Model Predictive Control of a Mobile Robot Using Linearization

Felipe Kühne Walter Fetter Lages João Manoel Gomes da Silva Jr.
Federal University of Rio Grande do Sul
Electrical Engineering Department
Av. Oswaldo Aranha, 103
Porto Alegre, RS 90035-190 Brazil
Email: {kuhne,fetter,jmgomes}@eletro.ufrgs.br

***Abstract*— This paper presents an optimal control scheme for a wheeled mobile robot (WMR) with nonholonomic constraints. It is well known that a WMR with nonholonomic constraints can not be feedback stabilized through continuously differentiable, time-invariant control laws. By using model predictive control (MPC), a discontinuous control law is naturally obtained. One of the main advantages of MPC is the ability to handle constraints (due to state or input limitations) in a straightforward way. Quadratic programming (QP) is used to solve a linear MPC by successive linearization of an error model of the WMR.**

I. Introduction

The field of mobile robot control has been the focus of active research in the past decades. Despite the apparent simplicity of the kinematic model of a wheeled mobile robot (WMR), the existence of nonholonomic constraints turns the design of stabilizing control laws for those systems in a considerable challenge. Due to Brockett conditions [1], a continuously differentiable, time-invariant stabilizing feedback control law can not be obtained. To overcome these limitations most works uses non-smooth and time-varying control laws [2]–[6]. Recent works dealing with robust and adaptive control of WMRs can be found in [7], [8].

However, in realistic implementations it is difficult to obtain good performance, due to the constraints on inputs or states that naturally arise. None of the previously cited works have taken those constraints into account. This can be done in a straightforward way by using model predictive control (MPC) schemes. For a WMR this is an important issue, since the position of the robot can be restricted to belong to a safe region of operation. By considering input constraints, control actions that respect actuators limits can be generated.

Furthermore, coordinate transformations of the dynamic system to *chained* or *power* forms [2] are not necessary anymore, which turns the choice of tuning parameters for the MPC more intuitive. Regarding the nonholonomic features of the WMR, a piecewise-continuous (non-smooth) control law is implicitly generated by MPC.

Model predictive control is an optimal control strategy that uses the model of the system to obtain an optimal control sequence by minimizing an objective function. At each sampling interval, the model is used to predict the behavior of the system over a prediction horizon. Based on these predictions, an objective function is minimized with respect to the future sequence of inputs, thus requiring the solution of a constrained optimization problem for each sampling interval. Although prediction and optimization are performed over a future horizon, only the values of the inputs for the current sampling interval are used and the same procedure is repeated at the next sampling time. This mechanism is known as *moving* or *receding horizon* strategy, in reference to the way in which the time window shifts forward from one sampling time to the next one.

For complex, constrained, multivariable control problems, MPC has become an accepted standard in the process industries [9]. It is used in many cases, where plants being controlled are sufficiently *slow* to allow its implementation [10]. However, for systems with fast and/or nonlinear dynamics, the implementation of such technique remains fundamentally limited in applicability, due to large amount of *on-line* computation required [11].

The model of a WMR is nonlinear. Although nonlinear model predictive control (NMPC) has been well developed [10], [12], [13], the computational effort necessary is much higher than the linear version. In NMPC there is a nonlinear programming problem to be solved on-line, which is nonconvex, has a larger number of decision variables and a global minimum is in general impossible to find [14]. In this paper, we propose a strategy to overcome at least part of these problems. The fundamental idea consists in using a successive linearization approach, as briefly outlined in [14], yielding a linear, time-varying description of the system beeing solved through linear MPC. Then, considering the control inputs as the decision variables, it is possible to transform the optimization problem in a Quadratic programming (QP) problem. Since this is a convex problem, QP problems can be solved by numerically robust solvers which lead to global optimal solutions. It is then shown that even a real-time implementation is possible. Although MPC is not a new control method, works dealing with MPC of WMRs are recent and sparse [15]–[17].

The remainder of this paper is organized as follows: in the next section the kinematic model of the WMR is shown. The MPC algorithm is depicted in section III. Simulation results in MATLAB are shown in section IV. Section V presents some

 ISBN: 3-938153-30-X

considerations regarding real-time implementation.

II. Kinematic Model of the WMR

A mobile robot made up of a rigid body and non deforming wheels is considered (see Fig. 1). It is assumed that the vehicle moves on a plane without slipping, i.e., there is a pure rolling contact between the wheels and the ground. The kinematic model of the WMR then is given by [18]:

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

or, in a more compact form as

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

where $\mathbf{x} \triangleq [x \;\; y \;\; \theta]^T$ describes the configuration (position and orientation) of the center of the axis of the wheels, C, with respect to a global inertial frame $\{O, X, Y\}$. $\mathbf{u} \triangleq [v \;\; w]^T$ is the control input, where v and w are the linear and the angular velocities, respectively.

A linear model is obtained by computing an error model with respect to a reference car. To do so, consider a reference car also described by (2). Hence, its trajectory $\mathbf{x}_r$ and $\mathbf{u}_r$ are related by:

$$\dot{\mathbf{x}}_r = f(\mathbf{x}_r, \mathbf{u}_r) \tag{3}$$

By expanding the right side of (2) in Taylor series around the point $(\mathbf{x}_r, \mathbf{u}_r)$ and discarding the high order terms it follows that

$$\dot{\mathbf{x}} = f(\mathbf{x}_r, \mathbf{u}_r) + \left.\frac{\partial f(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\mathbf{x}_r\\ \mathbf{u}=\mathbf{u}_r}} (\mathbf{x} - \mathbf{x}_r) + \\ + \left.\frac{\partial f(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}=\mathbf{x}_r\\ \mathbf{u}=\mathbf{u}_r}} (\mathbf{u} - \mathbf{u}_r), \tag{4}$$

or

$$\dot{\mathbf{x}} = f(\mathbf{x}_r, \mathbf{u}_r) + f_{\mathbf{x},r}(\mathbf{x} - \mathbf{x}_r) + f_{\mathbf{u},r}(\mathbf{u} - \mathbf{u}_r), \tag{5}$$

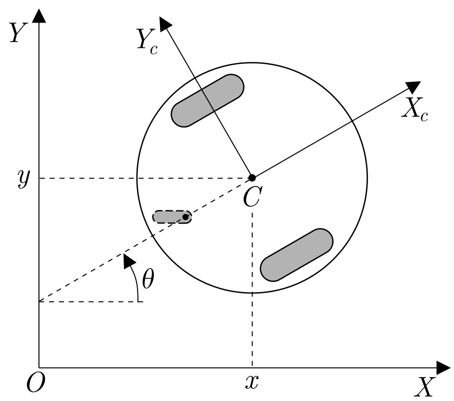

Fig. 1. Coordinate system of the WMR.

where $f_{\mathbf{x},r}$ and $f_{\mathbf{u},r}$ are the jacobians of f with respect to $\mathbf{x}$ and $\mathbf{u}$, respectively, evaluated around the reference point $(\mathbf{x}_r, \mathbf{u}_r)$.

Then, the subtraction of (3) from (5) results in:

$$\dot{\tilde{\mathbf{x}}} = f_{\mathbf{x},r}\tilde{\mathbf{x}} + f_{\mathbf{u},r}\tilde{\mathbf{u}} \tag{6}$$

Hence, $\tilde{\mathbf{x}} \triangleq \mathbf{x} - \mathbf{x}_r$ represents the error with respect to the reference car and $\tilde{\mathbf{u}} \triangleq \mathbf{u} - \mathbf{u}_r$ is its associated perturbation control input.

The approximation of $\dot{\mathbf{x}}$ by using forward differences gives the following discrete-time system model:

$$\tilde{\mathbf{x}}(k+1) = \mathbf{A}(k)\tilde{\mathbf{x}}(k) + \mathbf{B}(k)\tilde{\mathbf{u}}(k), \tag{7}$$

with

$$\mathbf{A}(k) \triangleq \begin{bmatrix} 1 & 0 & -v_r(k)\sin\theta_r(k)T \\ 0 & 1 & v_r(k)\cos\theta_r(k)T \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{B}(k) \triangleq \begin{bmatrix} \cos\theta_r(k)T & 0 \\ \sin\theta_r(k)T & 0 \\ 0 & T \end{bmatrix}$$

where T is the sampling period and k is the sampling time.

Indeed, the convergence of $\mathbf{x}$ to $\mathbf{x}_r$ is equivalent to the convergence of $\tilde{\mathbf{x}}$ to the set $\mathcal{O} = \{\mathbf{x}|(\tilde{x}, \tilde{y}, \tilde{\theta}) = (0, 0, 2\pi n)\}, n \in \{0, \pm1, \pm2, \ldots\}$.

In [2] it is shown that the nonlinear, nonholonomic system (1) is fully controllable, i.e., it can be steered from any initial state to any final state in finite time by using finite inputs. It is easy to see that when the robot is not moving, the linearization about a stationary operating point is not controllable. However, this linearization becomes controllable as long as the control input $\mathbf{u}$ is not zero [3]. This implies the tracking of a reference trajectory being possible with linear MPC [17].

III. The MPC Algorithm

It was said in section I that the essence of a MPC scheme is to optimize predictions of process behavior over a sequence of future control inputs. Such a prediction is accomplished by using a process model over a finite time interval, called the *prediction horizon*. At each sampling time, the model predictive controller generates an optimal control sequence by solving an optimization problem. The first element of this sequence is applied to the plant. The problem is solved again at the next sampling time using the updated process measurements and a shifted horizon.

For the sake of simplicity, we assume in this work that the states of the plant are always available for measurement and that there are no plant/model mismatch.

The objective function to be minimized can be stated as a quadratic function of the states and control inputs:

$$\Phi(k) = \sum_{j=1}^{N} \tilde{\mathbf{x}}^T(k+j|k)\mathbf{Q}\tilde{\mathbf{x}}(k+j|k) + \\ + \tilde{\mathbf{u}}^T(k+j-1|k)\mathbf{R}\tilde{\mathbf{u}}(k+j-1|k), \tag{8}$$

 ISBN: 3-938153-30-X

where N is the prediction horizon and $\mathbf{Q}$, $\mathbf{R}$ are weighting matrices, with $\mathbf{Q} \geq 0$ and $\mathbf{R} > 0$. The notation $a(m|n)$ indicates the value of a at the instant m predicted at instant n.

Hence, the optimization problem can be stated as to find $\tilde{\mathbf{u}}^\star$ such that:

$$\tilde{\mathbf{u}}^\star = \arg\min_{\tilde{\mathbf{u}}} \{\Phi(k)\} \tag{9}$$

The problem of minimizing (8) is solved at each time step k, yielding a sequence of optimal control $\{\tilde{\mathbf{u}}^\star(k|k), \cdots, \tilde{\mathbf{u}}^\star(k+N-1|k)\}$ and the optimal cost $\Phi^\star(k)$. The MPC control law is implicitly given by the first control action of the sequence of optimal control, $\tilde{\mathbf{u}}^\star(k|k)$. A block diagram with all components of the system is shown in Fig. 2, where the indexes $(k|k)$ are omitted.

To recast the optimization problem in a usual quadratic programming form, we introduce the following vectors:

$$\bar{\mathbf{x}}(k+1) \triangleq \begin{bmatrix} \tilde{\mathbf{x}}(k+1|k) \\ \tilde{\mathbf{x}}(k+2|k) \\ \vdots \\ \tilde{\mathbf{x}}(k+N|k) \end{bmatrix} \quad \bar{\mathbf{u}}(k) \triangleq \begin{bmatrix} \tilde{\mathbf{u}}(k|k) \\ \tilde{\mathbf{u}}(k+1|k) \\ \vdots \\ \tilde{\mathbf{u}}(k+N-1|k) \end{bmatrix}$$

Thus, (8) can be rewritten as:

$$\Phi(k) = \bar{\mathbf{x}}^T(k+1)\bar{\mathbf{Q}}\bar{\mathbf{x}}(k+1) + \bar{\mathbf{u}}^T(k)\bar{\mathbf{R}}\bar{\mathbf{u}}(k), \tag{10}$$

with

$$\bar{\mathbf{Q}} \triangleq \begin{bmatrix} \mathbf{Q} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{Q} \end{bmatrix} \quad \bar{\mathbf{R}} \triangleq \begin{bmatrix} \mathbf{R} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{R} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{R} \end{bmatrix}$$

Therefore, it is possible from (7) to write $\bar{\mathbf{x}}(k+1)$ as:

$$\bar{\mathbf{x}}(k+1) = \bar{\mathbf{A}}(k)\tilde{\mathbf{x}}(k|k) + \bar{\mathbf{B}}(k)\bar{\mathbf{u}}(k), \tag{11}$$

with

$$\bar{\mathbf{A}}(k) \triangleq \begin{bmatrix} \mathbf{A}(k|k) \\ \mathbf{A}(k|k)\mathbf{A}(k+1|k) \\ \vdots \\ \alpha(k,0) \end{bmatrix}$$

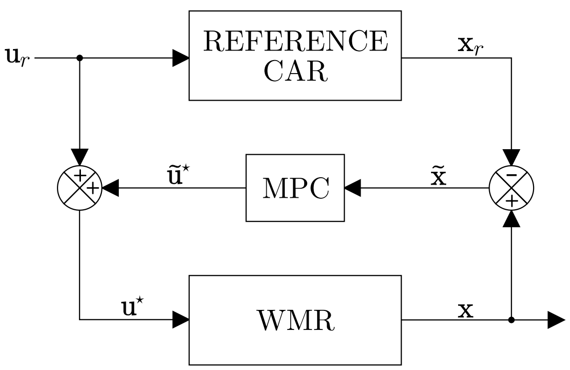

Fig. 2. Block diagram of the system.

and

$$\bar{\mathbf{B}}(k) \triangleq$$

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

where $\alpha(k,j)$ is defined as:

$$\alpha(k,j) \triangleq \prod_{i=j}^{N-1} \mathbf{A}(k+i|k),$$

¿From (10) and (11), we can rewrite the objective function (8) in a standard quadratic form:

$$\Phi(k) = \frac{1}{2}\bar{\mathbf{u}}^T(k)\mathbf{H}(k)\bar{\mathbf{u}}(k) + \mathbf{f}^T(k)\bar{\mathbf{u}}(k) + \mathbf{d}(k)$$

with

$$\begin{aligned} \mathbf{H}(k) &\triangleq 2\left(\bar{\mathbf{B}}(k)^T(k)\bar{\mathbf{Q}}\bar{\mathbf{B}}(k) + \bar{\mathbf{R}}\right) \\ \mathbf{f}(k) &\triangleq 2\bar{\mathbf{B}}^T(k)\bar{\mathbf{Q}}\bar{\mathbf{A}}(k)\tilde{\mathbf{x}}(k|k) \\ \mathbf{d}(k) &\triangleq \tilde{\mathbf{x}}^T(k|k)\bar{\mathbf{A}}^T(k)\bar{\mathbf{Q}}\bar{\mathbf{A}}(k)\tilde{\mathbf{x}}(k|k) \end{aligned}$$

The matrix $\mathbf{H}$ is a *Hessian* matrix, and must be positive definite. It describes the quadratic part of the objective function, and the vector $\mathbf{f}$ describes the linear part. $\mathbf{d}$ is independent of $\tilde{\mathbf{u}}$ and does not matter for the determination of $\mathbf{u}^\star$.

Model predictive control is based on the assumption that for a small time horizon plant and model behavior are the same. For this assumption to hold the plant/model mismatch should be kept small. Obviously, for any real world plant, control inputs are subject to physical limitations. Hence, to avoid large plant/model mismatch those limitations should be considered while computing control inputs. This can be done in a straightforward way by defining upper and lower bounds on the control input. The optimization problem must then be solved while ensuring that the control will remain between certain lower and upper bounds. Hence, the following control constraint can be written:

$$\mathbf{u}_{min}(k) \leq \mathbf{u}(k) \leq \mathbf{u}_{max}(k), \tag{12}$$

where the subscripts min and max stands for lower and upper bounds, respectively.

Hence, the optimization problem in (9) can be reformulated as to find $\tilde{\mathbf{u}}^\star$ such that:

$$\tilde{\mathbf{u}}^\star = \arg\min_{\tilde{\mathbf{u}}} \{\Phi(k)\} \tag{13}$$

s. a.

$$\mathbf{D}\tilde{\mathbf{u}} \leq \mathbf{d} \tag{14}$$

where $\Phi(k)$ is the objective function and $\tilde{\mathbf{u}}$ is the free variable in the optimization. Inequality (14) is a general way to describe constraints in the control variables. For instance, when there are just control amplitude constraints as in (12), we have

$$\begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \tilde{\mathbf{u}} \leq \begin{bmatrix} \tilde{\mathbf{u}}_{max} \\ -\tilde{\mathbf{u}}_{min} \end{bmatrix},$$

 ISBN: 3-938153-30-X

which turns out that

$$\tilde{\mathbf{u}}_{min} \leq \tilde{\mathbf{u}} \leq \tilde{\mathbf{u}}_{max}$$

Since the free variable in the optimization is $\tilde{\mathbf{u}}(k)$, the constraint (12) must be rewritten with respect to this variable:

$$\mathbf{u}_{min}(k) - \mathbf{u}_r(k) \leq \tilde{\mathbf{u}}(k) \leq \mathbf{u}_{max}(k) - \mathbf{u}_r(k),$$

or, in the vector form,

$$\bar{\mathbf{u}}_{min}(k) - \bar{\mathbf{u}}_r(k) \leq \bar{\mathbf{u}}(k) \leq \bar{\mathbf{u}}_{max}(k) - \bar{\mathbf{u}}_r(k)$$

with

$$\bar{\mathbf{u}}_{min}(k) \triangleq \begin{bmatrix} \mathbf{u}_{min}(k) \\ \mathbf{u}_{min}(k+1) \\ \vdots \\ \mathbf{u}_{min}(k+N-1) \end{bmatrix}$$

$$\bar{\mathbf{u}}_{max}(k) \triangleq \begin{bmatrix} \mathbf{u}_{max}(k) \\ \mathbf{u}_{max}(k+1) \\ \vdots \\ \mathbf{u}_{max}(k+N-1) \end{bmatrix}$$

$$\bar{\mathbf{u}}_{r}(k) \triangleq \begin{bmatrix} \mathbf{u}_{r}(k) \\ \mathbf{u}_{r}(k+1) \\ \vdots \\ \mathbf{u}_{r}(k+N-1) \end{bmatrix}$$

Since the state prediction is a function of the optimal sequence to be computed, it is easy to show that state constraints can also be described generically by (14). Furthermore, constraints in the rate of change of control and states can be formulated in a similar way.

IV. Simulation Results

In this section, simulation results are shown for the MPC applied to the WMR. The optimization problem has been solved with the MATLAB routine `quadprog`. The initial configuration of the WMR and the reference car are, respectively, $\mathbf{x}(0) = [0 \;\; -1 \;\; \pi/2]^T$ and $\mathbf{x}_r(0) = [0 \;\; 0 \;\; 0]^T$. The weighting matrices used are $\mathbf{Q} = diag(1, 1, 0.5)$ and $\mathbf{R} = 0.1\mathbf{I}_{2\times 2}$. The prediction horizon is $N = 5$. Constraints in the amplitude of the control variables are: $v_{min} = -0.4m/s$, $v_{max} = 0.4m/s$, $w_{min} = -0.4rad/s$ and $w_{max} = 0.4rad/s$.

It can be clearly seen that the state asymptotically converges to the reference. In Fig. 3 and 4, the dash-dotted line stands for the reference trajectory. Fig. 5 shows the errors of the states converging to zero. It must be noted that, even without error in the x state, the WMR must turn away from the reference trajectory due to the nonholonomic constraint. In Fig. 6, it can be seen that the control inputs are inside the limits imposed by the constraints. Since the state and control errors converge to zero (Fig. 5), one could say that the value of the objective function should also converges to zero. This fact can be observed in Fig. 7.

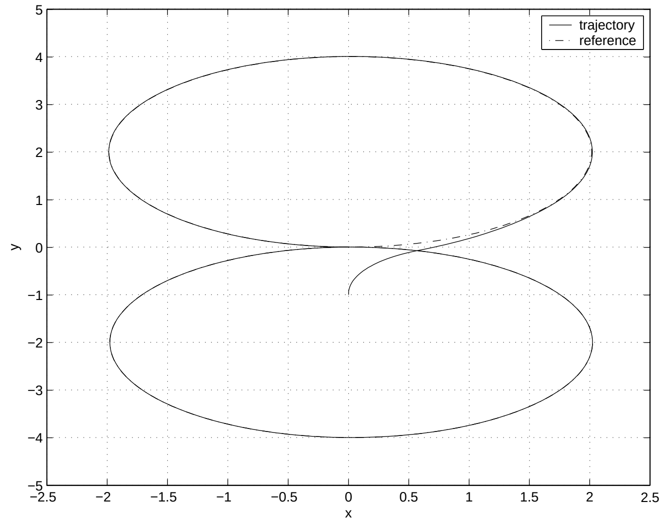

Fig. 3. Trajectory in the XY plane.

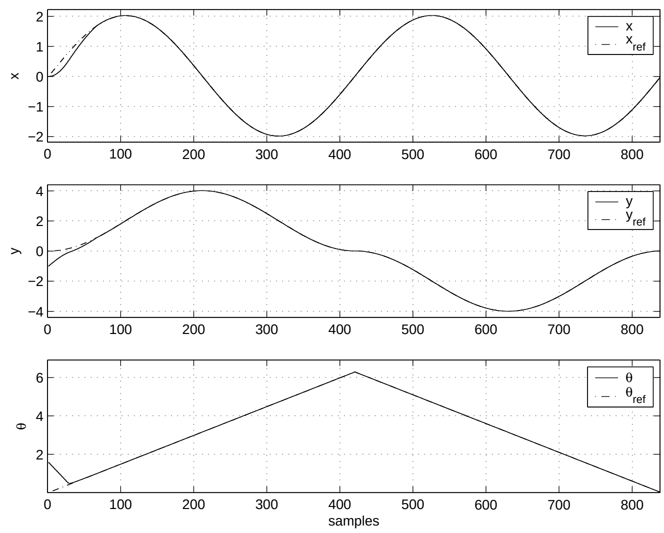

Fig. 4. States x, y and θ.

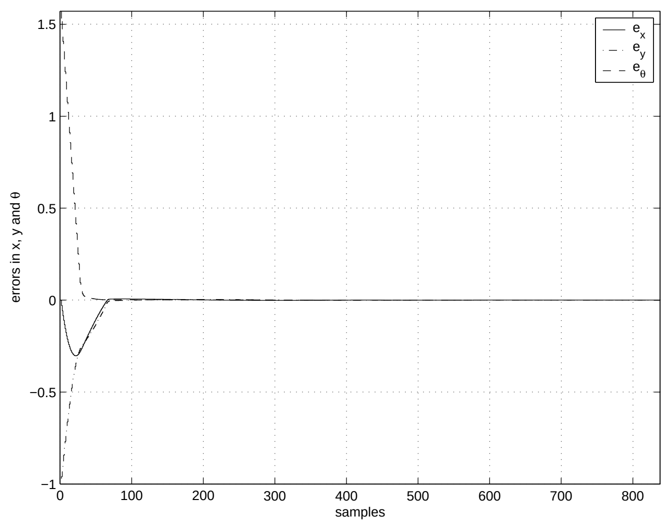

Fig. 5. Errors.

 ISBN: 3-938153-30-X

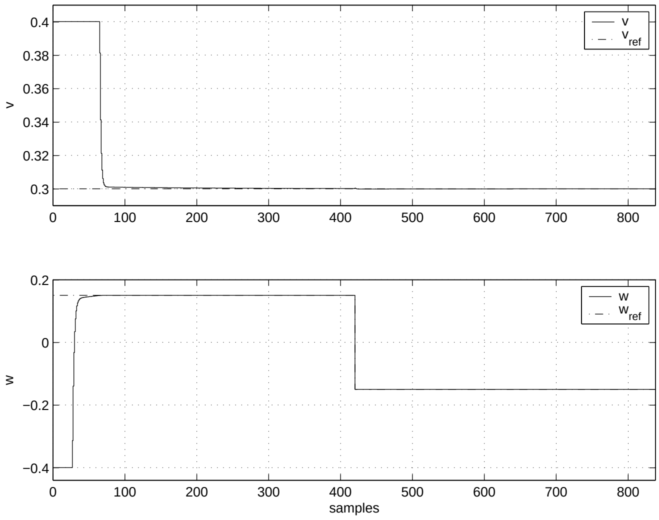

Fig. 6. Controls bounded by the constraints.

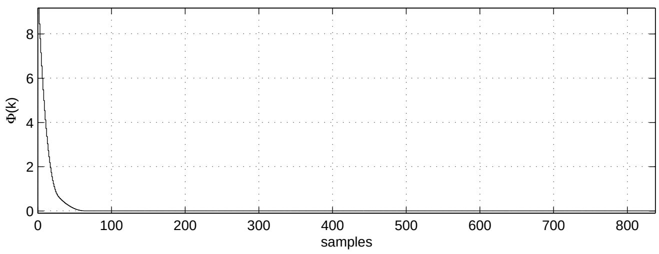

Fig. 7. Objective function $\Phi(k)$.

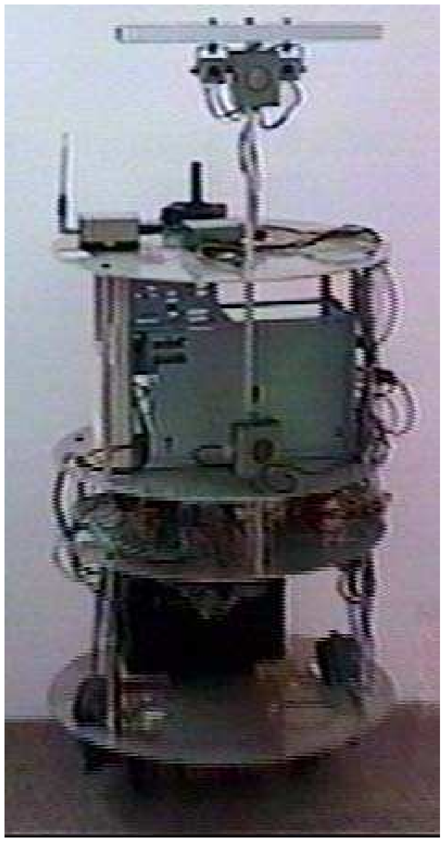

Fig. 8. The Twil Mobile Robot.

V. Real-time Experiments

Figure 8 shows the mobile robot developed in our labs and used in this work. It has a cylindrical geometry with 1.35m in height and 0.30 cm in radius and uses a differential-drive steering. The software is based on a real-time variation of the Linux operating system called RTAI [19].

The use of MPC for real-time control of systems with fast dynamics such as a WMR has been hindered for some time due to its numerical intensive nature [11]. However, with the development of increasingly faster processors the use of MPC in demanding applications becomes possible. Indeed, the data in Table I provides enough evidence that a standard of the shelf computer is able to run a MPC based controller for a WMR. An Athlon XP 2600+ gives an peak performance between 576 and 1100 Mflops using double precision computations accordingly to [20], a de-facto standard for floating point performance measurement. Therefore, the MPC algorithm proposed here could be computed for $N = 30$ in about 10ms, while the dynamics of the mobile robot used here is such that sampling periods between 50 and 100 ms are adequate, revealing that a real-time implementation is plenty possible.

A typical measure of convergence is the integrated error [17],

$$\varepsilon \triangleq \frac{1}{K_f} \sum_{k=0}^{K_f} \|\tilde{\mathbf{x}}\|^2 ,$$

where K_f is the number of steps to cover all the trajectory and $\|\cdot\|$ is the euclidian norm.

Table I shows some results relating the computational cost and the error ε as a function of the prediction horizon N. The computing time is the mean time to solve the optimization problem for one step of the trajectory. The number of flops needed to complete the calculations due at each sampling time is also shown.

TABLE I

Influence of prediction horizon in computing time and ε

Horizon	Computing time (s)	Flops	ε
1	0.0110	4343	3.2578
3	0.0114	9529	1.4384
5	0.0135	25643	1.3757
10	0.0271	160180	1.3695
15	0.0582	528570	1.3798
20	0.1156	1269500	1.3927
30	0.3402	4949000	1.4856

Obviously the prediction horizon must be chosen in a way such that the computing time is smaller than the sampling period. Here, $T = 0.1s$, therefore with $N = 20$ or above the MPC is not feasible. On the other hand, by increasing N

 ISBN: 3-938153-30-X

above 5 there is not sensible improvement on ε. Furthermore, for $N = 5$ the computing time is approximately seven times smaller than the sampling period. Hence, five steps ahead is a good choice for prediction horizon. The computations were carried out on a computer with an Athlon XP 2600+ processor running Linux operating system.

VI. Conclusion

This paper presented an application of MPC to the problem of trajectory tracking of a nonholonomic WMR. The solution of the optimization problem through a standard QP method was shown. The obtained control signals were such that the constraints imposed on the control variables were respected.

As shown above, the choice of MPC for the application given here is well justified by some advantages: the straightforward way in which state/input constraints can be handled; coordinate transformations to a chained or power form are not necessary; the MPC implicitly generates a piecewise-continuous control law, thus dealing with Brockett conditions.

It was clearly shown that with a successive linearization approach, the optimal control problem was successfully solved, arising the possibility of a real time implementation. With such technique, it was possible the transformation of the optimization problem in a standard QP formulation, which is fast and extremely robust numerically.

Acknowledgments

The authors gratefully acknowledge the financial support from CAPES.

References

[1] R. W. Brockett, *New directions in applied mathematics.* Springer-Verlag, New York, 1982.

[2] A. M. Bloch and N. H. McClamroch, "Control of mechanical systems with classical nonholonomic constraint," in *American Conference on Decision and Control*, Tampa, Florida, 1989, pp. 201–205.

[3] C. Samson and K. Ait-Abderrahim, "Feedback control of a nonholonomic wheeled cart in cartesian space," in *IEEE Int. Conf. on Robotics and Automation*, Sacramento, California, 1991, pp. 1136–1141.

[4] C. Canudas de Wit and O. J. Sørdalen, "Exponential stabilization of mobile robots with nonholonomic constraints," *IEEE Trans. Automatic Control*, vol. 37(11), pp. 1791–1797, Nov 1992.

[5] Y. Yamamoto and X. Yun, "Coordinating locomotion and manipulation of a mobile manipulator," *IEEE Trans. Automatic Control*, vol. 39(6), pp. 1326–1332, Jun 1994.

[6] R. T. McCloskey and R. M. Murray, "Exponential stabilization of driftless control systems using homogeneous feedback," *IEEE Trans. Automatic Control*, vol. 42(5), pp. 614–628, May 1997.

[7] M. Oya, C. Y. Su, and R. Katoh, "Robust adaptive motion/force tracking control of uncertain nonholonomic mechanical systems," *IEEE Trans. Robotics and Automation*, vol. 19(1), pp. 175–181, Jun 2003.

[8] W. E. Dixon, M. S. de Queiroz, D. M. Dawson, and T. J. Flynn, "Adaptive tracking and regulation of a wheeled mobile robot with controller/update law modularity," *IEEE Trans. Control Systems Technology*, vol. 12(1), pp. 138–147, Jan 2004.

[9] A. Bemporad, F. Borrelli, and M. Morari, "Model predictive control based on linear programming – The explicit solution," *IEEE Trans. Automatic Control*, vol. 47(12), pp. 1974–1985, Dec 2002.

[10] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: stability and optimality," *Automatica*, vol. 36(6), pp. 789–814, Jun 2000.

[11] M. Cannon and B. Kouvaritakis, "Continuous-time predictive control of constrained nonlinear systems," in *Nonlinear model predictive control, Progress in systems and control theory*, F. Allgöwer and A. Zeng, Eds. Basel/Switzerland: Birkhäuser Verlag, 2000, vol. 26, pp. 205–215.

[12] F. Allgower, T. A. Badgwell, J. S. Qin, J. B. Rawlings, and S. J. Wright, "Nonlinear predictive control and moving horizon estimation – an introductory overview," in *Advances In Control: Highlights of ECC'99*, P. M. Frank, Ed. New York: Springer-Verlag, 1999, pp. 391–449.

[13] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34(10), pp. 1205–1217, Oct 1998.

[14] M. A. Henson, "Nonlinear model predictive control: current status and future directions," *Computers and Chemical Engineering*, vol. 23(2), pp. 187–202, Dec 1998.

[15] A. Ollero and O. Amidi, "Predictive path tracking of mobile robots. Application to the CMU Navlab," in *5th. IEEE Int. Conf. on Advanced Robotics*, Pisa, Italy, 1991, pp. 1081–1086.

[16] J. E. Normey-Rico, J. Gómez-Ortega, and E. F. Camacho, "A Smith-predictor-based generalised predictive controller for mobile robot path-tracking," *Control Engineering Practice*, vol. 7(6), pp. 729–740, Jun 1999.

[17] H. V. Essen and H. Nijmeijer, "Non-linear model predictive control of constrained mobile robots," in *European Control Conference*, Porto, Portugal, 2001, pp. 1157–1162.

[18] G. Campion, G. Bastin, and B. D'Andréa-Novel, "Structural properties and classification of kinematic and dynamical models of wheeled mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 1, pp. 47–62, Feb 1996.

[19] L. Dozio and P. Mantegazza, "Linux real time application interface (RTAI) in low cost high performance motion control," in *Proceedings of the Motion Control Conference.* Milano, Italy: Associazione Nazionale Italiana per l'Automazione, Mar 2003.

[20] A. Aburto, "FLOPS C program (double precision) v2.0 18 dec 1992," Dec 1992. [Online]. Available: ftp://ftp.nosc.mil/pub/aburto/flops