
MOBILE ROBOT TRAJECTORY TRACKING USING MODEL PREDICTIVE CONTROL

Felipe Kühne*

kuhne@ece.ufrgs.br

João Manoel Gomes da Silva Jr.*

jmgomes@ece.ufrgs.br

Walter Fetter Lages*

fetter@ece.ufrgs.br

*Universidade Federal do Rio Grande do Sul
Departamento de Engenharia Elétrica
Av. Osvaldo Aranha, 103 – CEP 90035-190
Porto Alegre, RS, Brasil

ABSTRACT

This work focus on the application of model-based predictive control (MPC) to the trajectory tracking problem of nonholonomic wheeled mobile robots (WMR). The main motivation of the use of MPC in this case relies on its ability in considering, in a straightforward way, control and state constraints that naturally arise in practical problems. Furthermore, MPC techniques consider an explicit performance criterion to be minimized during the computation of the control law. The trajectory tracking problem is solved using two approaches: (1) nonlinear MPC and (2) linear MPC. Simulation results are provided in order to show the effectiveness of both schemes. Considerations regarding the computational effort of the MPC are developed with the purpose of analyzing the real-time implementation viability of the proposed techniques.

KEYWORDS: Mobile robots, trajectory tracking, nonholonomic systems, model-based predictive control.

1 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 design of stabilizing control laws for those systems can be considered a challenge due to the existence of nonholonomic (non-integrable) constraints. Due to Brockett's conditions (Brockett, 1982), a smooth, time-invariant, static state feedback control law cannot be used to stabilize a nonholonomic system at a given configuration. To overcome this limitation most works use non-smooth and time-varying control laws (Bloch e McClamroch, 1989; Samson e Ait-Abderrahim, 1991; Canudas de Wit e Sørtdalen, 1992; Yamamoto e Yun, 1994; McCloskey e Murray, 1997). On the other hand, this limitation can be avoided when the objective is the tracking of a pre-computed trajectory (Kanayama et al., 1990; Pomet

et al., 1992; Yang e Kim, 1999; Do et al., 2002; Sun, 2005).

Traditional techniques for the control of nonholonomic WMRs often do not present good results, due to constraints on inputs or states that naturally arise. Also, in general, the resulting closed-loop trajectory presents undesirable oscillatory motions. Furthermore, tuning parameters are difficult to choose in order to achieve good performance since the control laws are not intuitively obtained. Model-based predictive control (MPC) appears therefore as an interesting and promising approach for overcoming the problems above mentioned. In particular, by using MPC it follows that: the tuning parameters are directly related to a cost function which is minimized in order to obtain an optimal control sequence; constraints on state and control inputs can be considered in a straightforward way. Thus, control actions that respect actuators limits are automatically generated. By considering state constraints, the configuration of the robot can be restricted to belong to a safe region (Kühne, 2005). On the other hand, the main drawback of MPC schemes is related to its computational burden which, in the past years, had limited its applications only to sufficient slow dynamic systems. However, with the development of increasingly faster processors and efficient numerical algorithms, the use of MPC in faster applications, which is the case of WMRs, becomes possible.

Although MPC is not a new control method, works dealing with MPC of WMRs are sparse. In (Ollero e Amidi, 1991; Normey-Rico et al., 1999), GPC (Generalized Predictive Control) is used to solve the path following problem. In that work, it is supposed that the control acts only in the angular velocity, while the linear velocity is constant. Hence, an input-output linear model is used to compute the distance between the robot and a reference path. Note that differently from the trajectory tracking problem, in the path following the reference is not time-parameterized. In (Gómez-Ortega e Camacho, 1996) a nonlinear model of the WMR is used for trajectory tracking. The problem is solved considering unknown obstacles in the configuration space. A

neural network helps to solve the optimization problem. Also, in (Yang et al., 1998) the path following problem is solved by using a neural network to predict the future behavior of a car-like WMR. The modelling errors are corrected on-line with the neural network model. Using a nonlinear model of the robot, in (Essen e Nijmeijer, 2001) a nonlinear MPC (NMPC) algorithm in state-space representation is developed, which is applied to both problems of point stabilization and trajectory tracking. A modified cost function to be minimized is proposed. Accordingly to the authors, the MPC developed in that work can not be applied in real-time, given the high computational effort necessary in the optimization problem.

In this paper, we are interested in the application of MPC schemes to control a WMR in the problem of trajectory tracking. Two approaches based on state-space representation of the kinematic model of a nonholonomic WMR are developed. First, a nonlinear MPC (NMPC) is developed, which leads to a non-convex optimization problem. In a second approach, a linear technique is proposed to overcome the problem related to the computational burden of the NMPC. The fundamental idea consists in using a successive linearization approach, as briefly outlined in (Henson, 1998), yielding a linear, time-varying description of the system. From this linear description, the optimization problem to be solved at each sampling period is a quadratic program (QP) one, which is convex and computationally less expensive than the optimization problem that arises in the classical NMPC.

Finally, analysis regarding the computational effort are carried out in order to evaluate the real-time implementability of the MPC strategies proposed here.

2 PROBLEM FORMULATION

A mobile robot made up of a rigid body and non deforming wheels is considered (Fig. 1). It is assumed that the vehicle moves without slipping on a plane, i.e., there is a pure rolling contact between the wheels and the ground. The kinematic model of the WMR is then given by (Campion et al., 1996):

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

or, in a more compact form as

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

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

As described in the next sections, in MPC a prediction model is used and the control law is computed in discrete-time. Thus, a discrete-time representation of this model becomes necessary. Considering a sampling period T , a sampling instant k and applying the Euler's approximation to (1), we obtain the

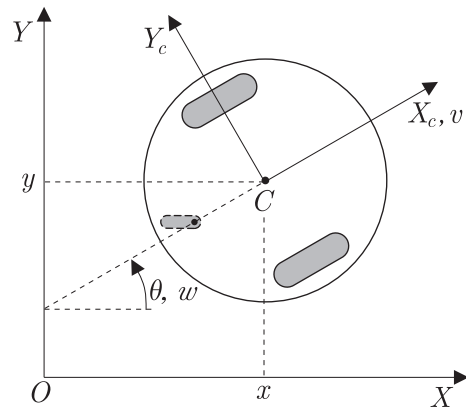


Figure 1: Coordinate system of the WMR.

following discrete-time model for the robot motion:

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

or, in a compact representation,

$$\mathbf{x}(k+1) = f_d(\mathbf{x}(k), \mathbf{u}(k)) \quad (3)$$

The problem of trajectory tracking can be stated as to find a control law such that

$$\mathbf{x}(k) - \mathbf{x}_r(k) = 0$$

where \mathbf{x}_r is a known, pre-specified reference trajectory. It is usual in this case to associate to this reference trajectory a *virtual* reference robot, which has the same model than the robot to be controlled. Thus, we have:

$$\dot{\mathbf{x}}_r = f(\mathbf{x}_r, \mathbf{u}_r), \quad (4)$$

or, in discrete-time, $\mathbf{x}_r(k+1) = f_d(\mathbf{x}_r(k), \mathbf{u}_r(k))$.

3 THE NONLINEAR MPC APPROACH

In this section, the trajectory tracking problem is solved with a NMPC strategy. In order to evaluate the performance of the approach, simulation results are shown.

MPC 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 instant, the model is used to predict the behavior of the system over a prediction horizon. Based on these predictions, the objective function is minimized with respect to the future sequence of inputs, thus requiring the solution of a constrained optimization problem for each sampling instant. Although prediction and optimization are performed over a future horizon, only the values of the inputs for the current sampling interval are used. The same procedure is repeated at the next sampling instant with updated process measurements and a shifted horizon. 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 instant to the next one (Camacho e Bordons, 1999; Allgower et al., 1999).

From (3), the prediction of the robot motion is obtained as follows:

$$\mathbf{x}(k+j+1|k) = f_d(\mathbf{x}(k+j|k), \mathbf{u}(k+j|k)),$$

where $j \in [0, N-1]$ and the notation $a(m|n)$ indicates the value of a at the instant m predicted at instant n . By defining error vectors $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_r$ and $\tilde{\mathbf{u}} = \mathbf{u} - \mathbf{u}_r$, we can formulate the following objective function to be minimized:

$$\Phi(k) = \sum_{j=1}^N \tilde{\mathbf{x}}^T(k+j|k) \mathbf{Q} \tilde{\mathbf{x}}(k+j|k) + \tilde{\mathbf{u}}^T(k+j-1|k) \tilde{\mathbf{R}} \tilde{\mathbf{u}}(k+j-1|k), \quad (5)$$

where N is the prediction horizon and $\mathbf{Q} \geq 0$, $\mathbf{R} > 0$ are weighting matrices for the error in the state and control variables, respectively.

We consider also the existence of bounds in the amplitude of the control variables:

$$\mathbf{u}_{min} \leq \mathbf{u}(k+j|k) \leq \mathbf{u}_{max}, \quad (6)$$

where \mathbf{u}_{min} stands for the lower bound and \mathbf{u}_{max} stands for the upper bound¹. Note that we can write (6) in a more general form as

$$\mathbf{D}\mathbf{u}(k+j|k) \leq \mathbf{d}$$

with:

$$\mathbf{D} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} \mathbf{u}_{max} \\ -\mathbf{u}_{min} \end{bmatrix},$$

and, in a similar way, state constraints can be generically expressed by $\mathbf{C}\mathbf{x}(k+j|k) \leq \mathbf{c}$.

Hence, the nonlinear optimization problem can be stated as:

$$\mathbf{x}^*, \mathbf{u}^* = \arg \min_{\mathbf{x}, \mathbf{u}} \{\Phi(k)\} \quad (7)$$

s. a.

$$\mathbf{x}(k|k) = \mathbf{x}_0, \quad (8)$$

$$\mathbf{x}(k+j+1|k) = f_d(\mathbf{x}(k+j|k), \mathbf{u}(k+j|k)), \quad (9)$$

$$\mathbf{D}\mathbf{u}(k+j|k) \leq \mathbf{d}, \quad (10)$$

where $j \in [0, N-1]$. \mathbf{x}_0 is the initial condition which corresponds to the value of the states measured at the current instant. Constraint (9) represents the prediction model and (10) is the control constraint which may be present or not in the optimization problem. Notice that the decision variables are both state and control variables.

The optimization problem (7)–(10) must therefore be solved at each sampling time k , yielding a sequence of optimal states $\{\mathbf{x}^*(k+1|k), \dots, \mathbf{x}^*(k+N|k)\}$, optimal controls $\{\mathbf{u}^*(k|k), \dots, \mathbf{u}^*(k+N-1|k)\}$ and the optimal cost $\Phi^*(k)$. The MPC control law is implicitly given by the first control action of the sequence of optimal control, $\mathbf{u}^*(k|k)$, and the remaining portion of this sequence is discarded.

As a case study, let us consider the robot Twil (Lages, 1998), which have the following limits in the amplitude of the control variables (Kühne, 2005):

$$\mathbf{u}_{max} = -\mathbf{u}_{min} = \begin{bmatrix} 0.47 \text{ m/s} \\ 3.77 \text{ rad/s} \end{bmatrix} \quad (11)$$

¹The symbol \leq stands for componentwise inequalities in this case.

Considering the robot model described by (1), the optimization problem (7)–(10) is used to solve the problem of trajectory tracking², with a cost function in the form of (5). Using a reference trajectory in "U" form and with the tuning parameters: $N = 5$, $\mathbf{Q} = \text{diag}(1; 1; 0, 5)$, $\mathbf{R} = \text{diag}(0, 1; 0, 1)$ and with an initial condition of $\mathbf{x}_0 = [-1 \ -1 \ 0]^T$, we have the simulation results shown in Figures 2 and 3.

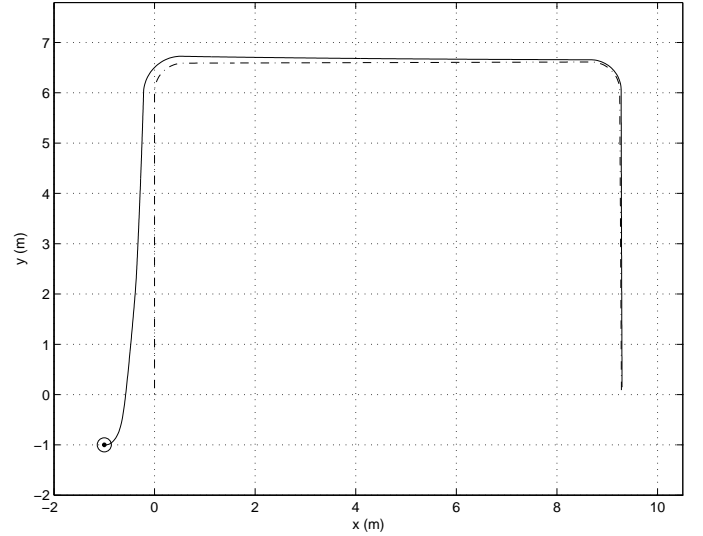


Figure 2: Trajectory in the XY plane.

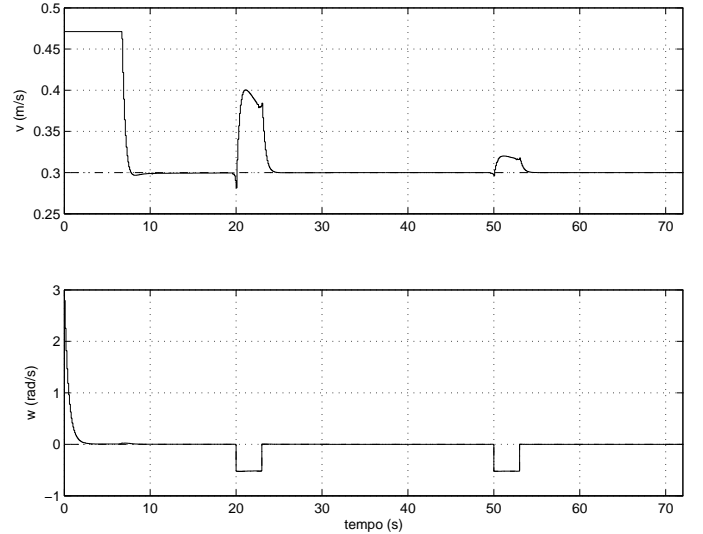


Figure 3: Controls inputs.

It can be noted in Fig. 2 that the problem is successfully solved, but with low convergence rate. Fig. 3 shows that the generated control signals respect the imposed constraints. The dash-dotted lines stands for the reference trajectories.

In the attempt of increasing the convergence rate, (Essen e Nijmeijer, 2001) proposed a modified cost function in order to increase the state penalty over the horizon, thus forcing the states to converge faster. Hence, the idea of exponentially increasing state weighting has been introduced. Also, a terminal state cost has been added to the cost function to be minimized. From these

²All the optimization problems in this section has been solved with the MATLAB routine `fmincon`.

modifications the cost function assumes therefore the following form:

$$\Phi(k) = \sum_{j=1}^{N-1} \tilde{\mathbf{x}}^T(k+j|k) \mathbf{Q}(j) \tilde{\mathbf{x}}(k+j|k) + \sum_{j=0}^{N-1} \tilde{\mathbf{u}}^T(k+j|k) \mathbf{R} \tilde{\mathbf{u}}(k+j|k) + \Omega(\tilde{\mathbf{x}}(k+N|k)), \quad (12)$$

where $\mathbf{Q}(j) = 2^{j-1} \mathbf{Q}$ and $\Omega(\tilde{\mathbf{x}}(k+N|k)) = \tilde{\mathbf{x}}^T(k+N|k) \mathbf{P} \tilde{\mathbf{x}}(k+N|k)$, with $\mathbf{P} \geq 0$, is the terminal state cost.

Thus, by using the same conditions of the first case with $\mathbf{P} = 30\mathbf{Q}(N)$, we have the results shown in Figures 4 and 5.

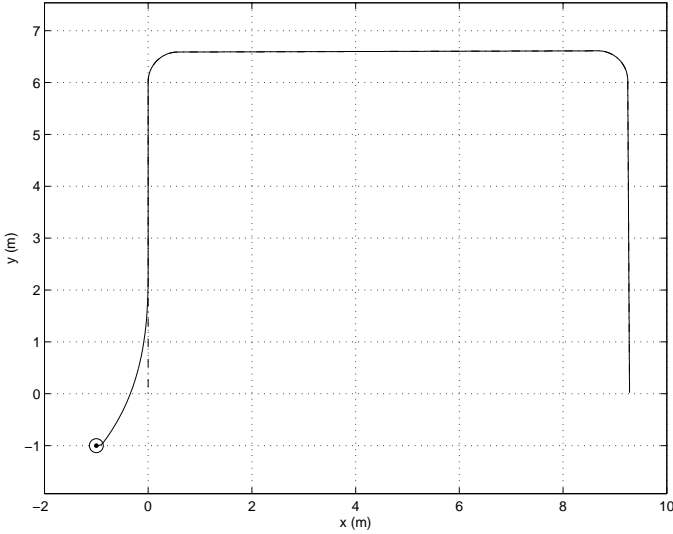


Figure 4: Trajectory in the XY plane.

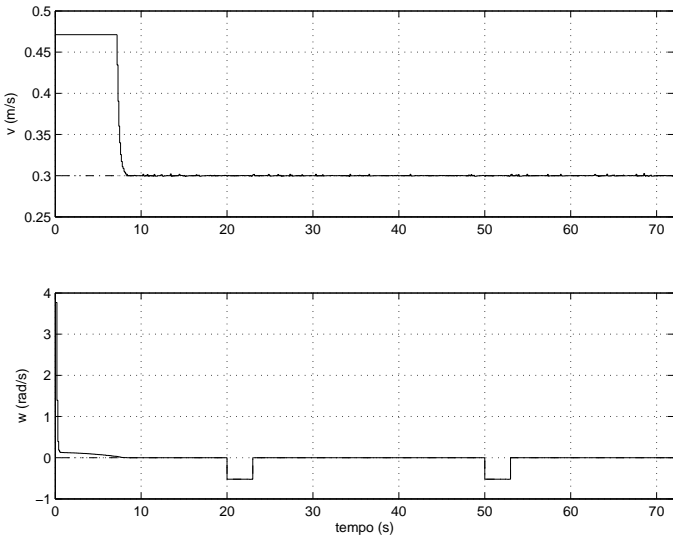


Figure 5: Controls inputs.

Thus, comparing Figures 2 and 4, it can be clearly seen that the robot presents a higher convergence rate, and Fig. 5 shows that the control signals respect the imposed constraints.

4 THE LINEAR MPC APPROACH

In this section we introduce a linear MPC (LMPC) scheme applied to the problem of trajectory tracking.

Although many NMPC techniques has been proposed in the literature (Chen e Allgöwer, 1998; Allgower et al., 1999), it should be noticed that the computational effort necessary in this case is much higher than in 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 (Henson, 1998). In this section, we propose a strategy in order to reduce the computational burden. The fundamental idea consists in using a successive linearization approach, yielding a linear, time-varying description of the system. Then, considering the control inputs as the decision variables, it is possible to transform the optimization problem to be solved at each sampling time in a QP problem (Kühne, 2005). Since they are convex, QP problems can be easily solved by numerically robust algorithms which lead to global optimal solutions.

A linear model of the system dynamics can be obtained by computing an error model with respect to a reference car. 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:

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

or,

$$\dot{\mathbf{x}} = f(\mathbf{x}_r, \mathbf{u}_r) + f_{\mathbf{x},r}(\mathbf{x} - \mathbf{x}_r) + f_{\mathbf{u},r}(\mathbf{u} - \mathbf{u}_r), \quad (13)$$

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 (4) from (13) results in:

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

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

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

$$\tilde{\mathbf{x}}(k+1) = \mathbf{A}(k) \tilde{\mathbf{x}}(k) + \mathbf{B}(k) \tilde{\mathbf{u}}(k), \quad (14)$$

with

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

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

In (Bloch e McClamroch, 1989) 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 by using finite inputs.

$$\bar{\mathbf{A}}(k) = \begin{bmatrix} \mathbf{A}(k|k) \\ \mathbf{A}(k+1|k)\mathbf{A}(k|k) \\ \vdots \\ \alpha(k, 2, 0) \\ \alpha(k, 1, 0) \end{bmatrix} \quad \bar{\mathbf{B}}(k) = \begin{bmatrix} \mathbf{B}(k|k) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}(k+1|k)\mathbf{B}(k|k) & \mathbf{B}(k+1|k) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha(k, 2, 1)\mathbf{B}(k|k) & \alpha(k, 2, 2)\mathbf{B}(k+1|k) & \cdots & \mathbf{0} \\ \alpha(k, 1, 1)\mathbf{B}(k|k) & \alpha(k, 1, 2)\mathbf{B}(k+1|k) & \cdots & \mathbf{B}(k+N-1|k) \end{bmatrix} \quad (15)$$

On the other hand, it is easy to see that when the robot is not moving (i.e., $v_r = 0$), the linearization around a stationary operating point is not controllable. However, this linearization becomes controllable as long as the control input \mathbf{u} is not zero (Samson e Ait-Abderrahim, 1991). This implies that the tracking of a reference trajectory is possible with linear MPC.

Now it is possible to recast the optimization problem in an usual quadratic programming form. Hence, we introduce the following vectors:

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

Thus, with $\bar{\mathbf{Q}} = \text{diag}(\mathbf{Q}; \dots; \mathbf{Q})$ and $\bar{\mathbf{R}} = \text{diag}(\mathbf{R}; \dots; \mathbf{R})$, the cost function (5) can be rewritten as:

$$\Phi(k) = \bar{\mathbf{x}}^T(k+1)\bar{\mathbf{Q}}\bar{\mathbf{x}}(k+1) + \bar{\mathbf{u}}^T(k)\bar{\mathbf{R}}\bar{\mathbf{u}}(k), \quad (16)$$

Therefore, it is possible from (14) to write $\bar{\mathbf{x}}(k+1)$ as (Kühne, 2005):

$$\bar{\mathbf{x}}(k+1) = \bar{\mathbf{A}}(k)\bar{\mathbf{x}}(k|k) + \bar{\mathbf{B}}(k)\bar{\mathbf{u}}(k), \quad (17)$$

where $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ are defined in (15) with $\alpha(k, j, l)$ given by:

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

From (16), (17) and after some algebraic manipulations, we can rewrite the objective function (5) in a standard quadratic form:

$$\bar{\Phi}(k) = \frac{1}{2}\bar{\mathbf{u}}^T(k)\mathbf{H}(k)\bar{\mathbf{u}}(k) + \mathbf{f}^T(k)\bar{\mathbf{u}}(k) + \mathbf{d}(k) \quad (18)$$

with

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

The matrix $\mathbf{H}(k)$ is a *Hessian* matrix, and it is always 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 $\bar{\mathbf{u}}$ and has no influence in the determination of $\bar{\mathbf{u}}^*$. Thus, we define

$$\bar{\Phi}'(k) = \frac{1}{2}\bar{\mathbf{u}}^T(k)\mathbf{H}(k)\bar{\mathbf{u}}(k) + \mathbf{f}^T(k)\bar{\mathbf{u}}(k),$$

which is a standard expression used in QP problems and the optimization problem to be solved at each sampling time is stated as follows:

$$\bar{\mathbf{u}}^* = \arg \min_{\bar{\mathbf{u}}} \{\bar{\Phi}'(k)\} \quad (19)$$

s. a.

$$\mathbf{D}\bar{\mathbf{u}}(k+j|k) \leq \mathbf{d}, \quad j \in [0, N-1] \quad (20)$$

Note that now only the control variables are used as decision variables. Furthermore, constraints for the initial condition and model dynamics are not necessary anymore, since now these informations are implicit in the cost function (18), and any constraint must be written with respect to the decision variables (constraint (20)). In this case, the amplitude constraints in the control variables of (6) can be rewritten as

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

and we have that:

$$\mathbf{D} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} \mathbf{u}_{max} - \mathbf{u}_r(k+j) \\ \mathbf{u}_{min} + \mathbf{u}_r(k+j) \end{bmatrix}$$

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 cast in the generic form given by (20). Furthermore, constraints on the control rate and states can also be formulated in a similar way.

Using the same procedure above, the matrix $\mathbf{H}(k)$ and the vectors $\mathbf{f}(k)$ and $\mathbf{d}(k)$ can be easily rewritten in order to consider the more generic cost function (12). In this case it suffices to consider $\bar{\mathbf{Q}}$ in (18) as $\bar{\mathbf{Q}} = \text{diag}(2^0\mathbf{Q}, 2^1\mathbf{Q}, \dots, 2^{N-1}\mathbf{P})$, where \mathbf{P} is the terminal state penalty matrix. Considering the same data used in the cases previously shown, the optimization problem (19)–(20) is solved at each sampling time. Figures 6 and 7 show the simulation results³ in this case, where the dash-dotted lines stand for the reference trajectories and the dashed line stand for the trajectories of the nonlinear case (Figures 4 and 5).

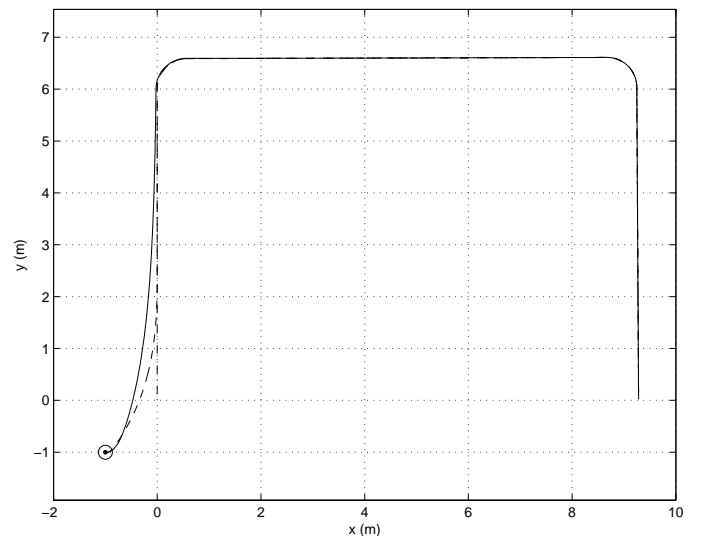


Figure 6: Trajectory in the XY plane.

³The optimization problem was solved with the MATLAB routine `quadprog`.

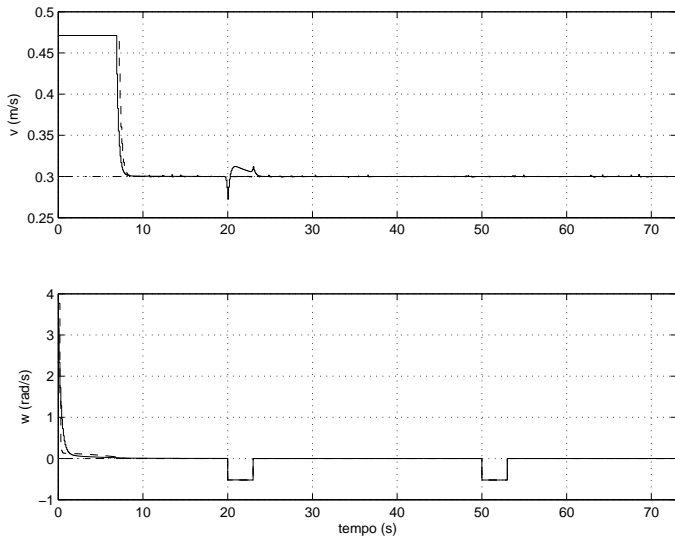


Figure 7: Controls inputs.

It can be seen that there are not significant difference between the nonlinear (Figures 4 and 5) and linear cases (Figures 6 and 7). Once more, the computed control signals respect the imposed constraints.

5 COMPUTATIONAL EFFORT

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 (Cannon e Kouvaritakis, 2000). However, with the development of increasingly faster processors the use of MPC in demanding applications becomes possible.

In order to evaluate the real-time implementability of the proposed MPC, we consider, as measurement criterion, the number of floating point operations per second (flops). With this aim we consider that the computations run in an Athlon XP 2600+ which is able to perform a peak performance between 576 and 1100 Mflops accordingly to (Aburto, 1992), a de-facto standard for floating point performance measurement.

The sampling period used in all examples presented here is $T = 100 \text{ ms}$. The data in Table 1 refers to the mean values of Mflops along the developed trajectory, for a MPC applied to the trajectory tracking with cost function of (Essen e Nijmeijer, 2001).

Table 1: Computational Effort.

Horizon	Mflops	
	NMPC	LMPC
5	11.1	0.17
10	502	0.95
15	5364	3.5
20	—	9.1

The data in Table 1 provides enough evidence that a standard of-the-shelf computer is able to run a MPC-based controller for a WMR. Note that for $N = 5$ both cases are feasible in real-time. It is important to note the difference between the nonlinear

and linear cases. For a horizon higher than 10, the NMCP would not be applicable, while the LMPC presents admissible computational effort for a horizon of 20 or even higher. On the other hand, comparing the performance showed in Fig. 6 and Table 1, we can say that the linear approach developed here is a good alternative when lower computational effort is necessary since it does not presents a significative loss in performance.

6 CONCLUSION

This paper has presented an application of MPC to solve the problem of mobile robot trajectory tracking. Two approaches have been presented, first with nonlinear MPC (leading to a non-convex optimization problem) and after that with linear MPC (by recasting the problem as a quadratic programming one). The obtained control signals are such that the constraints imposed on the control variables are respected and the convergence rates are improved by some modifications in the cost function.

In addition, a study regarding the computational effort necessary to solve the optimization problems has been carried out in order to evaluate the implementability of the proposed schemes in real-time.

With the linear approach, it has been possible to reduce the computational effort, since the MPC optimization problem has been recasted as a QP one. In comparison with the nonlinear approach, it has been noted that the linear strategy maintains good performance, with lower computational effort. It is important to point out that the LMPC has the disadvantage that the linearized model is only valid for points near the reference trajectory. However, this problem can be easily solved with the *pure-pursuit* strategy (Normey-Rico et al., 1999).

ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support from CAPES and CNPq, Brazil.

REFERENCES

- Aburto, A. (1992). Flops c program (double precision) v2.0 18 dec 1992. Available *on-line* in <ftp://ftp.nosc.mil/pub/aburto/flops>. Accessed in Jan. 2005.
- Allgower, F., Badgwell, T. A., Qin, J. S., Rawlings, J. B. e Wright, S. J. (1999). Nonlinear predictive control and moving horizon estimation – an introductory overview, in P. M. Frank (ed.), *Advances In Control: Highlights of ECC'99*, Springer-Verlag, New York, USA, pp. 391–449.
- Bloch, A. M. e McClamroch, N. H. (1989). Control of mechanical systems with classical nonholonomic constraint, *Proceedings of the IEEE Conference on Decision and Control*, Tampa, FL, USA, pp. 201–205.
- Brockett, R. W. (1982). *New directions in applied mathematics*, Springer-Verlag, New York, USA.
- Camacho, E. F. e Bordons, C. (1999). *Model Predictive Control*, Advanced Textbooks in Control and Signal Processing, Springer-Verlag, London, UK.

- Campion, G., Bastin, G. e D'Andréa-Novel, B. (1996). Structural properties and classification of kinematic and dynamical models of wheeled mobile robots, *IEEE Transactions on Robotics and Automation* **12**(1): 47–62.
- Cannon, M. e Kouvaritakis, B. (2000). Continuous-time predictive control of constrained nonlinear systems, in F. Allgöwer e A. Zeng (eds), *Nonlinear model predictive control, Progress in systems and control theory*, Vol. 26, Birkhäuser Verlag, Basel, Switzerland, pp. 205–215.
- Canudas de Wit, C. e Sørtdalen, O. J. (1992). Exponential stabilization of mobile robots with nonholonomic constraints, *IEEE Transactions on Automatic Control* **37**(11): 1791–1797.
- Chen, H. e Allgöwer, F. (1998). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability, *Automatica* **34**(10): 1205–1217.
- Do, K. D., Jiang, Z. P. e Pan, J. (2002). A universal saturation controller design for mobile robots, *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, NV, USA, pp. 2044–2049.
- Essen, H. V. e Nijmeijer, H. (2001). Non-linear model predictive control of constrained mobile robots, *European Control Conference*, Porto, Portugal, pp. 1157–1162.
- Gómez-Ortega, J. e Camacho, E. F. (1996). Mobile robot navigation in a partially structured static environment using neural predictive control, *Control Engineering Practice* **4**(12): 1669–1679.
- Henson, M. A. (1998). Nonlinear model predictive control: current status and future directions, *Computers and Chemical Engineering* **23**(2): 187–202.
- Kanayama, Y., Miyazaki, Y. K. F. e Noguchi, T. (1990). A stable tracking control method for an autonomous mobile robot, *Proceedings of the IEEE International Conference on Robotics and Automation*, Cincinnati, OH, pp. 384–389.
- Kühne, F. (2005). *Predictive control of nonholonomic mobile robots (in portuguese)*, Dissertation (M.Sc.), Federal University of Rio Grande do Sul, Porto Alegre, RS, Brazil.
- Lages, W. F. (1998). *Control and Estimation of the Position and Orientation of Mobile Robots (in portuguese)*, Thesis (D.Sc.), Instituto Tecnológico de Aeronáutica, São José dos Campos, Brazil.
- McCloskey, R. T. e Murray, R. M. (1997). Exponential stabilization of driftless control systems using homogeneous feedback, *IEEE Transactions on Automatic Control* **42**(5): 614–628.
- Normey-Rico, J. E., Gómez-Ortega, J. e Camacho, E. F. (1999). A Smith-predictor-based generalised predictive controller for mobile robot path-tracking, *Control Engineering Practice* **7**(6): 729–740.
- Ollero, A. e Amidi, O. (1991). Predictive path tracking of mobile robots. Application to the CMU Navlab, *Proceedings of the 5th IEEE International Conference on Advanced Robotics*, Pisa, Italy, pp. 1081–1086.
- Pomet, J. B., Thuilot, B., Bastin, G. e Campion, G. (1992). A hybrid strategy for the feedback stabilization of nonholonomic mobile robots, *Proceedings of the IEEE International Conference on Robotics and Automation*, Nice, France, pp. 129–134.
- Samson, C. e Ait-Abderrahim, K. (1991). Feedback control of a nonholonomic wheeled cart in cartesian space, *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, CA, USA, pp. 1136–1141.
- Sun, S. (2005). Designing approach on trajectory-tracking control of mobile robot, *Robotics and Computer-Integrated Manufacturing* **21**(1): 81–85.
- Yamamoto, Y. e Yun, X. (1994). Coordinating locomotion and manipulation of a mobile manipulator, *IEEE Transactions on Automatic Control* **39**(6): 1326–1332.
- Yang, J. M. e Kim, J. H. (1999). Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots, *IEEE Transactions on Robotics and Automation* **15**(3): 578–587.
- Yang, X., He, K., Guo, M. e Zhang, B. (1998). An intelligent predictive control approach to path tracking problem of autonomous mobile robot, *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, San Diego, CA, USA, pp. 3301–3306.