Design and Implementation of Model-Predictive Control With Friction Compensation on an Omnidirectional Mobile Robot

Júlio César Lins Barreto S., André Gustavo Scolari Conceição, Carlos E. T. Dórea, Luciana Martinez, and Edson Roberto de Pieri

***Abstract*—This paper presents and discusses the implementation results of a model-predictive control (MPC) scheme with friction compensation applied to trajectory following of an omnidirectional three-wheeled robot. A cascade structure is used with an inverse kinematics block to generate the velocity references given to the predictive controller. Part of the control effort is used to compensate for the effects of static friction, allowing the use of efficient algorithms for linear MPC with constraints. Experimental results show that the proposed strategy is efficient in compensating for frictional effects as well as for tracking predefined trajectories.**

***Index Terms*—Friction, mobile robots, modeling, predictive control.**

I. Introduction

In recent years, great interest has been shown in the application of advanced control techniques to mobile robots. There are many types of mobile robots, such as fish robots [1], one-wheeled pendulum robots [2], humanoid robots [3], trident snake robots [4], and omnidirectional robots [5]. Omnidirectional robots stand out in these applications because they are capable of moving in any direction without the need to reorient themselves, which gives them better maneuverability compared to nonholonomic robots with, for example, the Ackerman or Differential configurations [6].

Model-predictive control (MPC) techniques are commonly applied to trajectory following with mobile robots, as in [7], where the problem was solved for terrestrial autonomous vehicles and in [8], where closed-loop stability was analyzed for nonholonomic robots, but without considering friction effects and using nonconvex optimization, as in [9], where the trajectory tracking problem was solved for a four-wheeled omnidirectional robot model using nonlinear MPC, considering a static friction model. In the same scenario, a methodology using linear matrix inequalities (LMIs) was proposed in [10], where the stability of the closed-loop system is guaranteed, but at a relatively high computational cost of complex optimization algorithms.

These techniques have been expanded to include obstacle avoidance [11] and several motion mechanisms [12]. The concept of prediction is ideal for applications with known future references. The main advantages of predictive control, which is more commonly used in industry, compared to classical control, are that it works better with multivariable problems, it considers constraints on the input and the state explicitly in its formulation, and it has better robustness characteristics adapting well to disturbances, nonlinearities, and modeling errors due to the moving horizon scheme [13]. Traditional controllers, like proportional integral derivative (PID), are not capable of compensating for frictional effects and cannot work with constraints [13].

The application of MPC requires a precise model of the system. Nonlinearities are always present in reliable models of mobile robots mainly due to the effects of friction. Various friction models have been proposed in the literature [14], [15]. Dynamic models are able to capture some friction phenomena better than stationary models. On the other hand, it is generally difficult to identify the parameters of such models [16]. In addition, the models are very complex and consequently require complex control methods [17].

Many authors have addressed the problem of friction compensation, using model-based as well as non-model-based techniques [15]. The model-based techniques can use fixed friction models, identified offline [18], or adaptive algorithms which estimate the friction parameters online [19]. Even though the later allow adaptations to changes in the robot environment, their implementation is much more involved because some conditions on the excitation of the system have to be met in order to assure the correct estimation of the parameters, and many sensors have to be used to provide the necessary information [19]. Most of non-model-based techniques are based on modifications of PID controllers in order to cope with the nonlinear effects of friction. In addition to not requiring a precise model, these techniques have the advantage of compensating for other phenomena than friction, such as disturbances. However, their implementation can also be very involved and the integral action can cause limit cycles if not properly designed.

Manuscript received August 21, 2012; revised December 4, 2012; accepted January 14, 2013. Recommended by Technical Editor J. M. Berg. This work was supported by CAPES Brazil.

J. C. L. Barreto S., A. G. S. Conceição, and L. Martinez are with the Department of Electrical Engineering, Universidade Federal da Bahia, Salvador, BA-40210630 Brazil (e-mail: julioclb@ufba.br; andre.gustavo@ufba.br; lucianam@ufba.br).

C. E. T. Dórea is with the Department of Computing Engineering and Automation, Universidade Federal do Rio Grande do Norte, Natal, RN-59078900 Brazil (e-mail: cetdorea@dca.ufrn.br).

E. R. de Pieri is with the Department of Automation and Systems, Universidade Federal de Santa Catarina, Florianópolis, SC-88040900 Brazil (e-mail: edson@das.ufsc.br).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMECH.2013.2243161

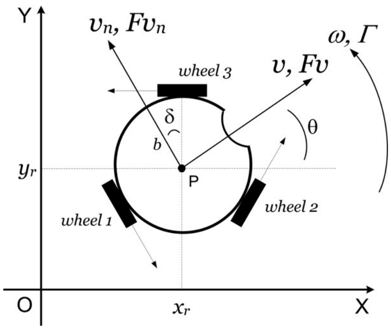


Fig. 1. Coordinate systems and geometric parameters.

In this paper, a static friction model for a three-wheeled omnidirectional robot is used, which takes into consideration the frictional forces related to the velocities of the robot center of mass. The parameters of the friction model are estimated using the methods proposed in [20] for a four-wheeled omnidirectional robot. The proposed control strategy uses a cascade structure with an inverse kinematics block which generates the velocity references to be sent to the predictive controller. Then, part of the control effort is used as compensation for static friction, in a sort of feedback linearization strategy, which allows the use of efficient constrained linear MPC algorithms. The experimental results show that the proposed strategy is efficient in compensating for the friction effects, as well as in tracking a predefined trajectory.

This paper is organized in the following sequence. In Section II, the robot nonlinear model is presented. In Section III, the development of the MPC with friction compensation is shown. In Section IV, the experimental results are presented and discussed. Finally, the conclusions are presented in Section V.

II. Robot Model

A. Robot Kinematics

Using the robot coordinate system geometry shown in Fig. 1, the robot kinematic equations of motion can be written as

$$\begin{bmatrix} \frac{dx_r(t)}{dt} \\ \frac{dy_r(t)}{dt} \\ \frac{d\theta(t)}{dt} \end{bmatrix} = R_o^T(\theta(t)) \begin{bmatrix} v(t) \\ v_n(t) \\ w(t) \end{bmatrix} \tag{1}$$

where the orthogonal rotation matrix $R_o(\theta(t))$ converts from the earth coordinate system to the robot coordinate system and vice versa

$$R_o(\theta(t)) = \begin{bmatrix} \cos(\theta(t)) & \sin(\theta(t)) & 0 \\ -\sin(\theta(t)) & \cos(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

The robot pose is represented by the vector $[x_r(t)\ \ y_r(t)\ \ \theta(t)]^T$ that describes the location of the robot in the earth coordinate system and the angular difference between the coordinate systems. The orthogonal components $v(t)$ and $v_n(t)$ of the linear velocity, and the angular velocity $w(t)$ of the robot are represented by the vector $[v(t)\ \ v_n(t)\ \ w(t)]^T$.

The relationship between the robot wheels' angular velocities ($w_{m_i}(t)$, where $i = 1, 2, 3$) and the robot velocities $(v(t), v_n(t), w(t))$ is given by

$$\begin{bmatrix} v(t) \\ v_n(t) \\ w(t) \end{bmatrix} = \begin{bmatrix} 0 & \frac{r_2\sqrt{3}}{3} & \frac{-r_3\sqrt{3}}{3} \\ \frac{2r_1}{3} & \frac{r_2}{3} & \frac{r_3}{3} \\ \frac{r_1}{3b} & \frac{r_2}{3b} & \frac{r_3}{3b} \end{bmatrix} \begin{bmatrix} w_{m_1}(t) \\ w_{m_2}(t) \\ w_{m_3}(t) \end{bmatrix} \tag{3}$$

where b is the distance between the robot center of mass and the wheels, and r_i is the radius of wheel i.

B. Robot Dynamics

The omnidirectional mobile robot model was developed based on the dynamics and the kinematics of the robot base and on the dynamics of the DC motors. This model is used to predict the future robot positions and orientations and takes into account elements such as saturation and friction. In this paper, a static friction model (coulomb and viscous friction) is used to represent the composition of the forces acting on the robot center of mass. The coulomb friction is proportional to the load; therefore, the friction between the robot and the ground surface is modeled as coulomb friction. The viscous friction force is caused by the viscosity of lubricants and is assumed to have a linear relationship with the velocity [21]. From the coordinates defined in Fig. 1 and Newton's second law, the equations relating translational and rotational forces acting in the robot can be written as

$$F_v(t) - B_v v(t) - C_v \mathrm{sgn}(v(t)) = M\frac{dv(t)}{dt} \tag{4}$$

$$F_{v_n}(t) - B_{v_n} v_n(t) - C_{v_n} \mathrm{sgn}(v_n(t)) = M\frac{dv_n(t)}{dt} \tag{5}$$

$$\Gamma(t) - B_w w(t) - C_w \mathrm{sgn}(w(t)) = I_n\frac{dw(t)}{dt} \tag{6}$$

$$\mathrm{sgn}(\alpha) = \begin{cases} 1, & \alpha > 0 \\ 0, & \alpha = 0 \\ -1, & \alpha < 0. \end{cases}$$

Here, F_v and F_{v_n} represent the force vectors in the robot coordinate system, and Γ represents the momentum about the robot center of mass (point P). M is the mass of the robot, and I_n is the inertial momentum. The viscous forces and torque are represented by $B_v v(t), B_{v_n} v_n(t)$ and $B_w w(t)$, and the coulomb forces and torque by $C_v \mathrm{sgn}(v(t)), C_{v_n} \mathrm{sgn}(v_n(t))$, and $C_w \mathrm{sgn}(w(t))$. The relationships between the robot traction

TABLE I
MODEL PARAMETERS

Symbol	*Description*	*Value*
$B_v(N/m/s)$	viscous friction relative to v	2
$B_{v_n}(N/m/s)$	viscous friction relative to v_n	1.5
$B_\omega(N/rad/s)$	viscous friction relative to ω	0.024
$C_v(N)$	Coulomb friction relative to v	1.2
$C_{v_n}(N)$	Coulomb friction relative to v_n	0.8
$C_\omega(N.m)$	Coulomb friction relative to ω	0.0035
$b(m)$	robot radius	0.1
$M(kg)$	robot mass	1.5
$I_n(kg.m^2)$	robot inertial momentum	0.025
δ	angle	30^o
$r_{1\ldots3}(m)$	wheels' radius	0.035
$l_{1\ldots3}$	reduction	19:1
$L_{a_{1\ldots3}}(H)$	armature inductances	0.00011
$R_{a_{1\ldots3}}(\Omega)$	armature resistances	1.66
$K_{v_{1\ldots3}}(Volts/rad/s)$	emf constants	0.0059
$K_{t_{1\ldots3}}(N.m/A)$	torque constants	0.0059

forces and the wheels' traction forces are given by

$$F_v(t) = \cos(\delta)(f_2(t) - f_3(t)) \tag{7}$$

$$F_{v_n}(t) = -f_1(t) + \text{sen}(\delta)f_2(t) + \text{sen}(\delta)f_3(t) \tag{8}$$

$$\Gamma(t) = (f_1(t) + f_2(t) + f_3(t))b. \tag{9}$$

The traction force on each wheel i (with $i = 1, 2, 3$) is given by

$$f_i(t) = \frac{T_i(t)}{r_i} \tag{10}$$

where T_i is the wheel's torque. The DC motor dynamics for $i = 1, 2, 3$ can be described by the following equations:

$$u_i(t) = L_{a_i}\frac{di_{a_i}(t)}{dt} + R_{a_i}i_{a_i}(t) + K_{v_i}w_{m_i}(t) \tag{11}$$

$$T_i(t) = l_iK_{t_i}i_{a_i}(t) \tag{12}$$

where u_i are the armature voltages, L_{a_i} are the armature inductances, R_{a_i} are the resistances, l_i are the motor reductions, w_{m_i} are the rotors' angular velocities, and i_{a_i} are the armature currents. The motors parameters, geometric parameters, and the estimated parameters are presented in Table I.

C. State-Space Representation

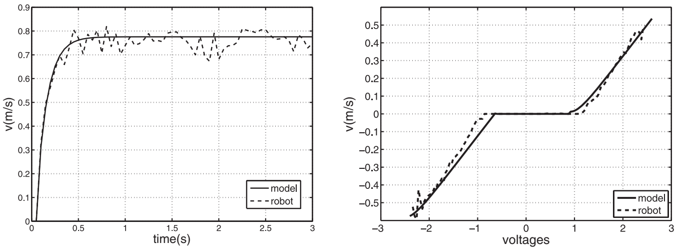
Writing the model equations in state-space form gives

$$\dot{x}(t) = Ax(t) + Bu(t) + K\text{sgn}(x(t)) \tag{13}$$

$$y(t) = Cx(t) \tag{14}$$

where the vector $u(t) = [u_1(t) \;\; u_2(t) \;\; u_3(t)]^T$ is the control input and the vectors $y(t) = x(t) = [v(t) \;\; v_n(t) \;\; w(t)]^T$ are the output and the state variables of the system. Considering $l = l_{1\ldots3}, r = r_{1\ldots3}, R_a = R_{a_{1\ldots3}}$ and $K_t = K_{t_{1\ldots3}}$, the simplified state-variable-form matrices defining this system are

$$A = \begin{bmatrix} -\frac{3l^2K_t^2}{2MR_ar^2} - \frac{B_v}{M} & 0 & 0 \\ 0 & -\frac{3l^2K_t^2}{2MR_ar^2} - \frac{B_{v_n}}{M} & 0 \\ 0 & 0 & -\frac{3b^2l^2K_t^2}{JR_ar^2} - \frac{B_w}{I_n} \end{bmatrix}$$



(a) (b)

Fig. 2. Model and robot velocities. (a) Linear velocity v, with $u_1 = 0, u_2 = 3V$ and $u_3 = -3V$. (b) Dead-zone effect.

$$B = \frac{lK_t}{R_ar}\begin{bmatrix} 0 & \frac{\cos(\delta)}{M} & -\frac{\cos(\delta)}{M} \\ \frac{-1}{M} & \frac{\text{sen}(\delta)}{M} & \frac{\text{sen}(\delta)}{M} \\ \frac{b}{I_n} & \frac{b}{I_n} & \frac{b}{I_n} \end{bmatrix}$$

$$K = \begin{bmatrix} -\frac{C_v}{M} & 0 & 0 \\ 0 & -\frac{C_{v_n}}{M} & 0 \\ 0 & 0 & -\frac{C_\omega}{I_n} \end{bmatrix}, \quad C = I.$$

Analyzing (13), one can observe that the nonlinearity lies in the $K\text{sgn}(x(t))$ term.

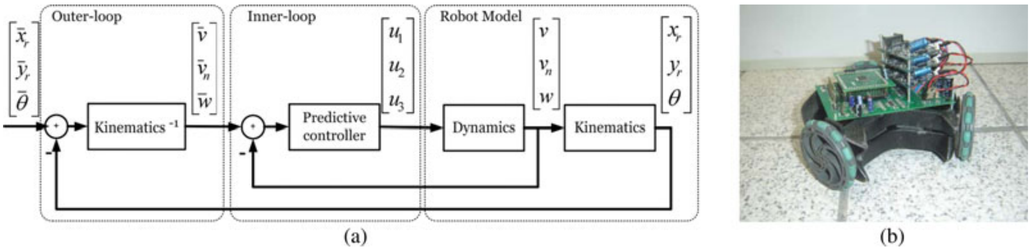
In static friction models, the friction depends only on the current velocity value. Other friction phenomena can be captured only by dynamic models [21]. The performance of the proposed strategy depends on the model performance. Fig. 2 shows experimental results and simulations to demonstrate the efficacy of the proposed modeling, where the solid curve represents the model velocities and the dashed curve represents the robot velocities. The parameters of the model must be estimated in advance; in this study, the robot environment has a carpet on the floor, similar to those used in RoboCup Soccer League competitions [22]. In the case of changes in the environment, such as using different types of carpets, a new parameter estimation is needed. The estimation methods used to obtain the friction coefficients and the inertial momentum can be found in [20].

III. PREDICTIVE CONTROLLER

The MPC scheme proposed in this paper is based on a cascade structure, as shown in Fig. 3. This scheme is similar to the scheme used in [23], where the internal loop controls the robot dynamics, and the velocity references are given by the external control loop. The block that implements the robot inverse kinematics generates the velocity references from knowledge of the robot position. Consequently, the internal controller should be designed to control the robot velocity, as shown in (13) and (14).

The use of MPC in the internal loop presents the following advantages [13], [24].

1) Optimum control effort with respect to the predicted future behavior of the system is obtained. This is an ideal condition for predefined trajectory tracking.



(a) (b)

Fig. 3. Control scheme. (a) Controller structure. (b) Mobile robot.

2) Constraints on the state and control variables are explicitly considered.
3) The optimum control effort is recalculated at each sample period (moving horizon), making the control less sensitive to disturbances and model mismatch.

MPC is more easily designed using discrete time models. The discretization of (13) using the step invariant method gives the following model:

$$x(k+1) = A_d x(k) + B_d u(k) + K_d \text{sgn}(x(k)). \tag{15}$$

The discretization was carried out assuming that no change in the signal of the components of $x(t)$ occurs between two consecutive samples.

Omnidirectional three-wheeled robots have three control entries and only three state variables for velocity control. Consequently, some of the degrees of freedom of the control can be used to compensate for nonlinearities. Then, the following control efforts can be implemented:

$$u(k) = u_f(k) + u_c(k)$$

where $u_f(k)$ is a linearizing feedback, such that

$$B_d u_f(k) = -K_d \text{sgn}(x(k)). \tag{16}$$

The equation above has an unique solution for any $x(k)$ because $B_d \in R^{3\times 3}$ and $\text{rank}(B_d) = 3$. From this choice of $u_f(k)$, the system dynamics becomes linear with respect to $u_c(k)$:

$$x(k+1) = A_d x(k) + B_d u_c(k). \tag{17}$$

The supply voltages of the motors are limited to 6 V. Therefore, the control entry $u(k)$ is subject to the following constraints:

$$|u_l(k)| \leq 6, \text{ for } l = 1, 2, 3.$$

Because of the linearizing term, the control efforts should obey the nonlinear constraints in (16), along the control horizon H_u. This difficulty can be overcome by imposing the control constraint only at time k because $x(k)$ is the known vector of measured velocities at instant k.

The optimization problem associated with MPC at each sample period k can now be formulated as (18). H_p is the prediction horizon; H_u is the control horizon; Q and R are symmetric weighting matrices (positive definite) with appropriate dimensions; $\bar{x}(k)$ is a vector of the reference velocities given by the inverse kinematics block, $\bar{u}(k)$ is the corresponding input, and $x(k)$ is the vector of measured velocities at time k. The optimization problem in (18) is a quadratic programming (QP) problem, for which efficient numerical methods are available.

A. *Interior Point Method*

Linear MPC involves the resolution of a QP problem at each sample interval. In this paper, the interior point method (IPM) is considered. Beginning with the Karmarker [25], IPMs have been developed in order to ensure polynomial-time performance for different applications. This method offers an alternative to active-set methods in MPC [26] [27].

Due to the fact that the objective function is quadratic and the constraints are linear on the variables $x(k+i), u_f(k)$ and $u_c(k+j)$, through standard manipulation in the linear MPC framework [13], the optimization problem (18), as shown at the bottom of the page, can be rewritten in the form of a QP problem:

$$\begin{aligned} &\min_{u_a} u_a^T\, H\, u_a - G^T\, u_a \\ &\text{subject to} \begin{cases} A_E\, u_a = b_E \\ A_I\, u_a \leq b_I \end{cases} \end{aligned} \tag{19}$$

where the decision variable of the problem is defined as $u_a = [u_c(k), \ldots u_c(k+Hu-1) u_f(k)]^T$ and the matrices $H \in$

$$\begin{gathered} \min_{u_f(k), u_c(k+j)} \sum_{i=1}^{H_p} ((x(k+i) - \bar{x}(k+i))^T Q (x(k+i) - \bar{x}(k+i))) + \sum_{j=0}^{H_u} ((u(k+j) - \bar{u}(k+j))^T R (u(k+j) - \bar{u}(k+j))) \\ \text{s.t.:} \quad x(k+i+1) = A_d x(k+i) + B_d u_c(k+i) \\ u(k) = u_f(k) + u_c(k) \\ u(k+j) = u_c(k+j) \text{ for } j > 0, \\ B_d u_f(k) = -K_d \text{sgn}(x(k)) \\ |u(k)| \leq 6 \end{gathered} \tag{18}$$

$\mathbb{R}^{3(H_u+1)\times 3(H_u+1)}$ and $G \in \mathbb{R}^{3(H_u+1)\times 3}$ are given by

$$H = \Theta^T Q_c \Theta \, + \, R_c \tag{20}$$

$$G = 2[\epsilon^T Q_c \Theta^T + u_a^T R_c] \tag{21}$$

with,

$$Q_c = \begin{bmatrix} Q & 0 & 0 & \dots & 0 \\ 0 & Q & 0 & \cdots & 0 \\ \vdots & & & \dots & \\ 0 & 0 & 0 & \dots & Q \end{bmatrix}$$

$$R_c = \begin{bmatrix} R & 0 & 0 & \dots & R \\ 0 & R & 0 & \cdots & 0 \\ \vdots & & & \dots & \\ R & 0 & 0 & \dots & R \end{bmatrix}$$

$$\Theta = \left[\begin{array}{cccc} B_d & 0 & 0 & 0 \\ A_d B_d & B_d & 0 & 0 \\ \vdots & & \dots & \\ A_d^{H_u} B_d & A_d^{Hu-1} B_d & \dots & A_d B_d \\ A_d^{H_u+1} B_d & A_d^{Hu} B_d & \dots & A_d^2 B_d \\ \vdots & & \dots & \\ A_d^{H_p-1} B_d & A_d^{H_p-2} B_d & \dots & \end{array} \right.$$

$$\left. \begin{array}{ccccc} & & \dots & 0 & 0 \\ & & \dots & 0 & 0 \\ & & \dots & 0 & 0 \\ B_d & 0 & \dots & 0 & 0 \\ A_d B_d & B_d & \dots & 0 & 0 \\ & & \dots & 0 & 0 \\ & & A_d B & B_d & 0 \end{array} \right]$$

where $R_c \in \mathbb{R}^{(3(Hu+1)\times 3(Hu+1))}, Q_c \in \mathbb{R}^{(3Hp\times 3Hp)}, R \in \mathbb{R}^{3\times 3}, Q \in \mathbb{R}^{3\times 3}$, and 0 are null matrices with suitable dimensions, and ϵ is the error between the reference trajectory and the past outputs given by $\epsilon(k) = \bar{x}(k) - C[A_d A_d^2 \dots A_d^{H_p}]^T x(k)$.

The matrices A_E, b_E and A_I, b_I represent the equality and inequality constraints of the problem (18), respectively, and are given by

$$A_E = [0 \;\; 0 \;\; 0 \;\; \dots \;\; B_d] \, ; b_E = [-K_d \, \text{sgn}(x(k))] \tag{22}$$

$$A_I = \begin{bmatrix} I \;\; 0 \;\; 0 \dots I \\ -I \;\; 0 \;\; 0 \dots -I \end{bmatrix} ; \;\; b_I = \begin{bmatrix} 6I \\ 6I \end{bmatrix} \tag{23}$$

where $I \in \mathbb{R}^{3\times 3}$ and $0 \in \mathbb{R}^{3\times 3}$.

The IPM is essentially a barrier method. In the problem (19), the barrier is of the form [28]

$$\min_{u_a} u_a^T H u_a - G^T u_a - \mu \sum_{i=1}^{m} ln \, s_i$$

$$\text{subject to} \begin{cases} A_E \, u_a = b_E \\ A_I \, u_a + s = b_I \end{cases} \tag{24}$$

where $\mu > 0$ is the barrier parameter and the slack variable $s \in \mathbb{R}^{m\times 1}$ is assumed to be positive (here $m = 6$). By letting μ converge to zero, the sequence of approximate solutions to (24) will normally converge to a minimizer of the original nonlinear program (19).

To characterize the solution of the barrier problem (24), the Lagrangian is considered

$$\mathcal{L}(u_a, s, \lambda_E, \lambda_I) = u_a^T \, H \, u_a - G^T \, u_a - \mu \sum_{i=1}^{m} ln \, s_i$$
$$+ \lambda_E^T (A_E \, u_a - b_E) + \lambda_I^T (A_I \, u_a + s - b_I) \tag{25}$$

where λ_E and λ_I are the multipliers associated with the equality and inequality constraints, respectively.

The primal-dual algorithm adopted is based on the Karush–Kuhn–Tucker conditions for the barrier problem given by

$$\begin{aligned} (2 \, H \, u_a - G) + A_E \lambda_E + A_I \lambda_I &= 0 \\ S \lambda_I - \mu e &= 0 \\ A_E \, u_a - b_E &= 0 \\ A_I \, u_a + s - b_I &= 0 \end{aligned} \tag{26}$$

where $e = [1, \dots, 1]^T, S = \text{diag}(s^1, \dots, s^m)$, with superscripts indicating components of a vector.

The solution of the nonlinear system (26) is then obtained by applying Newton's method [28]

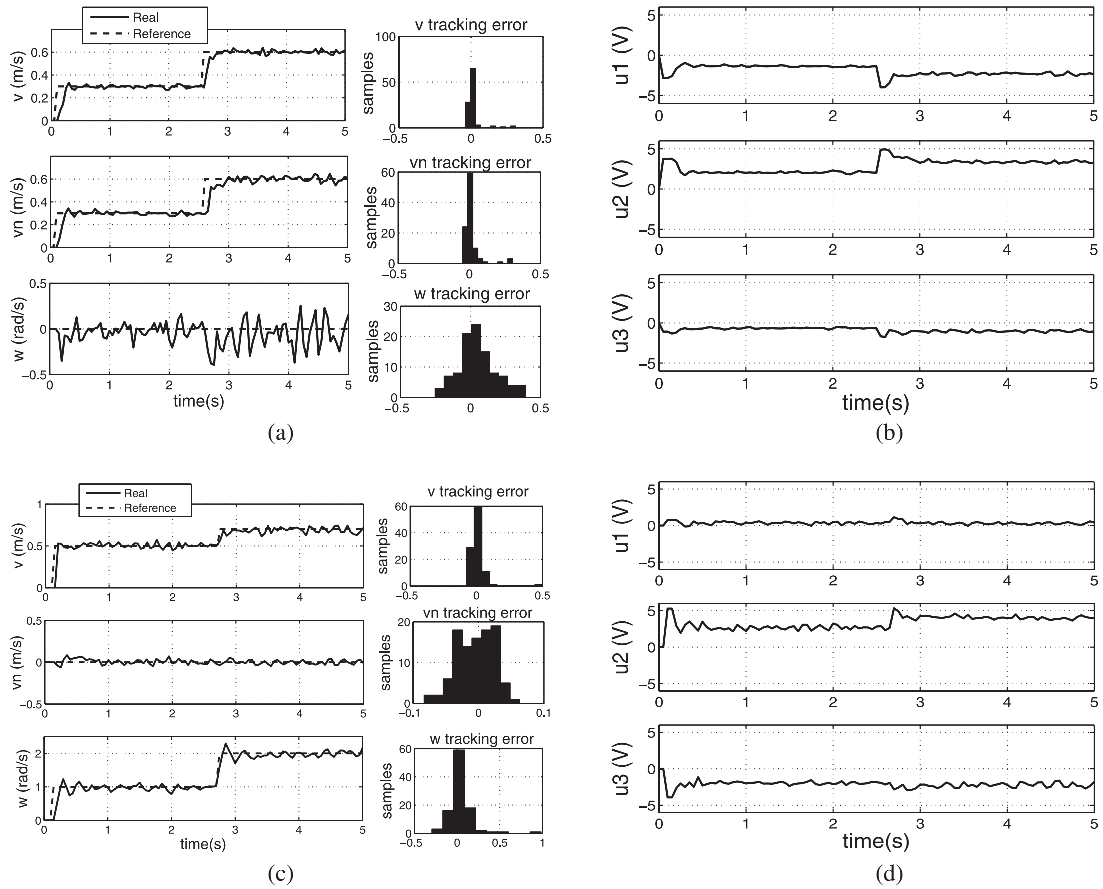
$$\begin{pmatrix} 2H & 0 & A_E & A_I \\ 0 & \sum_k & 0 & S \\ A_E^T & 0 & 0 & 0 \\ A_I^T & I & 0 & 0 \end{pmatrix} \begin{pmatrix} p_{u_a} \\ p_s \\ p_{\lambda_E} \\ p_{\lambda_I} \end{pmatrix} =$$
$$- \begin{pmatrix} 2Hu_a - G + A_E \lambda_E + A_I \lambda_I \\ S \lambda_I - \mu e \\ A_E \, u_a - b_E \\ A_I \, u_a + s - b_I \end{pmatrix} \tag{27}$$

where $\sum_k = S_k^{-1} \Lambda_I$. Here, $\Lambda_I = \text{diag}(\lambda_I^1, \dots, \lambda_I^m)$ contains the Lagrange multiplier estimates corresponding to the inequality constraints.

The system (27) is equivalent to a Newton iteration on the Karush–Kuhn–Tucker conditions of the barrier problem (19). By Newton's method, the values of the variables u_a, s, λ_E, and λ_I are updated through an iterative process, using the vector p as follows:

$$\begin{aligned} u_a^+ &= u_a + \alpha_s^{\max} p_{u_a} \\ s^+ &= s + \alpha_s^{\max} p_s \\ \lambda_E^+ &= \lambda_E + \alpha_{\lambda_E}^{\max} p_{\lambda_E} \\ \lambda_I^+ &= \lambda_I + \alpha_{\lambda_I}^{\max} p_{\lambda_I} \, . \end{aligned} \tag{28}$$

The step length α in each direction is determined in order not to violate the nonnegativity constraints on the variables λ_E, λ_I,



(a) (b)

(c) (d)

Fig. 4. Velocity control—internal control loop. (a) Robot velocities. (b) Motors voltages. (c) Robot velocities. (d) Motors voltages.

and s:

$$\begin{aligned}\alpha_s^{\max} &= \{\max \alpha \in (0,1] : s + \alpha p_s \geq (1-\tau)s\}\\ \alpha_{\lambda_E}^{\max} &= \{\max \alpha \in (0,1] : \lambda_E + \alpha p_{\lambda_E} \geq (1-\tau)\lambda_E\}\\ \alpha_{\lambda_I}^{\max} &= \{\max \alpha \in (0,1] : \lambda_I + \alpha p_{\lambda_I} \geq (1-\tau)\lambda_I\}. \end{aligned} \quad (29)$$

In this paper, the value of $\tau = 0.995$ is adopted [28].

IV. Experimental Results

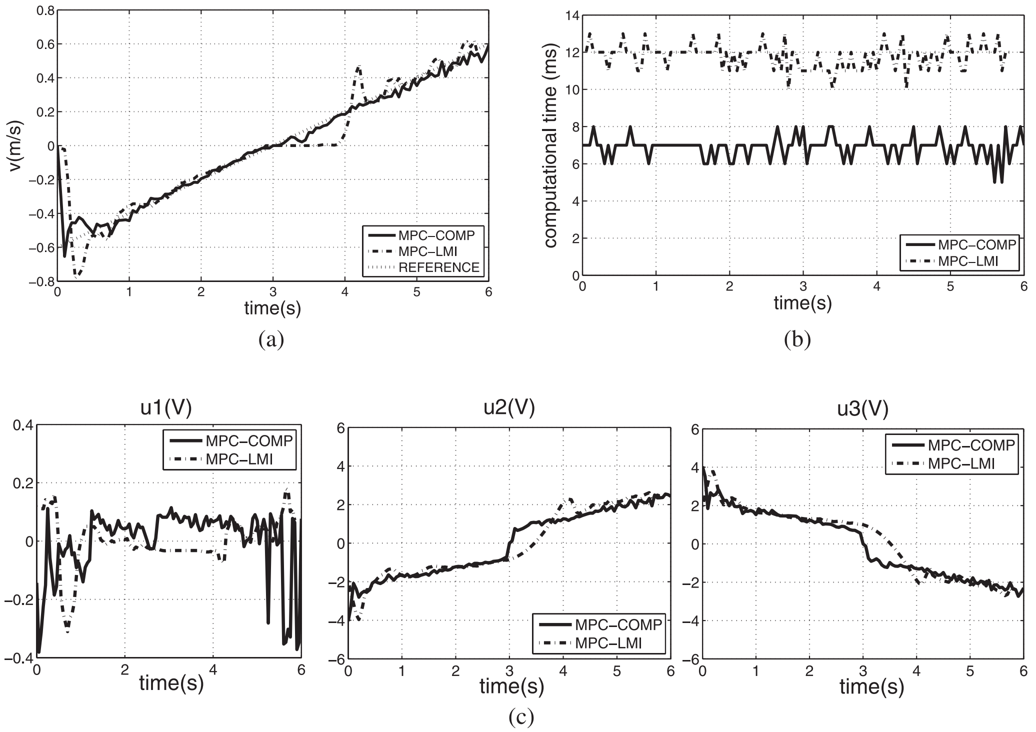
The mobile robot in Fig. 3(b) is composed of two main modules: a microprocessor system responsible for the implementation of the system's instrumentation and the time base generation module (sample time of 50 ms) that creates a real time clock. The high-level controllers and the supervisory system are implemented on a personal computer. These modules communicate through a Zigbee platform composed by two Maxstream's Xbee modules [29]. The robot odometry data and the control signals (motors voltages) are transmitted serially through 32-byte packets at a rate of 57 600 b/s. The DC motors are of A-max 22 type (nominal voltage: 6 V, power rating: 5 W) developed by Maxon Motors, and they are controlled by H-bridge circuits made by Acroname Robotics (part no. S17-3A-LV-BRIDGE). The supervisory system and the control algorithms were implemented via Lazarus IDE software for Linux/Ubuntu operation systems on a Pentium Core i7 (2.8 GHz clock speed). The interior point algorithm was implemented in Object Pascal. The MPC controller parameters for all tests in this paper were the following: the cost function weight matrices are defined as $Q = 100I$ and $R = 0.01I$. The prediction and control horizons are selected as $H_p = 3$ and $H_u = 3$ (three time periods = 150 ms). The input signal constraints are $|u_i| \leq 6, i = 1, 2, 3$.

A. Velocity Control

In order to verify the effectiveness of the internal control loop, a set of reference velocities was tested on the robot, as can be seen in Fig. 4. The temporal evolution of the robot velocities, tracking errors, and the control efforts are shown in Fig. 4(a) and (b) for linear reference velocities: $\bar{x} = [0.3\ \ 0.3\ \ 0]$ and $\bar{x} = [0.6\ \ 0.6\ \ 0]$. Fig. 4(c) and (d) shows the plots for linear and angular reference velocities: $\bar{x} = [0.5\ \ 0\ \ 1]$ and $\bar{x} = [0.7\ \ 0\ \ 2]$. The MPC controller tracked the reference and the constraints on the control variables (the voltages u_1, u_2 and u_3) are respected.

B. Friction Compensation

In this section, experimental results of the friction compensation method (here called MPC-COMP) described in the previous section are presented. For comparison, we also tested a recent approach of MPC based on LMIs, as described in [10]. This MPC controller based on LMIs, here called MPC-LMI, uses finite horizon together with a sufficient condition for closed-loop stability. Constraints on the input signals are also considered



(a)

(b)

(c)

Fig. 5. Dead-zone compensation. (a) Velocity curve, component v. (b) Computational time cost. (c) Motors voltages—Control signals.

in the control law design. The stability condition is obtained by guaranteeing that an upper bound for the finite horizon cost function is monotonically nonincreasing. The MPC-LMI tuning parameters used in the test are as follows: the cost function weighting matrices were defined as $Q = 100I$ and $R = 0.01I$. The prediction and control horizons were selected as $H_p = 3$ and $H_u = 3$. The input signals constraints were $|u_i| \leq 6, i = 1, 2, 3$.

The plots in Fig. 5(a) show the behavior of the robot velocity v and can be used to compare the behaviors of the controllers. The curve in dots represents a time-varying reference velocity $\bar{v}$. For the cases in which the robot has nonnull velocities v and v_n, the effect of static friction is larger when compared with the velocities w. This is easily explained by the geometry of the robot, which, in order to generate velocities in the v direction, activates only two of the three motors. In addition, the wheels have an angle of 30° related to velocity vector v (see Fig. 1). In the case of w, the three motors are active simultaneously in the same direction and with the same input voltage. One can observe that for the MPC-COMP case, the dead zone is considerably smaller than for the MPC-LMI case because of the static friction compensation effect. The difference in performance between the two controllers is due to the difference in control action. Comparing the dead zone effect, in the MPC-COMP case, one can note that the static friction compensation was corrected for a large part of the dead zone, and the voltages vary abruptly when they are close to zero, as shown in Fig. 5(c). The computation time in milliseconds required for solving the control law is shown in 5(b). In both cases, the control algorithms can be used in MPC controllers. However, the MPC-COMP has a lower computational time cost.

TABLE II
TRAJECTORY POINTS

Goals	$\bar{x_r}$ (m)	$\bar{y_r}$ (m)	$\bar{\theta}$ (rad)	v_{nav} (m/s)	Time to goal (s)
G1	1	0	$0.5pi$	0.2	6.45
G2	1	1	pi	0.5	2.3
G3	0	1	$1.5pi$	0.2	6.25
G4	0	0	$2pi$	0.5	2.3

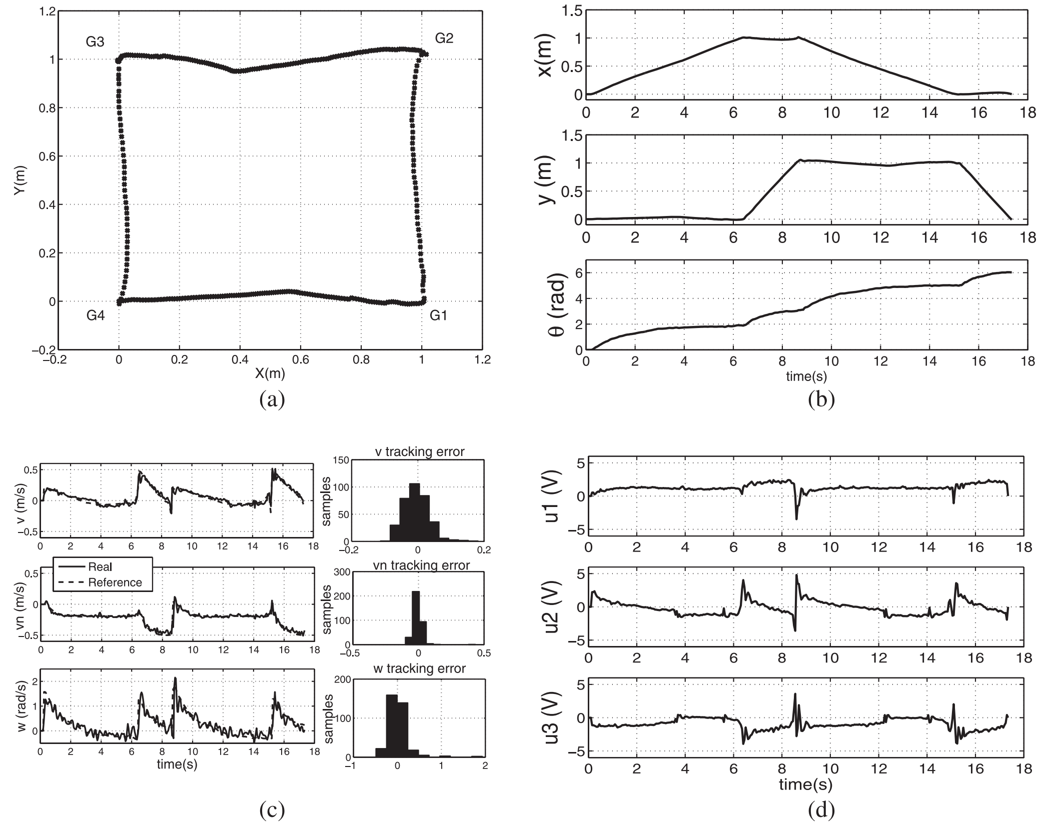
C. Outer-Loop Control

In this section, results of the cascade structure controller are presented. The trajectory of reference is defined as a set of points in the world frame (OXY): $Traj(k+j) = [\bar{x_r}(k+j)\ \bar{y_r}(k+j)\ \bar{\theta}(k+j)]^T, j = 0, 1, \ldots, H_p - 1$. Then, given the current position and heading of the robot, it is necessary to calculate the desired velocities for the next H_p periods of time. The vector of velocity references $\bar{x}(k+j|k) = [\bar{v}(k+j|k)\ \bar{v_n}(k+j|k)\ \bar{w}(k+j|k)]^T, j = 0, 1, \ldots, H_p - 1$, where j is an step prediction of the robot velocity made at instant k, is given by

$$\begin{bmatrix} \bar{v}(k+j|k) \\ \bar{v_n}(k+j|k) \\ \bar{w}(k+j|k) \end{bmatrix} = R_o(\theta(k)) \begin{bmatrix} e_{vx} \\ e_{vy} \\ e_w \end{bmatrix} \tag{30}$$

with

$$\begin{bmatrix} e_{vx} \\ e_{vy} \\ e_w \end{bmatrix} = \begin{bmatrix} v_{nav}\cos(\varphi) \\ v_{nav}\sin(\varphi) \\ \bar{\theta}(k+j|k) - \theta(k) \end{bmatrix} \tag{31}$$



(a) (b)

(c) (d)

Fig. 6. Square tracking performance. (a) Robot trajectory. (b) Pose response. (c) Robot velocities. (d) Motors voltages.

and

$$\varphi = a\tan2\left(\bar{y_r}(k+j|k) - y_r(k), \bar{x_r}(k+j|k) - x_r(k)\right) \quad (32)$$

where v_{nav} is the velocity of the robot center of mass, which is a design parameter. The reference trajectory points, the parameter v_{nav}, and the time to reach the goal (time to goal) of the experiment are listed in Table II.

The trajectory has special features, as sudden change of direction and orientation of the robot and movements of rotation and translation at the same time, in order to test the controller in hard condition. The initial robot pose was $[0(\mathrm{m})\ 0(\mathrm{m})\ 0(\mathrm{rad})]^T$ and its velocity equals zero. The linear velocity of navigation was $v_{nav} = 0.2(\mathrm{m/s})$ to reach the goals G1 and G3 and $v_{nav} = 0.5(\mathrm{m/s})$ to reach the goals G2 and G4.

The robot trajectory, pose response, robot velocities, tracking errors, and motors voltages are illustrated in Fig. 6. It can be observed from the robot position in space [see Fig. 6(a)] that the tracking performance was almost ideal, even during the transition between goals. The peak tracking error is under 5-cm bound. The orientation angle tracking error is less than 10°. It can be observed that with $v_{nav} = 0.2(\mathrm{m/s})$, the trajectory was tracked with a total navigation time of 6.45 s (to reach G1) and 6.25 s (to reach G3), and with $v_{nav} = 0.5(\mathrm{m/s})$, it was tracked with a time of 2.3 s (to reach G2 and G4). The motors voltages do not exceed the predefined limits, as can be seen in Fig. 6(d). The values of voltage reach a maximum absolute value of 5 V, which is below the saturation limit. It can also be noted that the control effort requires higher voltages at the change of goal points.

V. Conclusion

In this paper, a predictive control algorithm with friction compensation was proposed to solve the trajectory tracking problem for a three-wheeled omnidirectional mobile robot. In a cascade control structure, an inverse kinematics block was used to compute the velocity references at each step from information about the robot position and a desired trajectory. The adoption of a simplified friction model allowed the use of part of the control effort to linearize the system which, in turn, allowed the use of an efficient algorithm for linear MPC with constraints.

The reasons to choose the predictive control technique were justified in practice: the constraints were respected, frictional effects were satisfactorily compensated, and a multiple-input multiple-output nonlinear system was quite easily controlled. From the friction compensation test, it was shown that, even though the proposed control scheme did not eliminate the dead zone, it considerably reduced its size. The experimental results have shown the good performance of the proposed control strategy, its flexibility for mobile robot applications, and its vast calibration capacity. For example, the penalty coefficients, prediction and control horizons, the outer-loop controller, and navigation velocity can be adjusted depending on the application and/or type of trajectory.

Compared to other works on a similar problem, we believe that our approach presents the following advantages.

1) It uses a friction model that can be efficiently estimated and that allows for the use of a simple compensation technique.
2) From the compensation of the friction effect, a linear MPC formulation can be used for the velocity control, which leads to a simple and efficient implementation via convex optimization, suitable for real-time embedded systems.

In this paper, we focused on experimental results of the proposed technique. The analysis of closed-loop stability is a very important issue though. The most accepted technique used to prove stability in MPC with constraints relies on the use of a positive invariant set as a terminal constraint [30]. There are a number of methods for determination of such a set in the linear case. In the nonlinear case, however, there is no systematic way of doing that, to our knowledge. The use of a cascade structure with nonlinear elements makes this task even more difficult. Stability analysis and other theoretical issues should then be object of future work.

References

[1] C. Zhou and K. H. Low, "Design and locomotion control of a biomimetic underwater vehicle with fin propulsion," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 1, pp. 25–35, Feb. 2012.

[2] H. Jin, J. Hwang, and J. Lee, "A balancing control strategy for a one-wheel pendulum robot based on dynamic model decomposition: Simulations and experiments," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 4, pp. 763–768, Jul. 2011.

[3] Z. Xia, J. Xiong, and K. Chen, "Global navigation for humanoid robots using sampling-based footstep planners," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 4, pp. 716–723, Aug. 2011.

[4] M. Ishikawa, Y. Minami, and T. Sugie, "Development and control experiment of the trident snake robot," *IEEE/ASME Trans. Mechatronics*, vol. 15, no. 1, pp. 9–16, Feb. 2010.

[5] Z. Zhao, X. Deng, and J. Yi, "Motion and internal force control for omnidirectional wheeled mobile robots," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 3, pp. 382–387, Jun. 2009.

[6] X. Li and A. Zell, "Motion control of an omnidirectional mobile robot," in *Proc. 4th Int. Conf. Inf. Control Autom. Robot.*, 2007, pp. 125–132.

[7] L. Vachhani, A. D. Mahindrakar, and K. Sridharan, "Mobile robot navigation through a hardware-efficient implementation for control-law-based construction of generalized Voronoi diagram," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 6, pp. 1083–1095, Dec. 2011.

[8] D. Gu and H. Hu, "Receding horizon tracking control of wheeled mobile robots," *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 4, pp. 743–749, Jul. 2006.

[9] A. G. S. Conceicao, H. Oliveira, A. Sousa e Silva, D. Oliveira, and A. Moreira, "A nonlinear model predictive control of an omnidirectional mobile robot," in *Proc. IEEE Int. Symp. Ind. Electron.*, Jun. 2007, pp. 2161–2166.

[10] H. X. Araujo, A. G. S. Conceicao, J. Pitanga, and G. H. C. Oliveira, "Model predictive control based on LMIs applied to an omni-directional mobile robot," in *Proc. 18th World Congr. Int. Feder. Autom. Control*, 2011, vol. 18, pp. 8171–8176.

[11] H. Lim, Y. Kang, C. Kim, J. Kim, and B.-J. You, "Nonlinear model predictive controller design with obstacle avoidance for a mobile robot," in *Proc. IEEE/ASME Int. Conf. Mechatron. Embedd. Syst. Appl.*, Oct. 2008, pp. 494–499.

[12] K. Yeonsik and J. K. Hedrick, "Linear tracking for a fixed-wing UAV using nonlinear model predictive control," *IEEE Trans. Control Syst. Technol.*, vol. 1, no. 10, pp. 1202–1210, Sep. 2009.

[13] J. Maciejowski, *Predictive Control With Constraints*. London, U.K.: Pearson Education, 2002.

[14] C. Canudas de Wit, H. Olsson, K. J. Astrom, and P. Lischinsky, "A new model for control of systems with friction," *IEEE Trans. Autom. Control*, vol. 40, no. 3, pp. 419–425, Mar. 1995.

[15] B. Bona and M. Indri, "Friction compensation in robotics: An overview," in *Proc. 44th IEEE Conf. Decis. Control Eur. Control Conf.*, Dec. 2005, pp. 4360–4367.

[16] L. E. Ray, "Estimation of terrain forces and parameters for rigid-wheeled vehicles," *IEEE Trans. Robot.*, vol. 25, no. 3, pp. 717–726, Jun. 2009.

[17] T. H. Lee, K. K. Tan, and S. Huang, "Adaptive friction compensation with a dynamical friction model," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 1, pp. 133–140, Feb. 2011.

[18] L. Mostefai, M. Denai, and Y. Hori, "Robust tracking controller design with uncertain friction compensation based on a local modeling approach," *IEEE/ASME Trans. Mechatronics*, vol. 15, no. 5, pp. 746–756, Oct. 2010.

[19] B. Xu, C. Pradalier, A. Krebs, R. Siegwart, and F. Sun, "Composite control based on optimal torque control and adaptive Kriging control for the CRAB rover," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1752–1757.

[20] A. G. S. Conceicao, A. Moreira, and P. Costa, "Practical approach of modeling and parameters estimation for omnidirectional mobile robots," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 3, pp. 377–381, Jun. 2009.

[21] H. Olsson, K. J. Astrom, C. Canudas de Wit, M. Gafvert, and P. Lischinsky, "Friction models and friction compensation," *Eur. J. Contr.*, vol. 4, pp. 176–195, 1998.

[22] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawai, and H. Matsubara, "RoboCup: A challenge problem for AI and robotics," in *Proc. RoboCup-97: Robot Soccer World Cup I*, 1998, pp. 1–19.

[23] Y. Liu, J. J. Zhu, R. L. Williams II, and J. Wu, "Omni-directional mobile robot controller based on trajectory linearization," *Robot. Autonom. Syst.*, vol. 56, no. 5, pp. 461–479, May 2008.

[24] E. Camacho and C. Bordons, *Model Predictive Control*. New York, NY, USA: Springer-Verlag, 2004.

[25] N. Karmarker, "A new polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, pp. 373–395, Aug. 1984.

[26] A. G. Wills and W. P. Heath, "Interior-point algorithms for nonlinear model predictive control," in *Lecture Notes Control Inf. Sci.*, 2007, vol. 358, pp. 207–216.

[27] A. Shahzad and P. J. Goulart, "A new hot-start interior-point method for model predictive control," in *Proc. 18th World Congr. Int. Feder. Autom. Control*, 2011, vol. 18, pp. 2470–2475.

[28] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

[29] *Product Manual, v1.xAx - 802.15.4, Protocol for OEM RF Module Part Numbers: XB24-... -001, XBP24-... -001*, 1 ed., MaxStream Inc., Orem, UT, USA, 2006.

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

Júlio César Lins Barreto S. was born in 1987 in Aracaju, Brazil. He received the B.S. degree in electrical engineering with emphasis on control and automation in 2010, and the M.S. degree in 2011, both from the Federal University of Bahia, Salvador, Brazil, where he is currently working toward the Ph.D. degree in control and automation.

His main research interests include robotics, predictive control, robust control, Lyapunov functions, optimization theory, control systems, and automation.

André Gustavo Scolari Conceição received the B.S. and M.S. degrees in electrical engineering from the Pontifical Catholic University of Rio Grande do Sul, Porto Alegre, Brazil, in 2001 and 2004, respectively, and the Ph.D. degree in electrical and computer engineering from Oporto University, Oporto, Portugal, in 2007.

In 2009, he joined the Department of Electrical Engineering, Federal University of Bahia, Salvador, Brazil, as an Assistant Professor, and is currently a member of the Electrical Engineering Postgraduate Program. His main research interests include mobile robotics, multirobot systems, sensor fusion, state and parameter estimation, process control, and vehicle dynamics.

Carlos E. T. Dórea received the B.Sc. degree in electrical engineering from the Universidade Federal da Bahia (UFBA), Salvador, Brazil in 1990, the M.Sc. degree in electrical engineering from the Universidade Estadual de Campinas, Campinas, Brazil, in 1993, and the D.Sc. degree in automation from the Université Paul Sabatier, Toulouse, France, in 1997.

In 1998, he joined the UFBA as a Postdoctoral Fellow, where he became an Assistant Professor of control engineering in the Department of Electrical Engineering in 2001. In 2008, he spent a sabbatical year in the Department of Electrical Engineering, Katholieke Universiteit Leuven, Leuven, Belgium. Since 2010, he has been an Associate Professor in the Department of Computing Engineering and Automation, Universidade Federal do Rio Grande do Norte, Natal, Brazil. His main research interests include linear systems theory and design, set-invariance techniques for control of constrained systems, and predictive control applications.

Luciana Martinez received the B.Sc. degree in mathematics and the M.Sc. degree in computational mathematics from State University of São Paulo, São Paulo, Brazil, in 1994 and 1996, respectively, and the Ph.D. degree in electrical engineering from the State University of Campinas, Campinas, Brazil, in 2001.

She has been a Faculty Member with the Federal University of Bahia, Salvador, Brazil, since 2005, where she is currently an Assistant Professor in the Department of Electrical Engineering. Her research interests include optimization and operation research.

Edson Roberto de Pieri was born in Mogi Mirim, Brazil. He received the Statistics and Mathematics degrees in 1982 and 1983, respectively, and the M.Sc. degree in electrical engineering in 1987, all from the State University of Campinas, Campinas, Brazil. In 1991, he received the Doctoral degree from the Pierre et Marie Curie University, Paris, France. In 2004, he received a CAPES (Brazil) grant for a one-year stay at the Ecole Normale Supérieure de Cachan, Cachan, France.

Since 1992, he has been a Professor in the Department of Automation and Systems, Universidade Federal de Santa Catarina, Florianópolis, Brazil. His research interests include nonlinear control, automation, and robotics.