EPSAC Predictive Control Applied to Path Tracking of Wheeled Mobile Robots ⋆

Batista, Italo J. L. * **Torrico, Bismark C.** * **Nogueira, Fabricio G.** *
Normey-Rico, Julio E. **

* *Department of Electrical Engineering, Federal University of Ceará (UFC), CE (e-mail: bismarmk@dee.ufc.br, fnogueira@dee.ufc.br)*
** *Automation and Systems Department, Federal University of Santa Catarina (UFSC), SC (e-mail: julio.normey@ufsc.br)*

Abstract: This paper presents the application of the Extended Prediction Self-Adaptive Control (EPSAC) to the path tracking problem of wheeled mobile robots (WMRs). The main complexity of the control problem is that a WMR is a MIMO (multiple-input and multiple-output) system, being also nonsquare, nonlinear, and with input constraints. EPSAC, which belongs to the class of Model-based Predictive Controllers (MPC), is an interesting option to overcome such difficulties because its main properties are: (i) the control performance can be improved when the future reference trajectory is previously known as in the case of WMR, (ii) it has the ability to deal with constraints during the calculation of the control law in a fairly straightforward way, and (iii) with proper tuning in nonlinear models, a sub-optimal solution closer to nonlinear MPC can be obtained if compared to the conventional linear MPC. Simulations and experiments on a real robot have shown improved performance regarding trajectory tracking associated with reduced computational cost if compared to other MPCs applied to WMR proposed in literature.

Keywords: Wheeled Mobile Robots, Path Tracking, MPC, EPSAC.

1. INTRODUCTION

Mobile robots have been extensively studied in recent years because it is a complex problem and with a wide range of applications, including exploration, observation, searching and mapping in various types of environments. Thus, high precision movement presented by mobile robots is desirable in many applications.

The main difficulty in controlling path tracking of mobile robots lies in the fact that they are nonlinear, multivariable, and underactuated systems with movement constraints. The control system becomes even more complex because their respective models are nonsquare and presents more outputs than inputs. Typically, the inputs are the linear velocity v and the angular velocity w, while the output variables are the orientation θ and the Cartesian coordinate system "XY".

During the past few years, extensive research works have been dedicated to path tracking control in nonholonomic mobile robots. In most of them, the control signals are obtained using a combination of feedforward action calculated from a reference trajectory and a feedback control law. These strategies include: linearization of the kinematic model, back stepping approach, sliding mode control, Lyapunov-based nonlinear control, visual servoing control, robust control and fuzzy control.

Although the future reference trajectory is known in the aforementioned studies, this information is not entirely used by the mentioned control strategies. However, there is a class of controllers that take into account the future reference such as Model Preditive Control (MPC) (Camacho and Bordons, 2004). MPC also presents the following advantages over the classical laws: the adjustments are relatively simple because there is a reduction in the amount of tuning parameters; the multivariable case (MIMO system) can be treated in a simple way; it introduces feedforward in a natural way to compensate measurable disturbances; its extension to the treatment of constraints is conceptually simple and can be consistently included during the project (Camacho and Bordons, 2004).

Many successful MPC implementations have already been reported in literature regarding path tracking of wheeled mobile robots (WMRs). The works by Klancar and Skrjanc (2007) and Raffo et al. (2009) use MPC techniques based on a linearized model in local coordinates of the robot, where a quadratic cost function is minimized and the tracking error and control are consequently affected. The linear model in local coordinates allows the use of convex optimization algorithms and considerably reduces the computational effort. Barreto et. al. (2014) and Shangming et. al. (2013) present the implementation of a linear MPC scheme with friction compensation applied to trajectory following of an omnidirectional three-wheeled mobile robot.

Peng et al. (2022) propose a nonlinear MPC approach for WMRs. Some studies developed by Dai et al. (2021) and Khan et al. (2022) show how it is possible to improve robustness when an estimation of the uncertainties is considered in the design stage. However, the results

⋆ The authors gratefully acknowledge funding agencies CAPES e CNPq for financial support of this project.

obtained with nonlinear approaches usually demand high computational cost, which can make the implementation unfeasible in embedded real-time computing systems.

The EPSAC strategy uses both linear and nonlinear models to compute predictions. However the optimization procedure has the same complexity as that of linear-constrained MPC, but the solution is closer to the nonlinear MPC if compared with the linear MPC (Keyser, 2003). Within this context, this work proposes a design methodology and application of EPSAC to the path tracking control of WMRs, which is a multivariable, nonlinear, and nonsquare system. Besides, a procedure to adjust the control approach in order to obtain a solution closer to the nonlinear optimal solution is proposed. The results are compared with a standard linear MPC formulation, which is based on the successive linearization method (MPC-SL) along the reference trajectory (Kuhne et al., 2004).

The rest of the paper is organized as follows. Section 2 presents the robot kinematic model. Section 3 discusses the formulation of MPC-SL. Section 4 describes the EPSAC control theory used in this work, and also the algorithm used in the mobile robot. Simulation results are discussed and analyzed in Section 5. Experimental results on a real mobile robot are detailed in Section 6. Finally, Section 7 presents remarkable conclusions, while relevant issues are discussed in detail.

2. ROBOT MODELING

Let us consider the WMR shown in the coordinate system, whose rotation and translation movements are performed by two dc motors coupled directly to the drive from the robot wheels. Besides, a third wheel exists to support the robot base.

Point "C" is considered as a central reference point to describe the robot position. The Cartesian coordinate system "XY" denotes an inertial frame, while $(x(t), y(t))$ represents the coordinated movement of the platform relative to the point "C". The angle between the longitudinal axis of the robot and the horizontal axis is the orientation angle given by $\theta(t)$. Moreover, $v(t)$ and $w(t)$ represent the angular and linear velocities of the robot, respectively.

The kinematic equations of the robot motion are described by (Campion et al., 1996):

$$\begin{array}{c} \dot{x}(t) = v(t)\cos(\theta(t)), \\ \dot{y}(t) = v(t)\sin(\theta(t)), \\ \dot{\theta}(t) = w(t). \end{array} \tag{1}$$

The aforementioned expressions can also be represented in a simplified form as:

$$\dot{\mathbf{y}} = f(\mathbf{y}, \mathbf{u}) \tag{2}$$

where $\mathbf{y} = [x(t), y(t), \theta(t)]^T$ is the posture of the robot and $\mathbf{u} = [v \ w]^T$ is the control action.

The problem of trajectory tracking can be stated as to find a given control law such that $\mathbf{y}(t) - \mathbf{y_r}(t) = 0$, where $\mathbf{y_r} = [x_r(t), y_r(t), \theta_r(t)]^T$ is a pre-specified reference trajectory. In this case, the reference trajectory is often associated to a virtual reference robot, which has the same model as that of the robot to be controlled. Thus, it is possible to derive the following statement:

$$\dot{\mathbf{y}}_\mathbf{r} = f(\mathbf{y_r}, \mathbf{u_r}). \tag{3}$$

3. MPC-SL APPLIED TO PATH TRACKING

This section aims to review the MPC-SL strategy (Kuhne et al., 2004) applied to the path trajectory problem of a nonholonomic WMR. This strategy will be adopted for comparison purposes in Sections 5 and 6.

3.1 Kinematic model in local coordinates

The model used by MPC-SL is obtained from the linearization of (2) around the robot reference given by (3) (Kuhne et al., 2004):

$$\dot{\tilde{\mathbf{y}}} = f_{\mathbf{y},r}\tilde{\mathbf{y}} + f_{\mathbf{u},r}\tilde{\mathbf{u}} \tag{4}$$

where $\tilde{\mathbf{y}} = \mathbf{y} - \mathbf{y}_r$ represents the error with respect to the reference robot, $\tilde{\mathbf{u}} = \mathbf{u} - \mathbf{u}_r$ is associated with disturbances in the control action, $f_{\mathbf{y},r} = \dfrac{\partial f(\mathbf{y},\mathbf{u})}{\partial \mathbf{y}}$, and $f_{\mathbf{u},r} = \dfrac{\partial f(\mathbf{y},\mathbf{u})}{\partial \mathbf{u}}$.

Applying the forward Euler approximation to (4) gives:

$$\tilde{\mathbf{y}}(k+1) = \tilde{\mathbf{a}}(k)\,\tilde{\mathbf{y}}(k) + \tilde{\mathbf{b}}(k)\,\tilde{\mathbf{u}}(k) \tag{5}$$

where $\tilde{\mathbf{y}}(k) = \mathbf{y}(k) - \mathbf{y}_r(k)$, $\tilde{\mathbf{u}}(k) = \mathbf{u}(k) - \mathbf{u}_r(k)$,

$$\tilde{\mathbf{a}}(k) = \begin{bmatrix} 1 & 0 & -v_r(k)sin\theta_r(k)T \\ 0 & 1 & v_r(k)cos\theta_r(k)T \\ 0 & 0 & 1 \end{bmatrix},\ \tilde{\mathbf{b}}(k) = \begin{bmatrix} cos\theta_r(k)T & 0 \\ sin\theta_r(k)T & 0 \\ 0 & T \end{bmatrix},$$

T is the sampling period, and k is the sampling instant.

$$\tilde{\mathbf{Y}}(k) = \begin{bmatrix} \tilde{\mathbf{y}}(k+1|k) \\ \tilde{\mathbf{y}}(k+2|k) \\ \vdots \\ \tilde{\mathbf{y}}(k+N|k) \end{bmatrix},\ \tilde{\mathbf{U}}(k) = \begin{bmatrix} \tilde{\mathbf{u}}(k|k) \\ \tilde{\mathbf{u}}(k+1|k) \\ \vdots \\ \tilde{u}(k+N-1|k) \end{bmatrix},$$

$$\alpha(k,j,l) = \prod_{i=N-j}^{l} \tilde{\mathbf{a}}(k+i),\ \tilde{\mathbf{A}}(k) = \begin{bmatrix} \tilde{\mathbf{a}}(k) \\ \tilde{\mathbf{a}}(k+1)\tilde{\mathbf{a}}(k) \\ \vdots \\ \alpha(k,2,0) \\ \alpha(k,1,0) \end{bmatrix},$$

and

$$\tilde{\mathbf{B}}(k) = \begin{bmatrix} \tilde{\mathbf{b}}(k) & \mathbf{0} & \cdots & \mathbf{0} \\ \tilde{\mathbf{a}}(k+1))\tilde{\mathbf{b}}(k) & \tilde{\mathbf{b}}(k+1) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha(k,2,1) & \alpha(k,2,2) & \cdots & \mathbf{0} \\ \tilde{\mathbf{b}}(k) & \tilde{\mathbf{b}}(k+1) & & \mathbf{0} \\ \alpha(k,1,1) & \alpha(k,1,2) & \cdots & \tilde{\mathbf{b}}(k+N-1|k) \\ \tilde{\mathbf{b}}(k) & \tilde{\mathbf{b}}(k+1) & & \tilde{\mathbf{b}}(k+N-1|k) \end{bmatrix}.$$

3.2 Optimization Process

The concept used by the controller consists in calculating a control law so that it minimizes the cost function, which is defined by:

$$J(k) = \sum_{j=1}^{N} \tilde{\mathbf{y}}^T(k+j|k)\mathbf{Q}\tilde{\mathbf{y}}(k+j|k) + \sum_{j=0}^{N-1} \tilde{\mathbf{u}}(k+j|k)^T\mathbf{R}\tilde{\mathbf{u}}(k+j|k)]^2 \quad (6)$$

subject to:

$$\mathbf{u}_{\min} \le \mathbf{u}(k+j|k) \le \mathbf{u}_{\max} \ \ \forall \ \ j = 0, ..., N-1,$$

where N is the prediction window, $\mathbf{Q}$ and $\mathbf{R}$ are the output and the control weight respectively, and $\mathbf{u}_{\min}$ and $\mathbf{u}_{\max}$ are the lower and upper bounds of the control action, respectively. The solution of such optimization problem and also computational cost are crucial in MPC algorithms so that their implementation in real physical systems becomes feasible.

In order to use such algorithms, the cost function must be represented as a function of the decision variable $\tilde{\mathbf{U}}(k)$ in the quadratic form, that is:

$$J(k) = \frac{1}{2}\tilde{\mathbf{U}}^T(k)\tilde{\mathbf{H}}(k)\tilde{\mathbf{U}}(k) + \tilde{\mathbf{f}}^T(k)\tilde{\mathbf{U}}(k) + \tilde{\mathbf{k}}(k), \quad (7)$$

subject to: $\mathbf{G}(k)\tilde{\mathbf{U}}(k) \le \mathbf{M}(k)$

where:
$\tilde{\mathbf{H}}(k) = 2\left(\tilde{\mathbf{B}}^T(k)\tilde{\mathbf{Q}}\tilde{\mathbf{B}}(k) + \tilde{\mathbf{R}}\right)$,
$\tilde{\mathbf{f}}(k) = 2\tilde{\mathbf{B}^T}(k)\tilde{\mathbf{Q}}\tilde{\mathbf{A}}(k)\tilde{\mathbf{y}}(k)$,
$\tilde{\mathbf{k}}(k) = \tilde{\mathbf{y}}^T(k)\tilde{\mathbf{A}^T}(k)\tilde{\mathbf{Q}}\tilde{\mathbf{A}}(k)\tilde{\mathbf{y}}(k)$,
$\tilde{\mathbf{Q}} = diag(\mathbf{Q}; \cdots; \mathbf{Q})$ and $\tilde{\mathbf{R}} = diag(\mathbf{R}; \cdots; \mathbf{R})$.

In order to reduce computational cost in practice, it is common to solve (7) analytically without considering the constraints and saturating the control action using $\mathbf{u}_{\min}$ and $\mathbf{u}_{\max}$. This approach is called MPC-SL-Sub in this work.

4. EPSAC APPLIED TO PATH TRACKING

The EPSAC is a linear MPC strategy proposed in De Keyser and Van Cauwenberghe (1985). In the 1980s several linear MPC techniques have been proposed De Keyser and Van Cauwenberghe (1985); Clarke et al. (1987) which at the time had a major impact. The relevance of these techniques has been mitigated due to the vested interest in the control of nonlinear systems. Thus many important work has emerged as regards its and stability however with a high degree of complexity with regard to practical implementation. Thus was born the NEPSAC for control of nonlinear systems in the literature and are several studies that showed success in the practical implementation, especially in processes with relatively large time constant. However, in cases such as mobile robotic sampling time can be critical. So this paper proposes the EPSAC based on some principles of NEPSAC.

$$\mathbf{u}(k+j|k) = \mathbf{u}_{\mathrm{b}}(k+j|k) + \mathbf{u}_{\mathrm{o}}(k+j|k), \quad (8)$$

where $\mathbf{u}_{\mathrm{b}}(k+j|k)$ is a constant called base input and $\mathbf{u}_{\mathrm{o}}(k+j|k)$ is the manipulated variable. Besides, the future output sequence can be written as:

$$\mathbf{y}(k+j|k) = \mathbf{y}_{\mathrm{b}}(k+j|k) + \mathbf{y}_{\mathrm{o}}(k+j|k), \quad (9)$$

where $\mathbf{y}_{\mathrm{b}}(k+j|k)$ is due to $\mathbf{u}_{\mathrm{b}}(k+j|k)$ and $\mathbf{y}_{\mathrm{o}}(k+j|k)$ is due to $\mathbf{u}_{\mathrm{o}}(k+j|k)$. $\mathbf{u}_{\mathrm{b}}(k+j|k)$ is chosen a priori and $\mathbf{u}_{\mathrm{o}}(k+j|k)$ is the decision variable of the cost function optimization problem which can be represented as a quadratic problem optimization with linear constraints. In this way, the computational cost is lower in comparison with the traditional nonlinear MPC and equivalent to linear MPC. This fact makes the EPSAC more attractive for practical applications. It is worth to mention that the choice of $\mathbf{u}_{\mathrm{b}}(k+j|k)$ is quite important in EPSAC. Improved response can be obtained if $\mathbf{u}_{\mathrm{b}}(k+j|k)$ is closer to the optimal nonlinear MPC $\mathbf{u}^*(k+j|k)$ (Keyser, 2003). Thus, this work proposes a method to improve the choice of $\mathbf{u}_{\mathrm{b}}(k+j|k)$ at each sampling period. EPSAC applied to WMR is presented as follows.

4.1 Computing the predictions

In order to compute the predictions, the discrete kinematic model of the WMR is used, which is obtained by applying the forward Euler approximation to (1), thus:

$$\begin{cases} x(k+1) = x(k) + v(k)cos\theta(k)T \\ y(k+1) = y(k) + v(k)sin\theta(k)T \\ \theta(k+1) = \theta(k) + w(k)T \end{cases} \quad (10)$$

or, in a compact representation,

$$\mathbf{y}(k+1) = f_d(\mathbf{y}(k), \mathbf{u}(k)). \quad (11)$$

By using EPSAC (Keyser, 2003), the output can be represented as:

$$\mathbf{y}(k) = \mathbf{x}(k) + \mathbf{n}(k), \quad (12)$$

where $\mathbf{x}(k)$ is the model output when a control input $\mathbf{u}(k)$ is applied and $\mathbf{n}(k)$ represents the effect of disturbances and modeling errors. The following expressions are valid for a WMR:

$\mathbf{y}(k) = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix}$,u(k)=$\begin{bmatrix} v(k) \\ w(k) \end{bmatrix}$ *and*

$$\mathbf{x}(k) = \begin{bmatrix} x(k-1) + Tv(k-1)\cos(\theta(k-1)) \\ y(k-1) + Tv(k-1)\sin(\theta(k-1)) \\ \theta(k-1) + T\omega(k-1) \end{bmatrix}.$$

Parameter $\mathbf{x}(k)$ can be represented in a matrix form as:

$$\mathbf{x}(k) = f[\mathbf{y}(k-1), \mathbf{u}(k-1)]. \quad (13)$$

Furthermore, disturbance $\mathbf{n}(k)$ can be modeled by:

$$\mathbf{n}(k) = \frac{1}{\Delta(q^{-1})}\mathbf{T}(q^{-1})\mathbf{e}(k), \quad (14)$$

where $\mathbf{e}(k)$ is 3×1 uncorrelated white noise vector with zero mean, and $\mathbf{T}(q^{-1})$ and $\Delta(q^{-1})$ are in the backward shift operator q^{-1} as:

$$\mathbf{T}(q^{-1}) = I_{3\times3} + T_1 q^{-1} + T_2 q^{-2} + ... + T_{nt} q^{-nt},$$
$$\Delta(q^{-1}) = 1 - q^{-1}.$$

4.2 Proposed EPSAC algorithm applied to the mobile robot

The proposed method is explained as follows:

Step 1. Initializations.

a) Definition of the relevant parameters:
- Desired reference trajectory $\mathbf{Y}_\mathrm{r}(k)$.
- Input constraints: v_{min}, v_{max}, w_{min} and w_{max}.
- Tuning parameters of the controller: N, $\mathbf{Q}$ and $\mathbf{R}$.

b) Measurement (or estimation) of the robot position $\mathbf{y}(k) = [x(k), y(k), \theta(k)]$.

Step 2. Calculate the predictions.

a) **if** $k = 1$
$\mathbf{U}_\mathrm{b}(k) = [v_i\ \omega_i\ v_i\ \omega_i \dots v_i\ \omega_i]^T$, where v_i and w_i are desired initial values of the linear and angular velocity respectively. For simplicity, they are chosen as $v_i = v_r$ and $\omega_i = 0$ in this work.
else
Use the optimized control sequence obtained in the previous sampling time, that is
$\mathbf{U}_\mathrm{b}(k) = [\mathbf{u}(k|k-1)^T, \cdots, \mathbf{u}(N-2|k-1)^T, \mathbf{u}(N-2|k-1)^T]^T$.
It is worth to notice that the last element of $\mathbf{U}_\mathrm{b}(k)$ is $\mathbf{u}(N-2|k-1)$ instead of $\mathbf{u}(N-1|k-1)$ because $\mathbf{u}(N-1|k-1)$ is not calculated at time $k-1$.

b) Calculate $\mathbf{Y}_\mathrm{b}(k)$.

c) A dynamic matrix $\mathbf{G}(k)$ is calculated by small perturbations applied to the WMR model ((10)) around $\mathbf{U}_\mathrm{b}(k)$.

Step 3. Calculation of the control action.

a) The control action can be determined by using two approaches. In the first case, the solution is obtained solving the cost function subject to its constraints by using quadratic programming methods. In the second case, aiming to reduce the computational cost, the control action is computed free from constraints by solving the optimization of analytically, that is:

$$\mathbf{U}_\mathrm{o}(k) = -\left(\mathbf{H}(k)\right)^{-1}\mathbf{b}(k). \qquad (15)$$

Next, $\mathbf{U}_\mathrm{o}(k)$ is limited using the robot constraints (also known as clipping). The controller used in the second approach in this work is so called EPSAC-Sub.

b) Calculate $\mathbf{U}(k) = \mathbf{U}_\mathrm{b}(k) + \mathbf{U}_\mathrm{o}(k)$. It is worth to mention that only the first element of $\mathbf{U}(k)$ is applied to the robot. In the next sampling period, the algorithm returns to Step 1.

Remark. The proposed approach uses some concepts related to nonlinear EPSAC (NEPSAC), where steps 2 and 3 are iterated several times in the same sampling period until $\mathbf{U}_\mathrm{o}(k)$ converges to a value close to zero. Therefore, the computed value of $\mathbf{U}(k)$ is supposed to be close enough to the optimal non-linear MPC (Keyser, 2003). Steps 2 and 3 are performed only once at each sampling period in the proposed approach, although the optimal solution of the previous sampling time $\mathbf{U}(k-1)$ is used to improve the initial "guess" of $\mathbf{U}_\mathrm{b}(k)$. Thus, the computed value of $\mathbf{U}(k)$ can be closer to the optimal nonlinear MPC.

5. SIMULATION RESULTS

This section presents simulation results for MPC-SL and EPSAC-Sub methods applied to the trajectory tracking problem in a WMR. The results obtained with EPSAC and MPC-SL-Sub controllers were omitted because they are nearly the same ones provided by EPSAC-Sub and MPC-SL, respectively. Several simulations were performed using the reference trajectory presented in Kuhne et al. (2004) considering different initial conditions.

The tuning parameters of the controllers are defined as: $N = 5$, $T = 0.1s$, $\mathbf{Q} = diag(1, 1, q)$, while two values of q have been chosen: $q = 0.5$ and $q = 0.1$, $\mathbf{R} = 0.1I_{2x2}$, the constraints that limit the control variables are $v_{max} = 0.4m/s$, $v_{min} = -0.4m/s$, $w_{max} = 0.4rad/s$ and $w_{min} = -0.4rad/s$.

The results are presented in Figures 1, 2, 3, and 4. Figure 1 corresponds to MPC-SL using four initial positions $\mathbf{y}(0)$ ([0,-1,$\frac{\pi}{2}$], [0,-3,$\frac{\pi}{2}$], [0,-1,0], and [0,-3,0]) and two values of q. It is worth to mention that $q = 0.1$ is proposed in this work, while $q = 0.5$ has been used in Kuhne et al. (2004). In addition, note that the trajectory tracking is improved by using $q = 0.1$ except for the particular case of $\mathbf{y}(0) =$ ([0,-1,$\frac{\pi}{2}$]. The same simulations tests using MPC-SL were performed with EPSAC-Sub controller. In order to establish a proper comparative analysis, the quadratic error performance index was used, which is defined as:

$$Q_\mathrm{e} = \sum_{k=0}^{N_\mathrm{s}} [x_\mathrm{r}(k) - x(k)]^2 + [y_\mathrm{r}(k) - y(k)]^2, \qquad (16)$$

where N_s is the number of samples. Of course, small values of Q_e are necessary so that the WMR trajectory can be as close as possible to the reference. The results for eight case studies are summarized in Table 1 where it can be seen that MPC-SL presents better performance only in the first case ($q = 0.5$ and $\mathbf{y}(0) = [0, -1, \frac{\pi}{2}]$). Analogously, the performance of EPSAC-Sub is improved when the proposed value $q = 0.1$ is adopted.

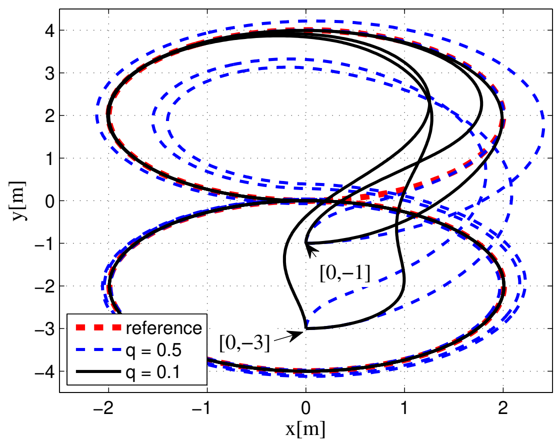

Figure 1. MPC-SL controller.

In order to show a graphical comparison between MPC-SL and EPSAC-Sub, two initial positions were selected: [0,-1,0] and [0,-3,$\frac{\pi}{2}$] with $q = 0.1$. According to Figure 2, the trajectory for EPSAC-Sub is closer to reference than that for MPC-SL. In addition, the difference between EPSAC-Sub and MPC-SL is more evident when the initial condition is farther away from the reference around which the MPC-SL model was linearized. Figure 3 shows the posture errors, which are similar except for x when $\mathbf{y}(0) = [0, -3, \frac{\pi}{2}]$, where the overshoot achieved by MPC-SL approach is about 100% higher than that for EPSAC-Sub. In addition, the highest difference between MPC-SL and EPSAC-Sub occurs with ω for $\mathbf{y}(0) = [0, -3, \frac{\pi}{2}]$.

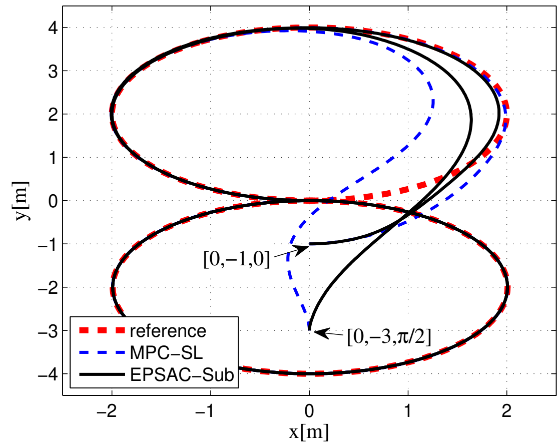

Figure 2. MPC-SL vs EPSAC-Sub: Path tracking.

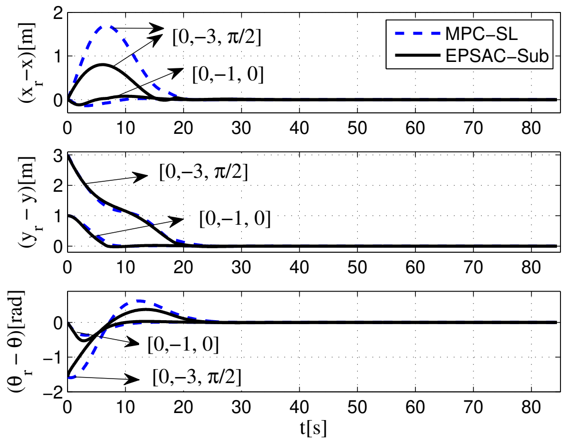

Figure 3. MPC-SL vs EPSAC-Sub: Errors.

It is important to notice that MPC-SL and EPSAC-Sub use the same cost function, while similar or identical results could be expected. However, this is only true when the initial position of the WMR is close enough to the reference. In the case of MPC-SL, the general performance is worse when the WMR is far from the reference. However, the performance of EPSAC-Sub can be improved by using an adequate base control.

Table 1. Quadratic error.

y(0)		[0,-1,$\frac{\pi}{2}$]	[0,-3,$\frac{\pi}{2}$]	[0,-1,0]	[0,-3,0]
MPC-SL	$q = 0.5$	13.13	594.75	69.77	865.41
	$q = 0.1$	24.72	591.77	29.97	645.64
EPSAC-Sub	$q = 0.5$	13.60	484.74	60.75	672.19
	$q = 0.1$	15.74	445.09	26.94	622.41

6. EXPERIMENTAL RESULTS

The wheeled mobile robot (DANi) developed by National Instruments (NI) was used in the experimental tests, so that it is possible to compare the processing time for the studied controllers in a real application. The control and processing system of the robot is based on the NI sbRIO-9632 hardware, using Real-Time programming in LabVIEW and also software LabVIEW Robotics. The posture of the robot is estimated at each sampling time by using an odometry algorithm based on internal sensors (incremental encoders). This estimation technique presents a cumulative error, while the odometer system must be updated with data from an external sensing system (video camera, laser, and others) when long trajectories are considered. However, this is not part of the paper scope.

6.1 Path tracking experiments

This subsection presents trajectory tracking experiments using the EPSAC-Sub algorithm embedded in the real robot. The trajectory reference and controller parameters are the same ones used in the simulation tests presented in Section 5. Four experiments were performed using two different initial postures, $\mathbf{y}(0) = [0, -1, \frac{\pi}{2}]$ and $\mathbf{y}(0) = [0, -3, \frac{\pi}{2}]$, and two different weights $q = 0.1$ and $q = 0.5$. Figure 4 presents the estimated positions of the robot in the XY plane and Figure 5 presents a time response of the errors ($x_r(k) - x(k)$, $y_r(k) - y(k)$, and $\theta_r(k) - \theta(k)$) for the four experiments. As can be seen the results are consistent with the simulations, that is, for the initial posture $\mathbf{y}(0) = [0, -1, \frac{\pi}{2}]$ better performance is obtained with $q = 0.5$, however, for initial posture $\mathbf{y}(0) = [0, -3, \frac{\pi}{2}]$ much better results are obtained with $q = 0.1$. Figure 6 shows the control actions. The first two correspond to the initial posture $\mathbf{y}(0) = [0, -1, \frac{\pi}{2}]$ and the two others correspond to $\mathbf{y}(0) = [0, -3, \frac{\pi}{2}]$. In addition, the presence of noise can be noticed, however, due to the low-pass characteristics of the system the noise was attenuated and not propagated significantly in the output.

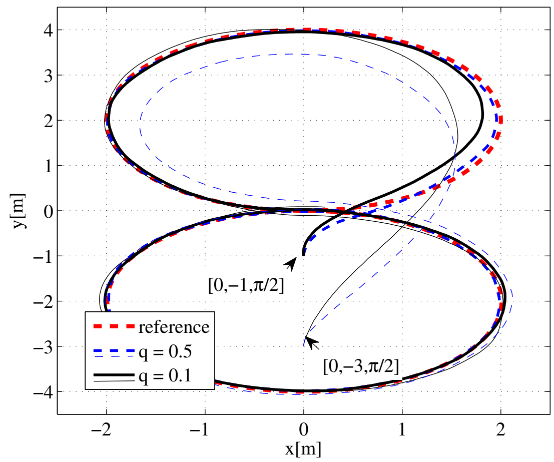

Figure 4. Experimental results of path tracking in the XY plane.

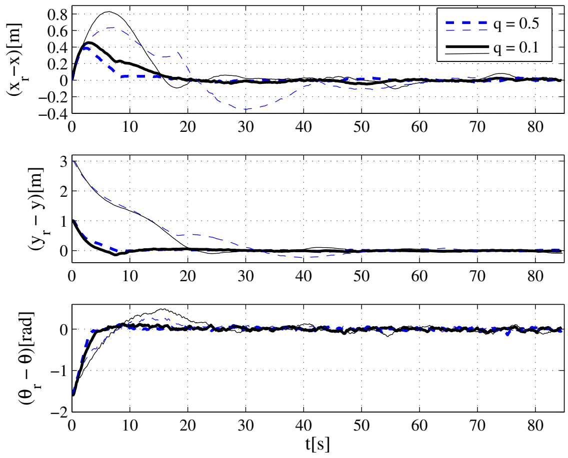

Figure 5. Errors of the robot posture.

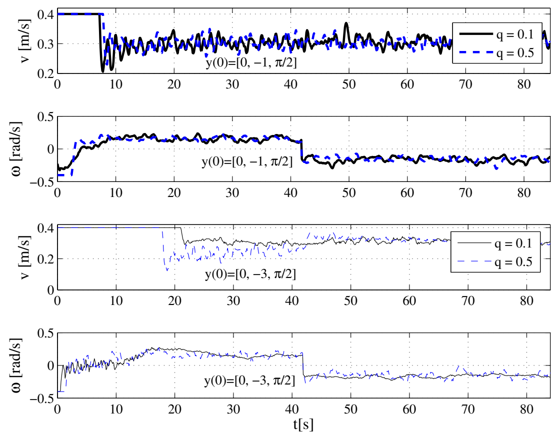

Figure 6. Control actions: v and w.

7. CONCLUSIONS

This paper has presented the design and application of EPSAC predictive control to the path tracking of WMR. Simulation results have shown improved performance of EPSAC in most cases when compared to the MPC-SL controller. The improvement of EPSAC becomes more evident when the initial posture of the WMR is far away from the reference around which MPC-SL is linearized. This is expected because the proposed EPSAC uses some concepts related to nonlinear EPSAC, which allow approximating the proposed solution to the optimal nonlinear MPC. In addition, the simulation results have shown that the performances of the constrained MPC-SL and EPSAC controllers are equivalent to the unconstrained case with clipping in the control action (also called MPC-SL-Sub and EPSAC-Sub).

In order to verify the implementation feasibility in a real robot, all controllers have been embedded in a didactic mobile robot, while processing time has been evaluated. Only the proposed EPSAC-Sub method has presented processing times lower than the sampling period, being the only feasible approach for an on-line implementation. Path tracking experiments have shown that the performance of EPSAC-Sub is similar to the one achieved simulation results even though uncertainties and measurement noise exist. Such results have effectively validated EPSAC algorithm as a prominent approach for mobile robotics applications.

REFERENCES

Barreto, J.C.L., Scolari, A.G. C., Dorea, C.E.T., Martinez, L. and Pieri, E. R.: 'Design and Implementation of Model-Predictive Control With Friction Compensation on an Omnidirectional Mobile Robot', IEEE/ASME Trans. Mechatronics, 2014, 61, (7), pp. 3671-3681.

Camacho, E. F. and Bordons, C.: 'Model Predictive Control' (Springer Verlag, 1999, 2nd edn 2004).

Campion, G., Bastin, G. and D'Andréa-Novel, B.: 'Structural properties and classification of kinematic and dynamical models of wheeled mobile robots', IEEE Trans. on Robotics and Automation, 1996, 12, (1), pp. 47-62.

Clarke, D.W.; Mohtadi, C. and Tuffs, P.S.: 'Generalized predictive control: I the basic algorithm and II Extensions and interpretations', Automatica, 1987, 23, (2), pp. 137-160.

Dai,L.; Lu, Y.; Xie H.; Sun, Z. and Xia, Y., "Robust Tracking Model Predictive Control With Quadratic Robustness Constraint for Mobile Robots With Incremental Input Constraints," in IEEE Transactions on Industrial Electronics, vol. 68, no. 10, pp. 9789-9799, Oct. 2021.

De Keyser, R.M.C. and Van Cauwenberghe, A.R.: 'Extended prediction self-adaptive control', IFAC Symp. on Identification and System Parameter Estimation, York, UK, 1985, pp. 1255-1260.

De Keyser, R.M.C.: 'A 'Gent'le Approach to Predictive Control', in E. P. C. Ltd (Ed.): 'UNESCO Encyclopedia of Life Support Systems' (Eolss Publish, 2003, 1st edn., pp. 1-30).

Subhan Khan, Jose Guivant, Xuesong Li, Design and experimental validation of a robust model predictive control for the optimal trajectory tracking of a small-scale autonomous bulldozer, Robotics and Autonomous Systems, Volume 147, 2022, 103903, ISSN 0921-8890.

Klancar, G. and Skrjanc, I.: 'Tracking-error model-based predictive control for mobile robots in real time', Int. J. of Robotics and Autonomous Systems, 2007, 55, (6), pp. 460-469.

Kuhne F.; Lages W. F. and Da Silva, J. M. G.: 'Model predictive control of a mobile robot using linearization'. Proc. IEEE Mechatronics Robot, Aachen, Germany, September 2004, pp. 525-530.

Peng Li, Hongjiu Yang, Hongbo Li, Shiqing Liang, Nonlinear ESO-based tracking control for warehouse mobile robots with detachable loads, Robotics and Autonomous Systems, Volume 149, 2022.

Raffo, G., Gomes, G., Normey-Rico, J., Kelber, C. and Becker, L.: 'A predictive controller for autonomous vehicle path tracking', IEEE Trans. on Intelligent Transportation Systems, 2009, 10, (1), pp. 92-102.

Shangming, W., Natick, M., Uthaichana, K., Zefran, M. and Decarlo, R.: 'Hybrid Model Predictive Control for the Stabilization of Wheeled Mobile Robots Subject to Wheel Slippage', IEEE Trans. on Control Syst. Technol., 2013, 21, (6), pp. 2181-2193.