variation hypothesis of $\mathbf{E}_n$. It is assumed that $\mathbf{E}_n$is unknown and each component is an m-order polynomial.

Remark 3. The first order difference of $\mathbf{E}_n$ is defined as $\delta\mathbf{E}_n = \mathbf{E}_{n+1} - \mathbf{E}_n$, the second order difference as $\delta^2\mathbf{E}_n = \delta(\delta\mathbf{E}_n) = \delta(\mathbf{E}_{n+1} - \mathbf{E}_n) = \mathbf{E}_{n+2} - 2\mathbf{E}_{n+1} + \mathbf{E}_n$, and as a rule, the q-th order difference is defined as $\delta^q\mathbf{E}_n = \delta(\delta^{q-1}\mathbf{E}_n)$.

Remark 4. The q-th difference of a q-1 order polynomial is zero.

Let us consider a constant uncertainty $\mathbf{E}_n = const.$ that means: $\delta\mathbf{E}_n = \mathbf{E}_{n+1} - \mathbf{E}_n = \mathbf{0}$. In this case, an integrator for each state variable will force the error to converge to zero.

Denoting by $\mathbf{e}(t)$ the continuous time error in the state vector, define

$$\mathbf{U}_{1,n+1} = \mathbf{U}_{1,n} + \int_{nT_s}^{(n+1)T_s} \mathbf{e}(t)dt \cong \mathbf{U}_{1,n} + T_s\mathbf{e}_n \tag{20}$$

as the integral of the error, the control action (12) will be computed assuming a new term in (11), such as,

$$\mathbf{z}_{d,n+1} = \mathbf{z}_{ref,n+1} - \underbrace{k\,(\mathbf{z}_{ref,n} - \mathbf{z}_n)}_{\mathbf{e}_n} + K_1\mathbf{U}_{1,n+1} \tag{21}$$

where k, K_1 are, respectively, the proportional and integral control actions.

Replacing (21) in (12), and altogether in (18), after some simple operations, it yields

$$\mathbf{e}_{n+2} + (-k + K_1T_s - 1)\mathbf{e}_{n+1} + k\mathbf{e}_n = \underbrace{\begin{bmatrix} h_{1,n+1} - h_{1,n} \\ h_{2,n+1} - h_{2,n} \\ 0 \end{bmatrix}}_{\text{Nonlinearity}} - \underbrace{(\mathbf{E}_{n+1} - \mathbf{E}_n)}_{\delta\mathbf{E}_n = \mathbf{0}} \tag{22}$$

Therefore, k, K_1 are chosen in order to ensure the stability of the linear system represented in the left part of (22), that is, the zeros of this polynomial (r_i) should be inside the unit circle. Then $\sqrt{e_{x,n}^2 + e_{y,n}^2} \to 0$, as $n \to \infty$. That is, the tracking error tends to zero despite of uncertainties, if they are constant.

3.2. Multiple integral actions

Let us now consider that the uncertainty can be modeled by a function where the second order difference is zero, such that $\delta^2\mathbf{E}_n = \delta(\delta\mathbf{E}_n) = \delta(\mathbf{E}_{n+1} - \mathbf{E}_n) = \mathbf{E}_{n+2} - 2\mathbf{E}_{n+1} + \mathbf{E}_n = \mathbf{0}$. Then, a double integrator should be introduced in a similar way to (20), defining the integrating variables $\mathbf{U}_1$, $\mathbf{U}_2$.

$$\mathbf{U}_{2(n+1)} = \mathbf{U}_{2(n)} + \int_{nT_s}^{(n+1)T_s} \mathbf{U}_1(t)dt \cong \mathbf{U}_{2(n)} + T_s\mathbf{U}_{1(n+1)}$$

In this case, the control action (12) will be computed assuming an additional term in (20), such as

$$\mathbf{z}_{d,n+1} = \mathbf{z}_{ref,n+1} - \underbrace{k\,(\mathbf{z}_{ref,n} - \mathbf{z}_n)}_{\mathbf{e}_n} + K_1\mathbf{U}_{1,n+1} + K_2\mathbf{U}_{2,n+1} \tag{23}$$

where k, K_1, K_2 are, respectively, the proportional, integral and double integral control actions. Operating as before, and taking into account that $\delta^2\mathbf{E}_n = \mathbf{0}$, the error dynamics can be expressed by

$$\mathbf{e}_{n+3} + (-k + T_s(K_1 + T_sK_2) - 2)\mathbf{e}_{n+2} + (2k - T_sK_1 + 1)\mathbf{e}_{n+1} - k\mathbf{e}_n = \underbrace{\begin{bmatrix} h_{1,n+2} - 2h_{1,n+1} + h_{1,n} \\ h_{2,n+2} - 2h_{2,n+1} + h_{2,n} \\ 0 \end{bmatrix}}_{\text{Bounded Nonlinearity}} - \underbrace{\delta^2\mathbf{E}_n}_{=\mathbf{0}} \tag{24}$$

Now, as can be seen in (24), under the assumption of constant or linear varying uncertainty, $\delta^2\mathbf{E}_n = \mathbf{0}$, and the uncertainty has no influence on the error dynamics. The controller parameters, k, K_1, K_2 are chosen in order to ensure the stability of the linear system represented in the left part of (24), as shown in the previous case.

Following a similar reasoning, if the uncertainties can be approximated with a q-1 order polynomial, the influence of $\mathbf{E}_n$ on $\mathbf{e}_n$ will be eliminated by introducing q integrators.

Remark 5. The controller parameters can be chosen differently for each state variable, as pointed out in (11) for k, but not relevant benefit is obtained in the examples developed later on. The same could be done with all the controller parameters k, K_1, K_2

4. Simulations results

The simulation results for the performance evaluation of the trajectory tracking controller proposed in the previous section are presented in this section. Thus, our approach is applied to the dynamical model. As already discussed, the controlled system behavior depends on the parameters k, K_1, K_2,… K_p, where p is the number of integrators. Thus, in this work, and in order to determine values for the parameters of the controller for $p=0,1,2$, the MC method used in Auat Cheein and Carelli (2012), is applied. In addition, this experience allows measuring the controller behavior with $p=0$ (driver developed in Scaglia et al., 2010), with $p=1$ (controller proposed in this work with an integrator), and finally with $p=2$ (proposed in this paper by the use of two integrators) in order to compare the results.

4.1. Determination of the controller parameters by Monte Carlo method

In this section the controller's performance is analyzed by simulation when the controller parameters vary according to the MC experiment. An additional objective of the MC experiment is to find the parameter values optimizing a defined cost function. An idea widely used in the literature is to consider the cost incurred by the error. Let Φ be a desired trajectory, where $\#\Phi$ is the number of points of such trajectory. Let $C_x^\Phi = \sum_{i=0}^{\#\Phi} \frac{1}{2}(x_{(i)} - x_{ref\,(i)})^2$ the quadratic error in the *x-coordinate*; and $C_y^\Phi = \sum_{i=0}^{\#\Phi} \frac{1}{2}(y_{(i)} - y_{ref\,(i)})^2$the quadratic error in the *y-coordinate*, as proposed in Batavia & Singh (2002). Then, the cost function can be represented by the combination of both, the quadratic error in *x-coordinate* and the *y-coordinate* as shown in (25)

$$C^\Phi = C_x^\Phi + C_y^\Phi = \sum_{i=0}^{\#\Phi} \frac{1}{2}\left((x_{(i)} - x_{ref\,(i)})^2 + \left(y_{(i)} - y_{ref\,(i)}\right)^2\right) \tag{25}$$

Thus, the objective is to find k, K_1, K_2,… K_p, in such way that C^Φ

is minimized. In order to do so, in this work a MC based sampling experiment (Auat Cheein & Carelli, 2012) is carried out, with $p=0$, 1, 2. The MC experiment allows finding empirically the parameter values minimizing the cost function.

The MC experiment's considerations:

- The model mismatch between (1) and the Pioneer actual behavior is represented by the uncertainty $\mathbf{E}_n$, with high (unknown) order difference.
- The simulations are performed using MatLab software platform and Mobile Sim program provided by the manufacturer Pioneer Mobile Robot. The simulations are performed with $p=$ 0, 1, 2. For each controller 1000 simulations are run.
- All simulations are implemented with the same desired trajectory Φ. In this section, a sinusoidal trajectory is considered. The sampling time used is $T_s=$ 0.1 s.
- For each simulation, the controller parameters are chosen in a random way, such that the roots (r_i) of the linear systems defined in the right hand side of (19) and the left hand of (22) and (24) are $r_i \in (a, b)$, where b should be <1 to ensure system stability (error convergence) and a >0 and not too low for proper robot response. In our case, $a=0.88$ and $b=0.99$ were empirically chosen considering a trade off between the speed of convergence to zero of tracking errors and a "soft" robot response. That is, all the roots are $r_i = rand(0.88, 0.99)$.
- For $p=0$, 1000 simulations are performed and the controller parameter is chosen as $k = r_1$, where r_1 is the root of the linear system in (19).
- For $p=1$, 1000 simulations are performed again, and the controller parameters are chosen according to (26):

$$k = r_1 r_2$$
$$K_1 = \frac{-r_1 - r_2 + r_1 r_2 + 1}{T_s} \tag{26}$$

where, r_1 and r_2 are the roots of the linear system in (22).

- For $p=2$, 1000 simulations are performed and the controller parameters are chosen considering (27):

$$k = r_1 r_2 r_3$$
$$K_1 = \frac{2r_1 r_2 r_3 + 1 - r_1 r_2 - r_1 r_3 - r_2 r_3}{T_s}$$
$$K_2 = \frac{-r_1 - r_2 - r_3 - r_1 r_2 r_3 + 1 + r_1 r_2 + r_1 r_3 + r_2 r_3}{T_s^2} \tag{27}$$

Where r_1, r_2 and r_3 are the roots of the linear system in (24).

- The cost C^{Φ} for each controller is calculated when the robot reaches the desired trajectory. Due to the unknown character of the uncertainty, the steady state error will be affected by the higher order uncertainty differences, not canceled by the reduced number of integrators.

Fig. 2a shows the results of the 1000 simulations when the original controller developed by the authors in Scaglia et al. (2010) is used, ($p=0$ in the expression of controller proposed in this paper). The results show the values taken by the cost function for each simulation; scattered values are obtained due to the randomness with which the parameters were chosen in each simulation. The Fig. 2a shows that the lowest cost obtained with $p=0$ corresponds to $C^{\Phi} = 0.0439$. The Fig. 2b shows the results of the cost function C^{Φ} for 1000 iterations when using the controller proposed in this work when $p=1$, which corresponds to using one integrator. For this controller the lowest cost obtained is $C^{\Phi} = 0.0210$. In addition, all the cost obtained with $p=1$ are under $C^{\Phi} = 0.0439$, minimum value obtained when $p=0$. Finally Fig. 2c shows the results obtained when applying the proposed controller in this work with $p=2$. In this case the lowest cost obtained corresponds to $C^{\Phi} = 0.0058$. The minimum cost obtained for each controller is shown in Fig. 2d.

The analysis of the results shows that the performance of the controller improves as p increases. Thus, the results obtained by the MC experiment to choose the controller parameters verify the theoretical results obtained in the previous section. Table 1 shows the summary of the results obtained with each controller.

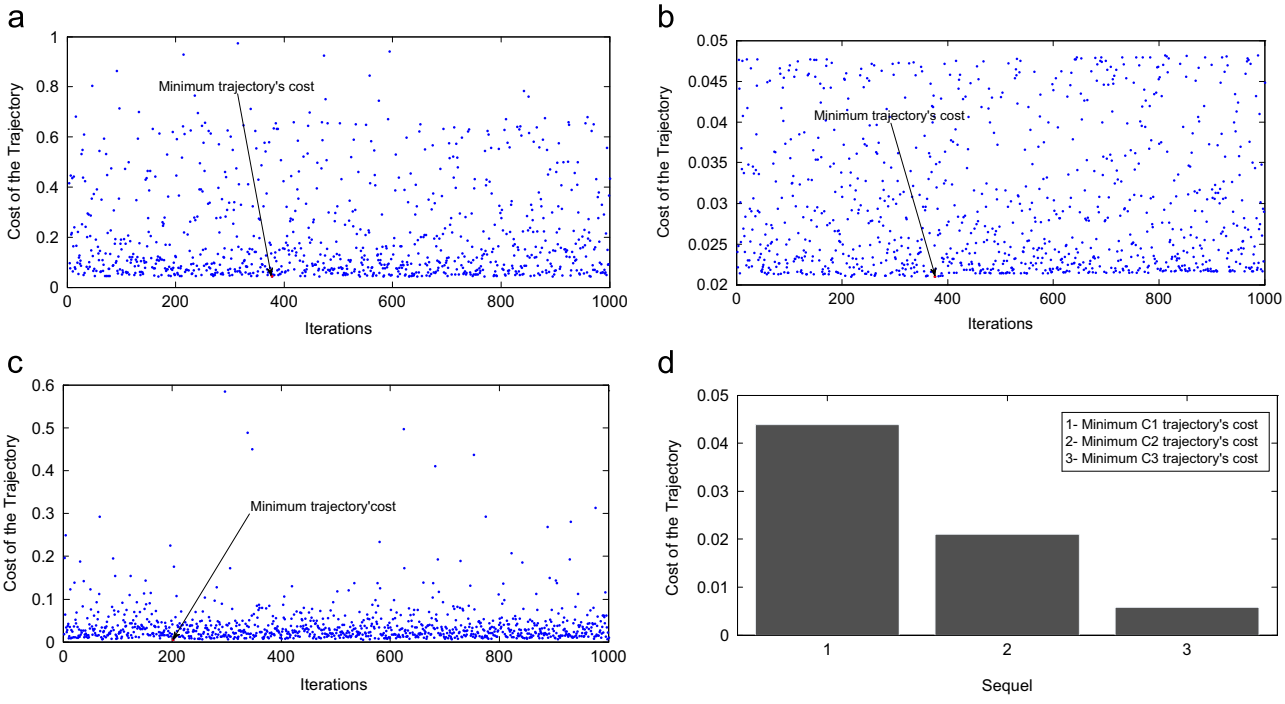

Fig. 2. a- The trajectory's cost whit $p=0$; b- the trajectory's cost whit $p=1$; c- the trajectory's cost whit $p=2$; d- the minimum cost of the trajectory obtained for each controller.

Table 1
Simulations summary of the Monte Carlo experiment.

Controller/ results	Minimum trajectory's cost: C^{Φ}	Controller's parameters
Controller with $p=0$	$C^{\Phi} = 0.0439$	$k = 0.96$
Controller with $p=1$	$C^{\Phi} = 0.0210$	$k = 0.883$ $K_1 = 0.027$
Controller with $p=2$	$C^{\Phi} = 0.0058$	$k = 0.8827$ $K_1 = 0.0269$ $K_2 = 0.0011$

5. Experimentation results

To support our claims, several experiments were performed by using a PIONEER 3AT (P3-AT) mobile robot. The P3-AT mobile robot, Fig. 3, includes an estimation system based on odometric-based positioning system. This system uses 100 tick encoders with inertial correction recommended for dead reckoning to compensate for skid steering. Updating through external sensors is necessary. This problem is separated from the strategy of trajectory tracking and it is not considered in this paper, (Rico, Gomez-Ortega & Camacho, 1999; Rico, Alcala, Gomez-Ortega & Camacho, 2001). The P3-AT has a PID velocity controller used to maintain the velocities of the mobile robot at the desired value. In this section, various experiments are shown to evaluate the performance of the proposed controller in different scenarios. First the controller behavior when the reference trajectory corresponds to a square trajectory is shown. The aim of this experiment is to evaluate the controller performance when the trajectory changes abruptly, in direction. Then, in a second experiment the behavior of the controller is shown with the presence of disturbances in the control actions. Finally, an experiment is performed in order to compare the controller proposed in this work with a controller recently developed by other authors in the literature.

Remark 6. The experimental results shown in Section 5 are carried out only using the internal odometers. In addition, it is assumed that the robot does not slip and odometer errors are practically zero.

5.1. Square trajectory

The application field of mobile robots has increased recently. In different applications the trajectory to be followed by the robot is usually re-planned. This strategy can be used in applications such as obstacle avoidance and contour-following. So if the danger of collision is large, the trajectory to be followed by the robot is modified abruptly and the robot must follow that path to avoid collision. Thus, the controller performance when the trajectory changes abruptly will be analyzed. In addition, in order to test the limits of our formulation, and as recommended by (Roth & Batavia, 2002), a square trajectory was chosen. Specifically, the robot has to track a square trajectory. The square reference trajectory is generated with constant linear velocity of $V=0.3$ m/s. The initial position of the robot is at the system origin and the trajectory begins in the position $(x_{ref(0)}, y_{ref(0)})=(1\text{m},1\text{m})$.

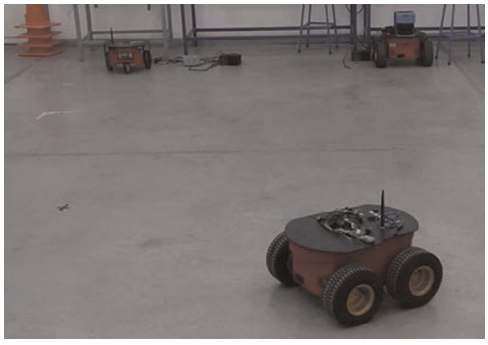

Fig. 3. The pionner 3AT mobile robot and the laboratory facilities.

In order to compare the performance of the controller proposed in this paper with the original controller (Scaglia et al., 2010), three experiments in the mobile robot Pioneer 3AT with three different controllers were performed. First, the original controller proposed by the authors in Scaglia et al. (2010) (called C1 in the sequel), which corresponds to $p=0$ in the approach proposed here. Then, an experiment with the controller proposed in this paper considering $p=1$ is addressed (C2 in the sequel). Finally, the performance of the controller proposed in this paper with two integrators, $p=2$ (C3 in the sequel) is shown. The controller parameters used for the experimentation are obtained from Table 1, and they were obtained by the MC experiment.

Fig. 4 shows the results of the implementation. By inspection, Fig. 4a shows that all controllers reach and follow the desired trajectory. However, for the square trajectory test, C3 shows the lowest error cost when compared to the rest of the controllers, Fig. 4b. This shows that if the number of integrators increases then the tracking error decreases. This result can also be observed in Fig. 4c and d, which shows the values taken tracking errors in the x-and y-coordinate coordinate. Note that due to the high order of the uncertainty difference, the tracking error is not fully canceled.

5.2. Disturbances in the control actions

In order to observe the behavior of the controller in the presence of additional uncertainty, disturbances are added to the control actions. The disturbance in the linear velocity at time n is represented by $V_n=V_n+\delta_V V_n$, where V_n is the linear velocity to be applied at the n instant and δ_V is represented by $\delta_V=\text{rand}(-0.2,0.2)$; i.e., rand$(-0.2,0.2)$ is a random noise with a magnitude of 0.2 and -0.2 lower bound. The disturbance in the angular velocity at time n is represented by $W_n=W_n+\delta_W W_n$, where W_n is the angular velocity to be applied in the n instant and δ_W is represented by $\delta_W=\text{rand}(-0.2,0.2)$; i.e., rand$(-0.2,0.2)$ is a random noise with a magnitude of 0.2 and -0.2 lower bound. The test performed herein, is aimed at showing the controller performance under disturbance in the control actions and sinusoidal trajectories, the desired trajectory is generated with a $V_{ref}=0.3$ m/s. The controller parameters are obtained from Table 1.

Fig. 5a shows the trajectory and the results obtained by implementing the controller proposed in this paper with $p=0, 1, 2$. As can be seen, the controllers reach and follow the desired trajectory without unexpected oscillations. Fig. 5b shows that C3 in the sequel has the lowest cost error when compared with the rest of the controller. Fig. 5c and d shows the absolute values of the tracking errors for the *x-coordinate* and *y-coordinate* respectively. By inspection, the tracking error when the controller has two integrators (C3) is the lowest and present a better performance against unwanted disturbances compared with C1 and C2.

5.3. Controller comparison

To test the advantages and drawbacks of our proposal, an experimental evaluation was carried out. In order to do so, two controllers previously published in the scientific literature were implemented for comparison on the mobile robot Pioneer 3AT. The controllers implemented for comparison are the following:

- Controller developed in Scaglia et al. (2010), C1 in the sequel ($p=0$ in our proposal).
- The controller proposed in this paper with $p=1$, C2 in the

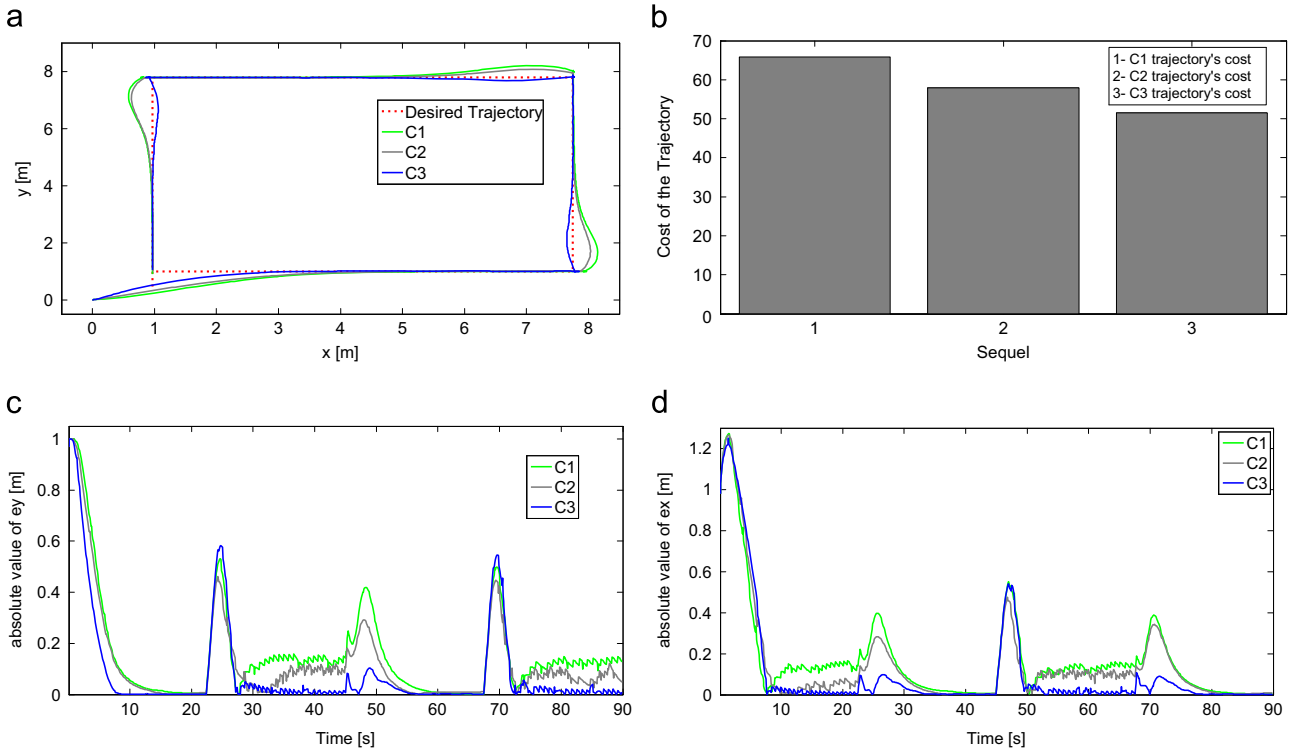

Fig. 4. a- Tracking trajectory of the mobile robot; b- cost of the trajectory tracking; c- absolute value of the tracking error in x; d- absolute value of the tracking error in y.

sequel.

- Our approach with $p=2$, C3 in the sequel.
- A non-linear trajectory tracking strategy for unicycle vehicles developed by Michalek & Kozlowski (2012), C4 in the sequel.

The design details of the drivers can be found in their respective references, and only the experimental results without a theoretical analysis of the controller's properties are shown here. For such, (Michalek & Kozlowski, 2012; Scaglia et al., 2010) offer a deep insight into the controller design.

The final test carried out is a curvature test, in which the controller's performance using different circle-sharped trajectories, as recommended in (Roth & Batavia, 2002) are evaluated. Three circle-trajectories were used in this work, whit different

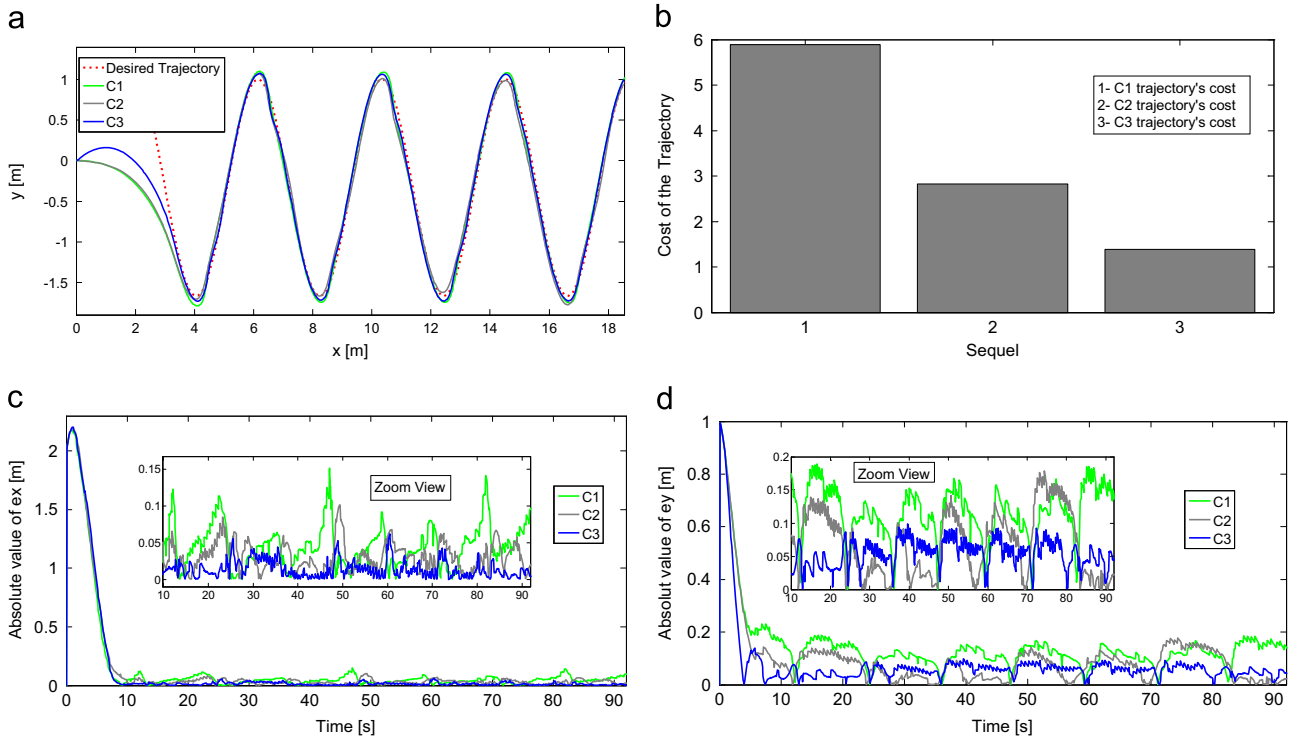

Fig. 5. a- Tracking trajectory of the mobile robot; b- cost of the trajectory tracking; c- absolute value of the tracking error in x; d- absolute value of the tracking error in y.

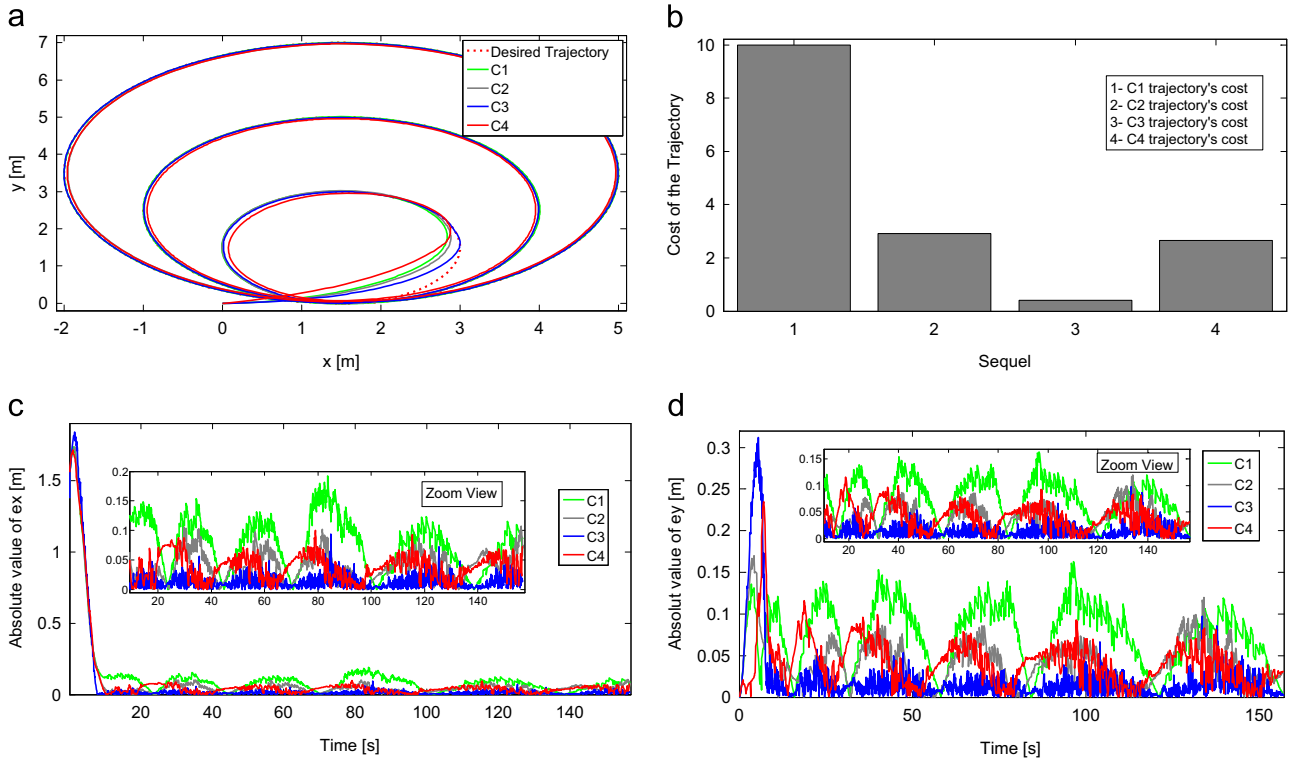

Fig. 6. a- Tracking trajectory of the mobile robot; b- cost of the trajectory tracking; c- absolute value of the tracking error in x; d- absolute value of the tracking error in y.

radius, as shown in Fig. 6a. The inner trajectory has a radius of $r=1.5$ m, the medium one $r=2.5$ m and the last one $r=3.5$ m. The initial position of the robot is at the system origin and the trajectory begins in the position $(x_{ref(0)}, y_{ref(0)}) = (1.5\text{ m}, 0\text{ m})$.

The reference trajectory and the results of the controllers are shown in Fig. 6a. As can be seen, all controllers reach and follow the desired trajectory. Fig. 6c and d shows the plots of the absolute value of the tracking error in the x-coordinate and y-coordinate according to each controller used in the test for the three curvatures shown in Fig. 6a. The controller proposed by Michalek & Kozlowski (2012), named C4 presents a similar performance that C2 (our approach whit $p=1$), but the lowest cost is obtained by C3 as can be seen in Fig. 6b.

The above results show the good performance of the controller presented against other controllers proposed in the literature. Furthermore, analyzing Fig. 6b it can be seen that the performance of the controller can be improved when the number of integrators is increased although the complexity of the control is also increased.

6. Conclusions

A methodology based on numerical methods and linear algebra to design control algorithms for a mobile robot when the system is represented by a multivariable non linear model with additive uncertainty, has been presented. New integrators have been added, which are chosen based on the variation hypothesis of $\mathbf{E}_n$. If $\mathbf{E}_n$ can be approximated by a q-grade polynomial, then $q+1$ integrators will be added in order to avoid the influence of the modeling error ($\mathbf{E}_n$) to tracking error (e_n).

Different tests were carried out to demonstrate the effectiveness of the proposed methodologies. The controller is tuned by simulation using MC method. These experiments show that the tracking error decreases when the number of integrators increases. The decrease of tracking error also is observable during experimental tests with disturbances in the control actions and in front of trajectories with sudden changes in their orientation. The performance of the proposed system is good, and the complexity of control algorithm does not increase in an excessive way.

In comparison with others previous published control laws Blažič (2011), Rossomando et al. (2011), Resende et al. (2013), Lee et al. (2009), Farooq et al. (2014) and Wang (2012) the proposed controller presents the advantages of being easy to design and to implement. Thus, the algorithm can be implemented directly on the robot's microcontroller without the need to implement it on an external computer, because the calculations are simple to perform. In addition, the method proposed here, does not need a model transformation compared with Blažič (2011). Furthermore, our controller does not present the disadvantage of Blažič (2011) where the reference velocities must meet the condition of persistent excitation.

Acknowledgments

This work was partially funded by the Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET - National Council for Scientific Research), Argentina, the Escuela Politécnica Nacional del Ecuador (Facultad de Ingeniería Eléctrica y Electrónica), and the Secretaría Nacional de Educación Superior, Ciencia y Tecnología (SENESCYT) through Prometeo Program.

References

Auat Cheein, F. L.Fd. F., & Carelli, R. (2012). Autonomous simultaneous localization and mapping driven by monte carlo uncertainty maps-based navigation. *The Knowledge Engineering Review*, *28*(1), 35–57.

Batavia, P., & Singh, S. (2002). Autonomous coverage operations in semi-structured outdoor environments. *IEEE Int. Conf. on intelligent Robots and Systems (IROS '02)*. Lausame, Switzerland, October, 2002.

Blažič, S. (2011). A novel trajectory-tracking control law for wheeled mobile robots.

Robotics and Autonomous Systems, *59*(11), 1001–1007.
Farooq, U., Hasan, K. M., Hanif, A., Amer, M., & Asad, M. U. (2014). Fuzzy logic based path tracking controller for wheeled mobile robots. *International Journal of Computer and Electrical Engineering*, *6*(2), 145–150.
Klancar, G., & Skrjanc, I. (2007). Tracking-error model-based predictive control for mobile robots in real time. *Robotics and Autonomous Systems*, *55*(6), 460–469.
Lee, J. H., Lin, C., Lim, H., & Lee, J. M. (2009). Sliding mode control for trajectory tracking of mobile robot in the RFID sensor space. *International Journal of Control Automation and Systems*, *7*(3), 429–435.
Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. M. (2000). Constrained model predictive control: stability and optimality. *Automatica*, *36*, 789–814.
Michalek, M., & Kozlowski, K. (2012). Feedback control framework for car-like robots using the unicycle controllers. *Robotica*, *30*, 517–535.
Michalska, H., & Mayne, D. Q. (1993). Robust receding horizon control of constrained nonlinear systems. *IEEE Transaction on Automatic Control*, *38*, 1623–1633.
Oriolo, G., Luca, A., & Vendittelli, M. (2002). WMR control via dynamic feedback linearization: design, implementation, and experimental validation. *IEEE Transactions on Control Systems Technology*, *10*(6), 835–852.
Resende, C. Z., Carelli, R., & Sarcinelli-Filho, M. (2013). A nonlinear trajectory tracking controller for mobile robots with velocity limitation via fuzzy gains. *Control Engineering Practice*, *21*(10), 1302–1309.
Rico, J. N., Alcala, I., Gomez-Ortega, J., & Camacho, E. (2001). Mobile robot path tracking using PID controller. *Control Engineering Practice*, *9*, 1209–1214.
Rico, J. N., Gomez-Ortega, J., & Camacho, E. (1999). A Smith predictor based generalized predictive controller for mobile robot path-tracking. *Control Engineering Practice*, *7*, 729–740.
Rosales, A., Scaglia, G. J. E., Mut, V., & Di Sciascio, F. (2009). Trajectory tracking of mobile robots in dynamic environments a linear algebra approach. *Robotica*, *27*, 981–997.
Rosales, A., Scaglia, G. J. E., Mut, V., & Di Sciascio, F. (2011). Formation control and trajectory tracking of mobile robotic systems – a Linear Algebra approach. *Robotica*, *29*(3), 335–349.
Rossomando, F. G., Soria, C., & Carelli, R. (2011). Autonomous mobile robots navigation using RBF neural compensator. *Control Engineering Practice*, *19*(3), 215–222.
Roth, S. & Batavia P. (2002). Evaluating path tracker performance for outdoor mobile robots. *Automation Technology for Off-Road Equipment*. July, 2002. ⟨http://www.ri.cmu.edu/publication_view.html?pub_id=3908⟩.
Scaglia, G. J. E., Mut, V. A., Jordan, M., Calvo, C., & Quintero, L. (2009). Robust-control-based controller design for a mobile robot. *Journal of Engineering Mathematics*, *63*(1), 17–32.
Scaglia, G. J. E., Quintero, L. M., Mut, V., & Di Sciascio, F. (2009). Numerical methods based controller design for mobile robots. *Robotica*, *27*(2), 269–279.
Scaglia, G. J. E., Rosales, A., Quintero, L. M., Mut, V., & Agarwal, R. (2010). A linear-interpolation-based controller design for trajectory tracking of mobile robots. *Control Engineering Practice*, *18*, 318–329.
Serrano, M. E., Scaglia, G. J. E., Godoy, S. A., Mut, V., & Ortiz, O. A. (2013). Tracking trajectory of underactuated surface vessels based on a linear algebra approach. *IEEE Transaction on Control Systems Technology*, *22*, 1103–1111.
Wang, K. (2012). Near-optimal tracking control of a nonholonomic mobile robot with uncertainties. *International Journal of Advanced Robotic Systems*, *9*, 1–11.