A NON-LINEAR PREDICTIVE CONTROL SCHEME FOR NONHOLONOMIC MOBILE ROBOTS

Gustavo H. C. Oliveira * **José R. H. Carvalho** **

* *e-mail:* gustavoc@rla01.pucpr.br
PPGEPS/CCET/PUCPR - R. Imaculada Conceição, 1155
Curitiba-PR, Zip code 80215-901, Brazil
** *e-mail:* rcarvalho@genius.org.br

Abstract: This paper addresses the problem of closed-loop position control and trajectory tracking of nonholonomic unicycle-type mobile robots with visual-based navigation. The robot focused here has two coaxial wheels, each one driven by an independent DC-motor. The control problem is solved by a Nonlinear Model Based Predictive Controller (NLMBPC) which contains some modifications in order to deal with the system characteristics and improve its tracking behavior. Stability properties of the closed loop system are analyzed and a comparison with a linear MBPC is presented. An actual nonholonomic robot platform, the *Khepera* robot, is used to validate the proposed strategy.

Keywords: predictive control, nonholonomic mobile robots, position control, trajectory tracking, visual-based navigation, Khepera robot.

1. INTRODUCTION

Practical applications of mobile robots require an analysis of the dynamic and kinematic modeling, sensing, spatial representation of the robot and its environment, control, navigation, and path planning. In this context, the low-level control of the robot movements is quite important since any upper level task depends directly on this robot's competence.

The present work addresses the problem of closed-loop position control and path tracking for a class of nonholonomic mobile robots. In the past decade, much attention has been given to this subject and this issue has been addressed by many authors in the literature, for instance, (Samson and Ait-Abderrahim, 1990) (Walsh *et al.*, 1994) (d'Andrea Novel *et al.*, 1995), etc.

An interesting solution for this problem is the one based on MBPC strategy. In (Ortega and Camacho, 1996), the use of the MBPC framework for closed-loop tracking of a trajectory with obstacle avoidance is presented. The *Labmate* mobile robot is considered. In this work, the issue of implementing the MBPC non-linear control law is highlighted and so an approximation of the control law by using artificial neural networks is proposed. In (Normey-Rico *et al.*, 1998), an application of the MBPC scheme in a *sinchro-drive* mobile robots (*Nomad 200* platform) is described. In both works, the stopping phase of the robot is not addressed. See also (Alamir and Bornard, 1996) for a control method in the context of receding horizon control.

The present paper reviews and extends the works (Oliveira *et al.*, 1998) (Oliveira and Carvalho, 1999) (Carvalho and Oliveira, 2001) which deal with the use of MBPC strategy in mobile robots. In this context, of a nonlinear MBPC law is proposed and compared with a sub-optimal linear MBPC strategy; closed loop stability is analyzed

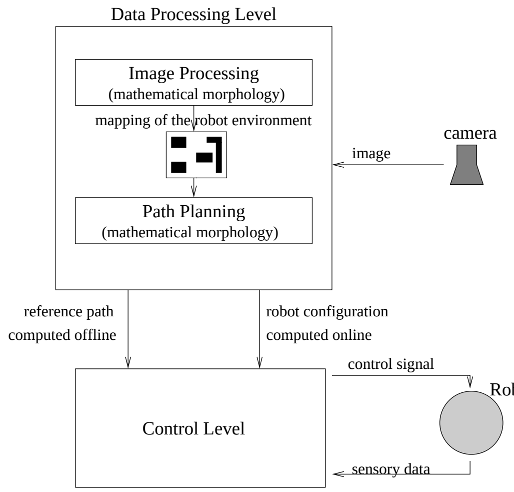

Fig. 1. Three-level structure of the visual-based navigation framework.

and a computation of the robot orientation set-point is presented in order to guarantee the convergence to a desired final position. Finally, the validation of the presented strategy by using an actual nonholonomic robot platform, the *Khepera* robot (K-Team, 1993), is performed.

2. PROBLEM STATEMENT

This work addresses the problem of navigating a mobile robot by using aerial images of its environment. For example, in the case of indoor navigation, cameras positioned under the roof can map the robot's working space and in the case of outdoor navigation, an aerial vehicle equipped with a camera can send to the robot images of the ground. In both cases, a 2D mapping, using 640×480 resolution, on a grey-scale can be used in the representation of workspace. The complete system can be structured as the tri-level framework described in Figure 1 and the relevance of such a structure system is discussed in (Carvalho *et al.*, 1999).

In this framework, the upper level is responsible for mapping the free space, detecting the presence of obstacles and returning an information corresponding of free space. The intermediate level finds the shortest path between the robot current and desired positions. The resulting trajectory is given in terms of a path (or sequence of 2D positions) to be followed by the mobile robot, which is located in the lower level. The present work is focused on a non-linear controller for the lower level of Figure 1.

If the aerial monocular imagery detects obstacles which, actually, do not exist (such as shadows or lightning disturbances), it is also possible to use robot on board range-sensors (i.e., infrared, sonars, lasers, etc.) for reducing this kind of problems, since the new measurement is orthogonal of the one given by the aerial image. The interaction between on board range sensors and the whole framework when the environment contains false obstacles is discussed in (Carvalho and Oliveira, 2001).

3. CONTROL LEVEL

The control level strategy proposed here is based on the Model Based Predictive Controller (MBPC). MBPC methods are based on the prediction of the future behavior of the process by using a model. So, in this section, a discrete-time model is obtained in order to compute the j-step ahead prediction equation. A complete description of the kinematic modelling of mobile robots can be found, for instance, in (d'Andrea Novel *et al.*, 1995) (Samson and Ait-Abderrahim, 1990). The model for the unicycle-type mobile robot used in this paper is given by:

$$\dot{X}(t) \;=\; G(\theta(t))\,U(t) \tag{1}$$

with $X(t) = \left[\, x(t)\;\; y(t)\;\; \theta(t) \,\right]^T$, $U(t) = \left[\, \dot{q}_1(t)\;\; \dot{q}_2(t) \,\right]^T$

and $G(\theta(t)) = \dfrac{r}{2}\begin{bmatrix} \cos\theta(t) & \cos\theta(t) \\ \sin\theta(t) & \sin\theta(t) \\ 1/R & -1/R \end{bmatrix}$

where $x(\cdot)$, $y(\cdot)$ and $\theta(\cdot)$ denotes the robot configuration with respect to a fixed frame. $\dot{q}_1(t)$ and $\dot{q}_2(t)$ are the angular speeds of left and right wheels, respectively. $r/2$ is the radius of the wheels and R is the wheels' half axis length. Considering a fixed sampling time denoted by δt and assuming a zero order hold, it follows that the discrete-time model is given by:

$$X(k+1) \;=\; X(k) + \Gamma(k)\,U(k) \tag{2}$$

where $k = \{k : k \in \mathbb{N}, k \geq 0\}$ denotes the samples instants. Since $\theta(t)$ is affine in relation to time between samples (due to zero order hold), the matrix $\Gamma(k)$ is given by (Oliveira and Carvalho, 1999):

$$\begin{aligned} \Gamma(k) \;&=\; \int_0^{\delta t} G(\theta(k) + \dot{\theta}(k)\tau))d\tau \\ &=\; \frac{r}{2}\begin{bmatrix} \delta\sin(k) & \delta\sin(k) \\ -\delta\cos(k) & -\delta\cos(k) \\ \delta t/R & -\delta t/R \end{bmatrix} \end{aligned} \tag{3}$$

where
$$\delta\sin(k) = \frac{\sin[\;\theta(k) + \dot{\theta}(k)\;] - \sin[\theta(k)]}{\dot{\theta}(k)}$$
$$\delta\cos(k) = \frac{\cos[\;\theta(k) + \dot{\theta}(k)\;] - \cos[\theta(k)]}{\dot{\theta}(k)}$$

By defining an operator Δ as $\Delta U(k) = U(k) - U(k-1)$, this model leads to the following j-step ahead prediction equation:

$$\begin{aligned} X(k+j|k) = X(k) + &\sum_{m=0}^{j-1}\sum_{i=m}^{j-1}\Gamma(k+i)\Delta U(k+m) \\ &+ \sum_{i=0}^{j-1}\Gamma(k+i)U(k-1) \end{aligned} \tag{4}$$

3.1 Nonlinear Model-Based Predictive Control for the Unicycle-type Mobile Robot

Model-based predictive control (MBPC) algorithms are nowadays used in many applications involving industrial processes. Usually based on linear models, MBPC methods are defined by using a model to compute the predicted process output and a cost function that describes the closed loop performance of the system; the cost function is then minimized in relation to the future control signals. Finally, the first of these control signals is applied to the process (receding horizon strategy).

Here, a nonlinear model-based predictive control (NLMBPC) law for the unicycle-type robot is derived by minimizing a quadratic cost function as follows:

$$\begin{aligned} &\Delta \boldsymbol{U}^* \;=\; \text{argmin } J(\Delta \boldsymbol{U}) \\ &\text{s. to} \quad \Delta\dot{q}_w(k+i-1) = 0 \\ &-\dot{q}_{max} \le \dot{q}_w(k+j-1) \le \dot{q}_{max} \\ &-\Delta\dot{q}_{max} \le \Delta\dot{q}_w(k+j-1) \le \Delta\dot{q}_{max} \\ &\forall\, w = 1,2;\; i = N_u,\ldots,N_y;\; \text{and } j = 1,\ldots,N_u \end{aligned} \tag{5}$$

with

$$\begin{aligned} J(\Delta \boldsymbol{U}) = &\sum_{j=N_1}^{N_y} ||X_d(k+j) - X(k+j|k)||_{\Xi} \\ &+ \sum_{j=1}^{N_u} ||\Delta U||_{\Lambda} \end{aligned} \tag{6}$$

In this optimization problem, N_1 and N_y define the output prediction horizon and N_u is the control horizon. Ξ and Λ are the weighting matrices and $||x||_A = x^TAx$. $X_d(k) = [\ x_d(k),\ y_d(k),\ \theta_d(k)\ ]^T$ is the set-point vector and $X(k+j|k)$ is the j-step ahead prediction of the robot output, which is computed using Equation (4). $\Delta\boldsymbol{U}$ is $[\ \Delta U(k)\cdots\Delta U(k+N_u-1)\ ]^T$, with $\Delta U(k+j) = [\ \Delta\dot{q}_1(k+j),\ \Delta\dot{q}_2(k+j)\ ]^T$. The set of constrains assures that the control signal is constant after the control horizon and respects the limitations in the wheels speed acceleration.

The optimal wheels' speed $U(k) = [\ \dot{q}_1(k),\ \dot{q}_2(k)\ ]^T$, applied in the process, is obtained from the optimal $\Delta\boldsymbol{U}^*$ vector computed at k, as $U(k){=}U(k-1) + \Delta U^*(k)$. In the next sampling time, all the procedure is repeated, following the classical receding horizon control strategy.

The unicycle-type robot is a non-square system, i.e., it has two input and three output signals. Thus, there are more degrees of freedom than manipulated variables and the NLMBPC method is not able to lead the robot states to a predefined configuration. To solve this problem, the orientation set-point is rewritten as a function of the position set-point, as follows: the third component of the set-point vector $X_d(k)$, i.e. $\theta_d(k)$, is computed as the angle (in the range $-\pi$ and π) between the x-axis and the segment that links the current robot position to the position set-point. In this way, one degree of freedom is eliminated since $\theta_d(k)$ is no more an independent variable, but it is a function of $x_d(k)$ and $y_d(k)$. This actually means that the orientation is not being controlled, however, this time-varying orientation set-point always tends to turn the robot in such a way that it follows the minimal path that leads to the desired position, assuring the system convergence.

3.1.1. Closed-loop system stability The case of position control, i.e., the problem of reaching a general point in the Euclidean space, is discussed now. Then, without loss of generality, it is assumed that $x_d(k)$ and $y_d(k)$ are equal to 0 for all k and the initial robot state is different from the space origin. Let us consider a state space model, with the state vector $Z(k) = \left[\ x(k),\ y(k),\ \theta_d(k) - \theta(k)\ \right]$, defined as follows:

$$Z(k+1) = Z(k) + \Gamma(k)U(k) \tag{7}$$

This system can be written as $Z(k+1) = \mathcal{F}(Z(k), U(k))$ since all signals involved in the right-hand side of this equation can be written using only the state $Z(k)$ and control $U(k)$ at the time instant k. Furthermore, the stabilization of system (7) yields the convergence of the states $x(\cdot)$ and $y(\cdot)$ of system (2) to the origin of the Euclidean space.

In this way, the cost function (6) is given by:

$$J(\Delta\boldsymbol{U}) = \sum_{j=N_1}^{N_y} ||Z(k+j|k)||_{\Xi} + \sum_{j=1}^{N_u} ||U||_{\Lambda} \tag{8}$$

Assuming $N_1 = 1$, and $N_y = N_u = N$, one obtains a receding horizon controller with infinite horizon. Many authors (Mayne and Michalska, 1990) (Rawlings and Muske, 1993) (Michalska and Mayne, 1993), and the references therein) have analyzed the stability of receding horizon control laws with quadratic cost functions, with the works of (Mayne and Michalska, 1990) and (Michalska and Mayne, 1993) being characterized by dealing with nonlinear systems. As discussed in (Michalska and Mayne, 1993), the latter can be extended to the case discrete-time case with integral control. According to (Mayne and Michalska, 1990) and (Michalska and Mayne, 1993), it follows that a nonlinear system $Z(k+1) = \mathcal{F}(Z(k), U(k))$ can be stabilized by the receding horizon control law (5) (without constraints in the input signals). In this case, the cost function (8) subject to the constraint $Z(k+N_y) = 0$ (with $N_y = N_u = N$) is used, where the parameter N is large enough to give the system enough time to move from the initial state to the origin. In (Mayne and Michalska, 1990) some assumptions were made in order to prove this closed-loop system stability. Some of them are easily verifiable in the present case, however, the complete proof is still an open problem.

4. THE QUADRATIC MODEL-BASED PREDICTIVE CONTROL (QMBPC)

In this section, it is shown that the control law given by problem (5) can be reduced to solving a quadratic program at each sampling time. This can be done by deriving an approximation of model (2) as follows.

Assuming that the sampling time or the variation of the robot orientation between two sample instants are small, it is possible to approximate the model (2) in such a way that:

$$\int_{k\delta t}^{k\delta t+\delta t} \sin(\theta(t))dt \approx \sin(\theta(k)) \int_{k\delta t}^{k\delta t+\delta t} dt$$

$$\int_{k\delta t}^{k\delta t+\delta t} \cos(\theta(t))dt \approx \cos(\theta(k)) \int_{k\delta t}^{k\delta t+\delta t} dt$$

Thus, $\Gamma(k)$ given by (3) can be approximated by:

$$\Gamma_{ap}(\theta(t)) = \delta t G(\theta(k)) = \delta t \frac{r}{2} \begin{bmatrix} \cos\theta(k) & \cos\theta(k) \\ \sin\theta(k) & \sin\theta(k) \\ 1/R & -1/R \end{bmatrix} \quad (9)$$

This leads to the following discrete-time model:

$$X(k+1) = X(k) + \Gamma_{ap}(k)U(k) \quad (10)$$

where $\Gamma_{ap}(k)$ is approximation of $\Gamma(k)$, with $\Gamma_{ap}(k) = \delta t G(\theta(k))$. By using this model, the j-step ahead prediction can be directly computed through equation (4), with the substitution of $\Gamma(\cdot)$ by $\Gamma_{ap}(\cdot)$. However, the matrix $\Gamma_{ap}(k+j)$ depends on future control actions and so the prediction equation is not affine with respect to the control signal. So, during the prediction horizon, the matrix $\Gamma_{ap}(k+j)$, for $j = N_1, \ldots, N_y$, is made equal to a constant matrix given by $\Gamma_k = \Gamma_{ap}(k)$. Therefore, the final prediction equation is given by:

$$\begin{aligned} X(k+j/k) = \sum_{i=0}^{j-1} j\Gamma_k \Delta U(k+i) \\ +X(k) + j\Gamma_k U(k-1) \end{aligned} \quad (11)$$

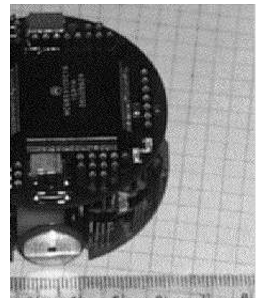

Fig. 2. The *Khepera* robot.

It is clear that the error associated with this last approximation is proportional to the prediction horizon. The larger is the prediction horizon, the larger will be this error.

In this procedure, the parameters of the model can be updated on-line using the output measurements at each sample instant. Furthermore, this nonlinear model can be viewed as a linear model with on-line parameter adaptation.

The prediction equation (11) is affine with respect to future control signal increments ($\{\Delta U(k+j-1)\}_{j=1}^{N_u}$). Therefore, the development of the cost function (6) by using equation (11) leads to a quadratic cost function such that the optimization problem (5) can be written as the following quadratic problem:

$$\begin{aligned} \min_{\Delta \boldsymbol{U}} \quad & \Delta \boldsymbol{U}^T Q \Delta \boldsymbol{U} + \boldsymbol{f}^T \Delta \boldsymbol{U} \\ \text{s. to} \quad & \mathcal{A}\, \Delta \boldsymbol{U} \le \boldsymbol{v} \end{aligned} \quad (12)$$

where Q and $\boldsymbol{f}$ are appropriate matrices, and $\mathcal{A}$ and $\boldsymbol{v}$ are built by using the information about the process constraints presented in equation (5). Thus, a single Quadratic Programming (QP) problem relative to the vector $\Delta \boldsymbol{U}$ must be solved to obtain the control law. This control law is sub-optimal with respect to the original problem. However, as it will be shown by simulation results, by computing this simpler and less time demanding implementation the closed-loop performance degradation with respect to the original solution is small.

5. EXPERIMENTAL RESULTS

In this section, the proposed MBPC method is applied to control a unicycle-type mobile robot and the hardware and software setup used in the experiments are described below.

i) Computer and robot platforms: The Computer Platform is a Sun UltraSparc 1 workstation, 167 MHz, 128 MB RAM and the Robot Platform is the mobile robot *Khepera* (Figure 2). The *Khepera*

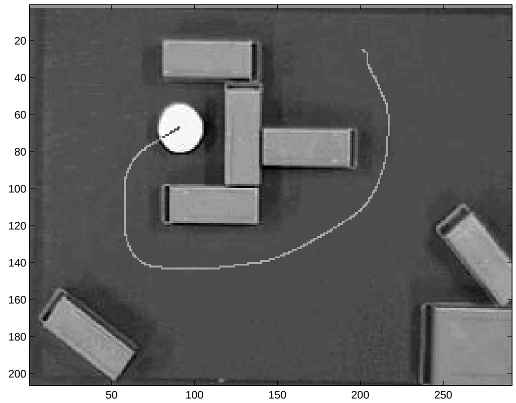

Fig. 3. The environment and the desired trajectory.

robot has wheel radius $r = 0.65\ cm$ and wheels' half axis length $R = 2.5\ cm$. The image acquisition is performed by: CCD B&W camera and Data Translation DT-3155 frame grabber.

ii) Software: MATLAB and Tcl/Tk script language. The predictive control is implemented by using MATLAB functions and scripting language. The Khepera serial communication is implemented in C language and encapsulated as MATLAB functions using C-MEX facilities. The vision based planner is implemented in Tcl/Tk based on a mathematical morphology. The frame grabber driver is implemented in C language and encapsulated as Tcl/Tk package.

The trajectory to be followed is provided by the vision-based planner and is composed by a sequence of pixels connecting initial and desired positions. Figure 3 shows an example of this procedure. In this Figure, the environment is populated by static obstacles. After computations, the planner finds the path, represented by the white line, to be followed by the robot.

5.1 Comparison between NLMBPC and QMBPC

A simulation example for comparison of the two predictive approaches in closed-loop control of the Khepera robot is presented. First, a sinusoidal trajectory is defined for the robot, i.e. $x^d(k) = k$ and $y_d(k) = 5\sin(0.2k)$. The tuning parameters for both control laws are selected as follows: $N_1 = 1$, $N_y = 3$ and $N_u = 1$. The control signal weighting Λ is set equal to a matrix of zeros in the $\mathbb{R}^{2\times 2}$ space. The matrix Ξ is initialized as an identity matrix in the $\mathbb{R}^{3\times 3}$ space, but the weight over the orientation error is a time-varying parameter, as proposed in (Oliveira and Carvalho, 1999). There are no bounds for the amplitude and increments in the control signal. The sampling time is equal to 0.1 seconds.

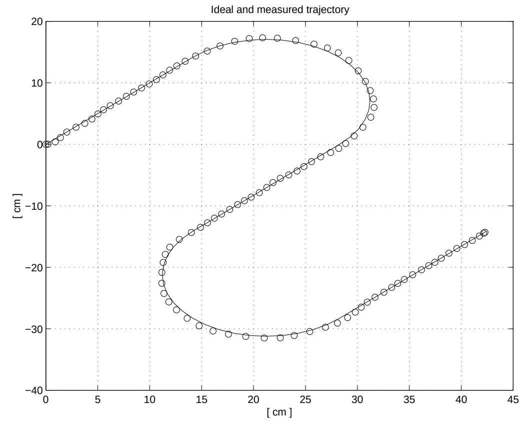

Fig. 4. The *Khepera* tracking a trajectory in the Euclidean space.

The sum of the Euclidean norms of the position and orientation errors is given by 2.39 and 2.37 for the QMBPC and NLMBPC respectively. The mean number of flops for computing the control law at each sampling time is 1095 and 21166 for the QMBPC and NLMBPC respectively. This result shows that the QMBPC represents a good approximation for the optimal NLMPBC algorithm, while requiring quite less computational effort.

5.2 Path tracking with stopping phase

The Quadratic MBPC is used in this example because it represents quite a good approximation of the optimal algorithm (the Nonlinear MBPC) and the latter cannot compute the control law rapid enough to match our requirements of sampling time, at least, for the available hadware/software setup.

In (Oliveira and Carvalho, 1999), a position control of the robot is presented and here the path tracking control case is discussed. The trajectory to be followed in this experiment is an S composed by three segments and two arcs of circumference (see Figure 4). As described before, the state of the robot is updated by using a sampling time equal to 0.5 seconds. The robot starts from $X(0) = [0,\ 0,\ 0]^T$ and the reference trajectory starts from $X_{ref} = [0,\ 0,\ \pi/4]^T$. The predictive control tuning parameters are the same as the ones of the previous example.

Figure 4 shows the reference trajectory (solid line) and the robot position sensed by the measurements (circles) at each sampling time. The speed of the robot wheels to perform this experiment are shown in Figure 5. These figures shows that the tracking errors remains within an acceptable bound and the robot is able to perform the stopping phase. This was achieved by using a simple control law that do not allows high changes in the control signal which could lead to some wheels slipping.

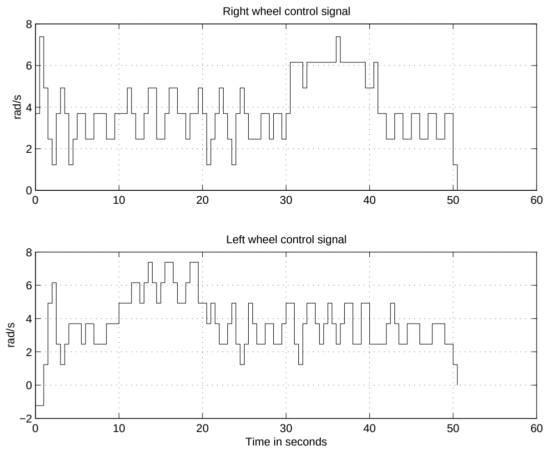

Fig. 5. The speed sent to each wheel to perform the path tracking.

6. CONCLUSIONS

In this work, the problem of closed-loop position control and path tracking for nonholonomic unicycle-type mobile robot was addressed. This type of robot is a non-linear, non-square and time-varying process.

In order to deal with the robot characteristics, the use of MBPC-type control algorithms was proposed. The characteristics of the closed-loop controllers are: an optimal nonlinear MBPC scheme and a sub-optimal QMPBC scheme with real-time updating of the robot model by using measurements, both with a new parameterization for the robot orientation set-point. The stability of the closed-loop was analyzed.

The proposed strategy presents some interesting characteristics in the context of mobile robots: *i)* it is based on a simple and well establish discrete time control design methodology; *ii)* the physical constraints in the actuators can be easily handled by the algorithm; *iii)* it is able to perform both position control and tracking of a general trajectory with stopping phase.

In order to validate the proposed strategy, a *Khepera* robot under MBPC control was used to perform the tracking of a pre-defined trajectory. It was shown that the proposed algorithm is effective for this type of problem.

REFERENCES

Alamir, M. and G. Bornard (1996). Globally stabilizing discontinuous feedback law for a class of nonlinear systems. *European Journal of Control* **2**, 44–55.

Carvalho, J. R. H. and G. H. C. Oliveira (2001). Range-sensor based trajectory tracking for mobile robots using predictive methods. In: *Proc. of the 5th World Multiconference on Systemics, Cybernatics and Informatics*. Orlando, USA.

Carvalho, J. R. H., A. Elfes, M. Bergerman, S. S. Bueno and E. C. de Paiva (1999). A methodology for aerial imagery-based ground robot navigation. In: *15th International Conference on CAD/CAM Robotics & Factories of the Future*. Águas de Lindóia, Brazil.

d'Andrea Novel, B., G. Campion and G. Bastin (1995). Control of nonholonomic wheeled mobile robots by state feedback linearization. *The International Journal of Robotics Research* **14**(6), 543–559.

K-Team (1993). *Khepera - users manual*. Laboratoire de microinformatique - Swiss Federal Institute of Technology (EPFL). INF - Ecublens, 1015 Lausanne/Switzerland.

Mayne, D. Q. and H. Michalska (1990). Receding horizon control of nonlinear systems. *IEEE Trans. on Automatic Control* **35**(7), 814–824.

Michalska, H. and D. Q. Mayne (1993). Robust receding horizon control of constrained nonlinear systems. *IEEE Trans. on Automatic Control* **38**(11), 1623–1633.

Normey-Rico, J., J. G. Ortega, I. A. Torrego and E. F. Camacho (1998). Low time-consuming implementation of predictive path-tracking control for a "syncro-drive" mobile robot. In: *Proc. of the AMC98*. Coimbra, Portugal. pp. 350–355.

Oliveira, G. H. C. and J. R. H. Carvalho (1999). Trajectory tracking for nonholonomic mobile robots using predictive methods. In: *Proc of the European Control Conference, 1999*. Germany.

Oliveira, G. H. C., J. R. H. Carvalho, J. F. Lopez, R. M. Nazzetta and G. Favier (1998). Controle de posição de robôs tipo cart via abordagem preditiva. In: 8$^{o.}$ *Latin American Congress of Automaic Control*. Viña del Mar, Chile.

Ortega, J. G. and E. F. Camacho (1996). Neural predictive control for mobile robot navigation in a partially structured static environment. In: *Proc. of 13th IFAC World Congress*. Vol. Q. San Francisco/USA. pp. 429–434.

Rawlings, J. B. and K. R. Muske (1993). The stability of constrained receding horizon control. *IEEE Transaction on Automatic Control* **38**(10), 1512–1516.

Samson, C. and K. Ait-Abderrahim (1990). Mobile robot control part 1 : Feedback control of a nonholonomic wheeled cart in cartesian space. Technical Report RR 1288. INRIA. Sophia-Antipolis.

Walsh, G., D Tilbury, S Sastry, R Murray and J. P. Laumond (1994). Stabilization of trajectory for systems with nonholonomic constraints. *IEEE Trans. on Automatic Control* **34**, 216–222.