

Tracking-error model-based predictive control for mobile robots in real time

Gregor Klančar*, Igor Škrjanc

Laboratory of Modelling, Simulation and Control, Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, SI-1000 Ljubljana, Slovenia

Received 21 December 2005; received in revised form 4 January 2007; accepted 24 January 2007

Available online 3 February 2007

Abstract

In this paper, a model-predictive trajectory-tracking control applied to a mobile robot is presented. Linearized tracking-error dynamics is used to predict future system behavior and a control law is derived from a quadratic cost function penalizing the system tracking error and the control effort. Experimental results on a real mobile robot are presented and a comparison of the control obtained with that of a time-varying state-feedback controller is given. The proposed controller includes velocity and acceleration constraints to prevent the mobile robot from slipping and a Smith predictor is used to compensate for the vision-system dead-time. Some ideas for future work are also discussed.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Predictive control; Trajectory tracking; Mobile robot

1. Introduction

In recent years there has been an increasing amount of research on the subject of mobile robotics. Mobile robots are increasingly used in industry, in service robotics, for domestic needs (vacuum cleaners, lawn mowers, pets), in difficult-to-access or dangerous areas (space, army, nuclear-waste cleaning) and also for entertainment (robotic wars, robot soccer).

Several controllers were proposed for mobile robots with nonholonomic constraints, where the two main approaches to controlling mobile robots are posture stabilization and trajectory tracking. The aim of posture stabilization is to stabilize the robot to a reference point, while the aim of trajectory tracking is to have the robot follow a reference trajectory. For mobile robots trajectory tracking is easier to achieve than posture stabilization. This comes from the assumption that the wheel makes perfect contact with the ground, resulting in nonholonomic constraints, which means that not all the velocities are possible at a certain moment. An extensive review of nonholonomic control problems can be found in [7]. According to Brockett's condition [2] nonholonomic systems cannot be asymptotically stabilized around an equilibrium using smooth time-invariant feedback. Completely nonholonomic, driftless systems are controllable

in a nonlinear sense; therefore, asymptotic stabilization can be obtained using time-varying, discontinuous or hybrid control laws. An exponentially stable, discontinuous feedback controller was proposed by [3] and the point stabilization of mobile robots via state-space exact-feedback linearization using proposed coordinates was studied in [17].

Trajectory tracking is more natural for mobile robots. Usually, the reference trajectory is obtained by using a reference robot; therefore, all the kinematic constraints are implicitly considered by the reference trajectory. The control inputs are mostly obtained by a combination of feedforward inputs, calculated from reference trajectory, and feedback control law, as in [22,10,16,1]. Lyapunov stable time-varying state-tracking control laws were pioneered by [5,20,21], where the system's equations are linearized with respect to the reference trajectory, and by defining the desired parameters of the characteristic polynomial the controller parameters are calculated. The stabilization to the reference trajectory requires a nonzero motion condition. Many variations and improvements of this simple and effective state-tracking controller followed in latter research. An adaptive extension of this work was introduced in [18], where adaptive capabilities are included to increase the robustness to robot-model uncertainties. A fuzzy inference mechanism extension that compensates for environmental perturbations such as variable friction is proposed in [19]. A tracking control using modified input–output linearization providing a least-squares solution

* Corresponding author. Tel.: +386 1 4768764; fax: +386 1 4264631.
E-mail address: gregor.klancar@fe.uni-lj.si (G. Klančar).

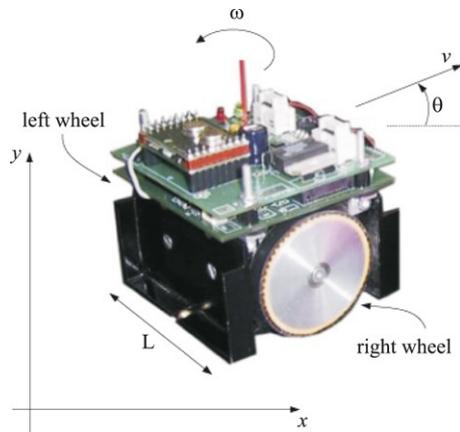


Fig. 1. The differentially driven mobile robot.

for a nonsquare system is dealt with in [6]. In [13] a trajectory-tracking state-feedback controller is combined with an observer that is used to estimate an unknown orientation error. Input–output linearization is used in [22] to follow reference trajectory dynamics considering a dynamic model. Some of the solutions to the controller design solve both the trajectory-tracking and posture-stabilization control problems, where the stabilization problem is usually converted to an equivalent tracking problem, as in [18,16,9]. In [9] a saturation feedback controller where saturation constraints of the velocity inputs are incorporated into the controller design is introduced. In [16] a dynamic feedback linearization technique is used to control a mobile robot platform.

Predictive control techniques are a very important area of research. In the field of mobile robotics predictive approaches to path tracking also seem to be very promising because the reference trajectory is known beforehand. Most model-based predictive controllers use a linear model of mobile-robot kinematics to predict future system outputs. In [15] a generalized predictive control is chosen to control a mobile robot, where a quadratic cost function penalizing the tracking errors and control effort is minimized. A generalized predictive controller using a Smith predictor to cope with an estimated system time delay is presented in [14]. In [8] a model-predictive control based on a linear, time-varying description of the system is used. The control law is again solved by an optimization of a cost function. The nonlinear predictive controller scheme for a path-tracking problem is proposed in [4]. Here, a multi-layer neural network is employed to model the nonlinear kinematic behavior of a mobile robot. However, the optimum solution of the control vector is still obtained by minimizing a cost function, like in previous studies.

This paper deals with a differentially driven mobile robot and trajectory-tracking control on a reference trajectory that is a smooth twice-differentiable function of time. The model-predictive control law is based on a linearized error dynamics model obtained around the reference trajectory. The main idea of the control law is to minimize the difference between the future trajectory-following errors of the robot and the reference robot with defined, desired dynamics. The proposed control law is analytically derived; therefore, it is computationally effective

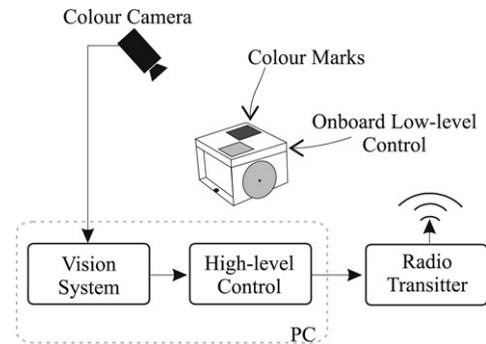


Fig. 2. System overview.

and can be easily used in fast real-time implementations. The main advantages over predictive control are an error model-based prediction and an explicitly obtained analytical control law. The model-predictive control obtained is compared to a well-known time-varying state-tracking control law [5,20,12], which is based on the literature review done in this work and works presented by [6,18,19], one of the most common and successful approaches in mobile robot tracking control. The design of the state-tracking control law in a discrete time domain is given. The experimental results for both control laws obtained for a real robot are evaluated and compared.

The remainder of the paper is organized as follows. In Section 2 is a description of the mobile robot, its control architecture and its kinematics. The concept of trajectory-tracking controller design, where the control strategy consists of feedforward and feedback actions, is introduced in Section 3. In Section 4 the proposed model predictive controller is derived. The experimental results for the predictive control obtained are presented in Section 5, and the conclusion is given in Section 6.

2. Mobile-robot control-system design

The control-system design proposed in this study, the experiments and the comparisons were performed on the small, two-wheeled, differentially driven mobile robot shown in Fig. 1.

2.1. Robot description and control architecture

The robot measures $7.5 \times 7.5 \times 7.5$ cm and weighs 0.6 kg. It contains a C167 microcontroller running at a 20 MHz clock, a 12 V battery supply, two powerful DC motors equipped with incremental encoders (512 pulses per revolution), and a gear reduction head.

The control of the mobile robot's motion is performed on two levels, as demonstrated in Fig. 2. The low-level control is in charge of controlling the robot's wheel speeds, while the high-level control determines the required robot speeds considering its first-order kinematics. This two-layer architecture is very common in practice because most mobile robots and manipulators usually do not allow the user to impose accelerations or torques at the inputs. It can also be viewed as a simplification to the problem as well as a more modular design approach.

The low-level control is implemented in the robot. Each wheel (motor) speed is controlled by a discrete PID controller

with a cycle time of 1 ms; this ensures that the robot drives with the required reference wheel speeds. These reference wheel speeds are obtained from the high-level control through a wireless connection. Finally, two power amplifiers are used to drive the motors with 33 kHz PWM signals.

High-level control is proposed in this study. It is implemented on a personal computer where the states of the mobile robots (posture) are obtained from a vision system with a 33 ms sample time. The calculated robot inputs (speeds) are sent via a wireless connection to the robot's low-level control.

2.2. Kinematics and driving constraints

The robot's architecture, together with its symbols, is shown in Fig. 1. The kinematic motion equations of the mobile robot are equivalent to those for a unicycle. Robots with such an architecture have a nonintegrable constraint in the form

$$A(q)\dot{q} = \dot{x} \sin \theta - \dot{y} \cos \theta \quad (1)$$

resulting from the assumption that the robot cannot slip in a lateral direction. In Eq. (1) $A(q)$ is the constraint matrix defined over the generalized coordinates $q(t) = [x(t) \ y(t) \ \theta(t)]^T$. The first-order kinematics model is obtained by expressing all the achievable velocities of the mobile robot as a linear combination of the vector fields $s_i(q)$ that span the null space of the matrix $A(q)$. The kinematics model then results in the following equation:

$$\begin{aligned} \dot{q}(t) &= [s_1(q) \ s_2(q)] \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} \\ &= \begin{bmatrix} \cos \theta(t) & 0 \\ \sin \theta(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} \end{aligned} \quad (2)$$

where $v(t)$ and $\omega(t)$ are the tangential and angular velocities of the platform in Fig. 1. The right and left velocities of the robot's wheels (needed for the low-level control) are then expressed as $v_R = v + \frac{\omega L}{2}$ and $v_L = v - \frac{\omega L}{2}$.

During low-level control the bounded velocity and acceleration constraints are considered. The robot's tangential and angular velocities are bounded with $v_{MAX} = \omega_M R$ and $\omega_{MAX} = 2\omega_M R/L$, where ω_M is the maximum angular velocity of the wheel and R is its radius. A saturation of the command velocities [16] that preserve the current curvature $\kappa = \frac{\omega}{v}$ is performed as

$$\sigma = \max\{|v|/v_{MAX}, |\omega|/\omega_{MAX}, 1\}$$

where the actual command velocities v_c and ω_c stand for

$$\begin{aligned} v_c &= \text{sign}(v)v_{MAX}, \quad \omega_c = \omega/\sigma; & \sigma &= |v|/v_{MAX} \\ v_c &= v/\sigma, \quad \omega_c = \text{sign}(\omega)\omega_{MAX}; & \sigma &= \omega/\omega_{MAX} \\ v_c &= v, \quad \omega_c = \omega; & \sigma &= 1. \end{aligned}$$

To prevent the mobile robot from slipping, the wheel's command velocities (v_{Rc} , v_{Lc}) are bounded with the allowable acceleration, as follows:

$$\begin{aligned} v_{Rc}(k) &= v_R(k); & |a_R| &\leq a_{MAX} \\ v_{Rc}(k) &= v_{Rc}(k-1) + \text{sign}(a_R)a_{MAX}T_s; & |a_R| &> a_{MAX} \end{aligned}$$

and

$$\begin{aligned} v_{Lc}(k) &= v_L(k); & |a_L| &\leq a_{MAX} \\ v_{Lc}(k) &= v_{Lc}(k-1) + \text{sign}(a_L)a_{MAX}T_s; & |a_L| &> a_{MAX} \end{aligned}$$

where the left- and right-hand wheel accelerations are computed as $a_R = \frac{v_R(k) - v_{Rc}(k-1)}{T_s}$ and $a_L = \frac{v_L(k) - v_{Lc}(k-1)}{T_s}$, and T_s is the sampling time. The maximum allowable acceleration, a_{MAX} , is determined experimentally so that the mobile robot never slips.

3. Definition of the trajectory-tracking problem

There are two basic control approaches to solving the mobile robot's motion task: stabilization to a fixed posture and tracking of the reference trajectory.

For nonholonomic systems, the trajectory-tracking problem is easier to solve and more natural than posture stabilization. According to Brockett's condition [2] asymptotic stability of a nonholonomic system to a fixed posture is only possible with a time-varying or discontinuous feedback. Stabilization, therefore, cannot be achieved by a continuous time-invariant feedback law.

In the case of a trajectory-tracking controller a linear time-varying system is obtained by approximate linearization around the trajectory. The linearization obtained is shown to be controllable [20,21,11,3] as long as the trajectory does not come to a stop, which implies that the system can be asymptotically stabilized by smooth linear or nonlinear feedback.

The rest of the paper deals with trajectory-tracking controller design where the control strategy combines a feedforward solution and a feedback action.

3.1. Feedforward control action

Open-loop control of feedforward control is an intuitive approach to steering nonholonomic systems. Having a feasible path that enables us to reach a desired posture, the feedforward control inputs $v_r(t)$ and $\omega_r(t)$ are derived using the kinematic model (2). For a given reference trajectory $(x_r(t), y_r(t))$ defined in a time interval $t \in [0, T]$ the feedforward control law is derived. However, the calculated robot inputs drive the robot on a desired path only if there are no disturbances and no initial state errors. The tangential velocity $v_r(t)$ is calculated as follows:

$$v_r(t) = \pm \sqrt{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \quad (3)$$

where the sign depends on the desired drive direction (+ for forward and – for reverse). The tangent angle of each point on the path is defined as

$$\theta_r(t) = \arctan2(\dot{y}_r(t), \dot{x}_r(t)) + k\pi \quad (4)$$

where $k = 0, 1$ defines the desired drive direction (0 for forward and 1 for reverse) and the function $\arctan2$ is a four-quadrant inverse tangent function. By calculating the time derivative of (4) the robot's angular velocity $\omega_r(t)$ is obtained:

$$\omega_r(t) = \frac{\dot{x}_r(t)\ddot{y}_r(t) - \dot{y}_r(t)\ddot{x}_r(t)}{\dot{x}_r^2(t) + \dot{y}_r^2(t)} = v_r(t)\kappa(t) \quad (5)$$

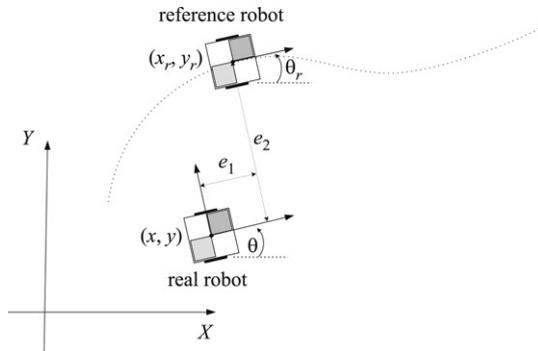


Fig. 3. Robot following error transformation.

where $\kappa(t)$ is the path curvature. By using relations (3) and (5) and the defined reference robot path $q_r(t) = [x_r(t), y_r(t), \theta_r(t)]^T$ the robot inputs $v_r(t)$ and $\omega_r(t)$ are calculated. The necessary condition in the path-design procedure is a twice-differentiable path and a nonzero tangential velocity $v_r(t) \neq 0$. If for some time t the tangential velocity is $v_r(t) = 0$, the robot rotates at a fixed point with the angular velocity $\omega_r(t)$. The angle $\theta_r(t)$ cannot be determined from Eq. (3) and therefore $\theta_r(t)$ must be given explicitly.

3.2. Feedback control action

When the robot is controlled to drive on a reference path, it usually has some state-following error. The state-tracking error $e(t) = [e_1(t) \ e_2(t) \ e_3(t)]^T$ expressed in the frame of the real robot, as shown in Fig. 3, reads

$$e = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} (q_r - q) \quad (6)$$

In Fig. 3 the reference robot is an imaginary robot that ideally follows the reference path. In contrast, the real robot (when compared to the reference robot) has some error when following the reference path. Therefore, the control algorithm should be designed to force the robot to follow the reference path precisely.

Taking into account the robot's kinematics (2) and deriving the relations (6) the following kinematic model is obtained:

$$\dot{e} = \begin{bmatrix} \cos e_3 & 0 \\ \sin e_3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} + \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix} u \quad (7)$$

where $u = [v \ \omega]^T$ is the velocity input vector and v_r, ω_r are already defined in Eqs. (3) and (5). The robot input vector u is further defined as the sum of the feedforward and feedback control actions, as follows:

$$u = u_F + u_B \quad (8)$$

where the feedforward input vector, u_F , is obtained by a nonlinear transformation of the reference inputs $u_F = [v_r \cos e_3 \omega_r]^T$ and the feedback input vector is $u_B = [u_{B1} \ u_{B2}]^T$, which is the output of the controller defined in Section 4 (see Fig. 4).

Using relation (8) and rewriting Eq. (7) results in the following tracking-error model:

$$\dot{e} = \begin{bmatrix} 0 & \omega & 0 \\ -\omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ \sin e_3 \\ 0 \end{bmatrix} \cdot v_r + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} u_B \quad (9)$$

Subsequently, by linearizing the error dynamics (9) around the reference trajectory ($e_1 = e_2 = e_3 = 0, u_{B1} = u_{B2} = 0$) the following linear model results:

$$\dot{e} = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} u_B \quad (10)$$

which is in the state-space form, $\dot{e} = A_c e + B_c u_B$. The controllability matrix $[B_c \ A_c B_c \ A_c^2 B_c]$ has full rank if either v_r or ω_r is nonzero, which is a sufficient condition for controllability only when the reference inputs v_r and ω_r are constant (linear and circular paths). In this case it is possible to stabilize the system with smooth static feedback. If the time-varying reference inputs v_r or ω_r are used the nonsingularity of the controllability gramian should be checked [20,12,16]. Although the system (2) with only one nonholonomic constraint (1) is completely nonholonomic and therefore controllable in a nonlinear sense (Chow's theorem), asymptotic stability with smooth static feedback, like with LTI systems, cannot be concluded. However, Brockett's condition [2] states that smooth stabilization of the system (2) is only possible with a time-varying feedback.

The schema of the control obtained is explained in Fig. 4. In the following, two approaches to the design of a trajectory-tracking controller are presented.

4. Design of the trajectory-tracking controller

To design the controller for trajectory tracking the system (10) will be written in discrete-time form as

$$e(k+1) = A e(k) + B u_B(k)$$

where $A \in \mathbb{R}^n \times \mathbb{R}^n$, n is the number of the state variables and $B \in \mathbb{R}^n \times \mathbb{R}^m$, and m is the number of input variables. The discrete matrices A and B can be obtained as follows:

$$\begin{aligned} A &= I + A_c T_s \\ B &= B_c T_s \end{aligned} \quad (11)$$

which is a good approximation for a short sampling time T_s .

4.1. Model-predictive control based on the robot-tracking-error model

The idea of the moving-horizon control concept is to find the control-variable values that minimize the receding-horizon quadratic cost function (in a certain interval denoted by h) based on the predicted robot-following error:

$$J(u_B, k) = \sum_{i=1}^h \epsilon^T(k, i) Q \epsilon(k, i) + u_B^T(k, i) R u_B(k, i) \quad (12)$$

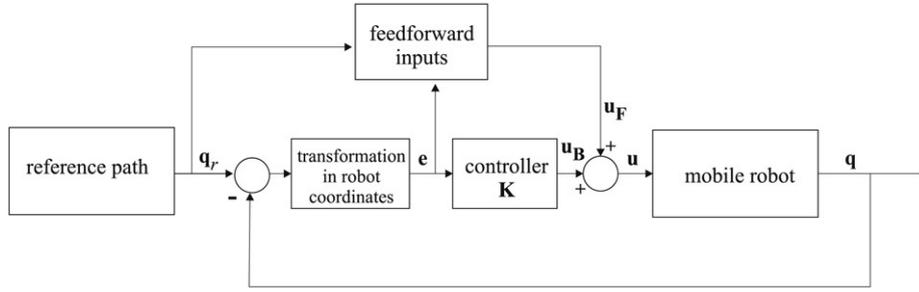


Fig. 4. Mobile-robot control schematic.

where $\epsilon(k, i) = e_r(k + i) - e(k + i|k)$ and $e_r(k + i)$ and $e(k + i|k)$ stand for the reference robot-following trajectory and the robot-following error, respectively, and Q and R stand for the weighting matrices where $Q \in \mathbb{R}^n \times \mathbb{R}^n$ and $R \in \mathbb{R}^m \times \mathbb{R}^m$, with $Q \geq 0$ and $R \geq 0$.

4.1.1. Output prediction in the discrete-time framework

In the moving time frame the model-output prediction at the time instant h can be written as

$$e(k + h|k) = \prod_{j=1}^{h-1} A(k + j|k)e(k) + \sum_{i=1}^h \left(\prod_{j=i}^{h-1} A(k + j|k) \right) \times B(k + i - 1|k)u_B(k + i - 1) + B(k + h - 1|k)u_B(k + h - 1). \quad (13)$$

Defining the robot-tracking prediction-error vector

$$E^*(k) = \left[e(k + 1|k)^T \ e(k + 2|k)^T \ \dots \ e(k + h|k)^T \right]^T.$$

where $E^* \in \mathbb{R}^{n \cdot h}$ for the whole interval of the observation (h) and the control vector

$$U_B(k) = \left[u_B^T(k) \ u_B^T(k + 1) \ \dots \ u_B^T(k + h - 1) \right]^T$$

and

$$A(k, i) = \prod_{j=i}^{h-1} A(k + j|k)$$

the robot-tracking prediction-error vector is written in the form

$$E^*(k) = F(k)e(k) + G(k)U_B(k) \quad (14)$$

where

$$F(k) = [A(k|k) \ A(k + 1|k)A(k|k) \ \dots \ A(k, 0)]^T, \quad (15)$$

and

$$G(k) = \begin{bmatrix} B(k|k) & 0 & \dots & 0 \\ A(k + 1|k)B(k|k) & B(k + 1|k) & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ A(k, 1)B(k|k) & A(k, 2)B(k + 1|k) & \dots & B(k + h - 1|k) \end{bmatrix} \quad (16)$$

and $F(k) \in \mathbb{R}^{n \cdot h} \times \mathbb{R}^n$, $G(k) \in \mathbb{R}^{n \cdot h} \times \mathbb{R}^{m \cdot h}$.

The objective of the control law is to drive the predicted robot trajectory as close as possible to the future reference trajectory, i.e., to track the reference trajectory. This implies that the future reference signal needs to be known. Let us define the reference error-tracking trajectory in state-space as

$$e_r(k + i) = A_r^i e(k) \quad (17)$$

for $i = 1, \dots, h$. This means that the future control error should decrease according to the dynamics defined with the reference model matrix A_r . Defining the robot reference-tracking-error vector

$$E_r^*(k) = \left[e_r(k + 1)^T \ e_r(k + 2)^T \ \dots \ e_r(k + h)^T \right]^T$$

where $E_r^* \in \mathbb{R}^{n \cdot h}$ for the whole interval of observation (h), the following is obtained:

$$E_r^*(k) = F_r e(k) \quad (18)$$

where

$$F_r = \left[A_r \ A_r^2 \ \dots \ A_r^h \right]^T, \quad (19)$$

and $F_r \in \mathbb{R}^{n \cdot h} \times \mathbb{R}^n$.

4.1.2. Control law

The idea of MPC is to minimize the difference between the predicted robot-trajectory error and the reference robot-trajectory error in a certain predicted interval.

The cost function is, in accordance with the above notation, now written as

$$J(U_B) = (E_r^* - E^*)^T \bar{Q} (E_r^* - E^*) + U_B^T \bar{R} U_B \quad (20)$$

The control law is obtained by minimizing the cost function as follows:

$$\frac{\partial J}{\partial U_B} = -2\bar{Q}G^T E_r^* + 2G^T \bar{Q} E^* + 2\bar{R}U_B = 0 \quad (21)$$

and the control vector becomes

$$U_B(k) = \left(G^T \bar{Q} G + \bar{R} \right)^{-1} G^T \bar{Q} (F_r - F) e(k). \quad (22)$$

where

$$\bar{Q} = \begin{bmatrix} Q & 0 & \dots & 0 \\ 0 & Q & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q \end{bmatrix} \quad (23)$$

and

$$\bar{R} = \begin{bmatrix} R & 0 & \dots & 0 \\ 0 & R & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & R \end{bmatrix} \quad (24)$$

This means that $\bar{Q} \in \mathbb{R}^{n \cdot h} \times \mathbb{R}^{n \cdot h}$ and $\bar{R} \in \mathbb{R}^{m \cdot h} \times \mathbb{R}^{m \cdot h}$.

Let us define the first m rows of the matrix $(G^T \bar{Q} G + \bar{R})^{-1} G^T \bar{Q} (F_r - F) \in \mathbb{R}^{m \cdot h} \times \mathbb{R}^n$ as K_{mpc} . Now the feedback-control law of the model-predictive control is given by

$$u_B(k) = K_{mpc} \cdot e(k) \quad (25)$$

with $K_{mpc} \in \mathbb{R}^m \times \mathbb{R}^n$. The schematic of the control obtained is explained in Fig. 4.

4.2. State-tracking controller

The proposed predictive controller is compared to a well-known state-tracking controller whose design can be found in [5,20,12]. Some basics steps of the design in a discrete notation are given in what follows.

The linear state-tracking controller for the linearized error dynamics (10) reads

$$u_B(k) = K_s \cdot e(k) \quad (26)$$

where the gain matrix $K_s \in \mathbb{R}^m \times \mathbb{R}^n$ is defined as follows:

$$K_s = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & \text{sign}(v_r(k))k_2 & k_3 \end{bmatrix} \quad (27)$$

An intuitive explanation of the gain matrix's structure (27) can also be obtained from observing Fig. 3. To reduce the error in the driving direction e_1 the tangential robot velocity should be changed accordingly. Similarly, the orientation error e_3 can be efficiently manipulated by the robot's angular speed. Finally, the error orthogonal to the driving direction can be reduced by changing the angular velocity. At the same time the robot's drive direction (forward or backward) should also be considered.

The controller gains (k_1, k_2, k_3) are determined by comparing the actual and the desired characteristic polynomial equations. The desired characteristic polynomial takes the following form:

$$(z + z_1)(z + z_2)(z + z_3) \quad (28)$$

where $z_1 = -e^{-2\zeta\omega_n T_s}$, $z_{2,3} = -e^{-\zeta\omega_n T_s \pm i\omega_n \sqrt{1-\zeta^2} T_s}$, the desired damping coefficient $\zeta \in (0, 1)$ and the characteristic frequency $\omega_n > 0$, are selected.

The characteristic polynomial of a closed loop with the controller (27) is

$$\begin{aligned} \det(zI - A + BK_s) &= z^3 + (T_s k_1 - 3 + T_s k_3) z^2 \\ &+ \left(T_s^2 v_r k_2 + T_s^2 \omega_r^2 - 2 T_s k_3 - 2 T_s k_1 + T_s^2 k_1 k_3 + 3 \right) z \\ &- T_s^2 k_1 k_3 + T_s^3 k_1 v_r k_2 - 1 - T_s^2 \omega_r^2 + T_s^3 \omega_r^2 k_3 \\ &+ T_s k_3 - T_s^2 v_r k_2 + T_s k_1. \end{aligned} \quad (29)$$

Comparing coefficients at the same power of z in Eqs. (28) and (29) results in

$$\begin{aligned} T_s k_1 + T_s k_3 - 3 &= z_1 + z_2 + z_3 \\ T_s^2 v_r k_2 + T_s^2 \omega_r^2 - 2 T_s k_3 - 2 T_s k_1 + T_s^2 k_1 k_3 + 3 \\ &= z_1 z_2 + (z_1 + z_2) z_3 \\ -T_s^2 k_1 k_3 + T_s^3 k_1 v_r k_2 - 1 - T_s^2 \omega_r^2 + T_s^3 \omega_r^2 k_3 \\ &+ T_s k_3 - T_s^2 v_r k_2 + T_s k_1 = z_1 z_2 z_3. \end{aligned} \quad (30)$$

Let us find the solution of Eq. (30) in the form $k_1 = k_3$:

$$k_1 = k_3 = \frac{z_{123} + 3}{2T_s} \quad (31)$$

and k_2 is then determined from the second row of (30) as

$$k_2 = \frac{-1/4 (z_{123} + 3)^2 - T_s^2 \omega_r^2 + 2 z_{123} + 3 + z_1 z_2 + z_3 z_1 + z_3 z_2}{T_s^2 v_r} \quad (32)$$

or from the third row of (30) as

$$k_2' = \frac{1/4 (z_{123} + 3)^2 - z_{123} - 2 + T_s^2 \omega_r^2 - 1/2 T_s^2 \omega_r^2 (z_{123} + 3) + z_1 z_2 z_3}{1/2 T_s^2 v_r (z_{123} + 1)} \quad (33)$$

where $z_{123} = z_1 + z_2 + z_3$.

For a short sampling time T_s (see Eq. (11)) the calculated gains limit to final gains k_1^* , k_2^* and k_3^* . Eq. (31) limits to k_1^* and k_3^* , while both Eqs. (32) and (33) limit to the same expression k_2^* as follows:

$$\begin{aligned} k_1^* &= \lim_{T_s \rightarrow 0} k_1 = 2\zeta\omega_n \\ k_2^* &= \lim_{T_s \rightarrow 0} k_2 = \lim_{T_s \rightarrow 0} k_2' = \frac{\omega_n^2 - \omega_r(k)^2}{|v_r(k)|} \\ k_3^* &= \lim_{T_s \rightarrow 0} k_3 = 2\zeta\omega_n \end{aligned} \quad (34)$$

where ω_n should be larger than the maximum-allowed robot angular velocity, $\omega_n \geq \omega_{rMAX}$. When v_r is close to zero, k_2 goes to infinity and therefore a gain scheduling [10] should be chosen for k_2^* as $k_2^* = g \cdot |v_r(k)|$. The system's characteristic frequency becomes

$$\omega_n(k) = \sqrt{\omega_r^2(k) + g v_r^2(k)} \quad (35)$$

and the final controller gains are

$$\begin{aligned} k_1^* &= k_3^* = 2\zeta\omega_n(k) \\ k_2^* &= g \cdot |v_r(k)|. \end{aligned} \quad (36)$$

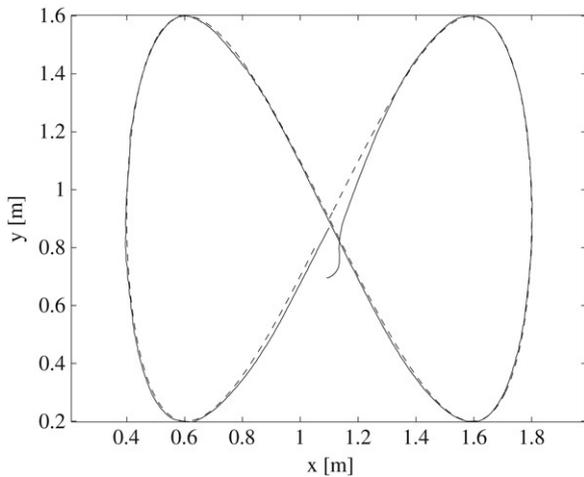


Fig. 5. Trajectory tracking with the model-predictive controller: robot path (—), reference path (---).

The tuning parameter $g > 0$ gives an additional freedom in controller design. The control obtained has a similar structure to that in [12,5,21]. The controller gains approach zero when the trajectory stops, at which time the robot is not controllable.

Even if the controller gains from Eqs. (31) and (32) are chosen to give stable and constant poles of the closed-loop system, the controller is still nonlinear and time varying. Therefore, the asymptotic stability is not guaranteed and should be checked with a Lyapunov stability analysis, as shown in [5, 12].

5. Experimental results

The experimental results were obtained on the real mobile-robot platform explained in Fig. 2. Two different design procedures which result in similar controller structures, given in Eqs. (25) and (26), are tested and compared. To ensure a fair comparison, a similar dynamics, which gives comparable actuator actions for the two controllers, was designed. The first control law is obtained by the proposed model-predictive control, and the second by the state-tracking controller, common in the literature. In both experiments the maximum control velocities and accelerations were limited, as explained in Section 2. The maximum allowed tangential velocity and angular velocity were $v_{\text{MAX}} = 0.5 \text{ m/s}$ and $\omega_{\text{MAX}} = 13 \text{ rad/s}$, while the maximum allowed tangential wheel acceleration was $a_{\text{MAX}} = 3 \text{ m/s}^2$. Additionally, the system delay, D , mainly originating from the vision system, was compensated. As shown in Fig. 2 the state measurements are obtained from a color camera and computer vision algorithm. The system delay is therefore caused by picture grabbing hardware and by the computationally demanding computer vision estimation algorithm. Other sources of system delay are the computational time for the control algorithms and the wireless connection. From experiments the estimated common system delay is $D = 2T_s$. The undelayed system output \bar{q} can be estimated from the delayed system output q and from the difference between the undelayed and delayed model outputs using the same inputs as

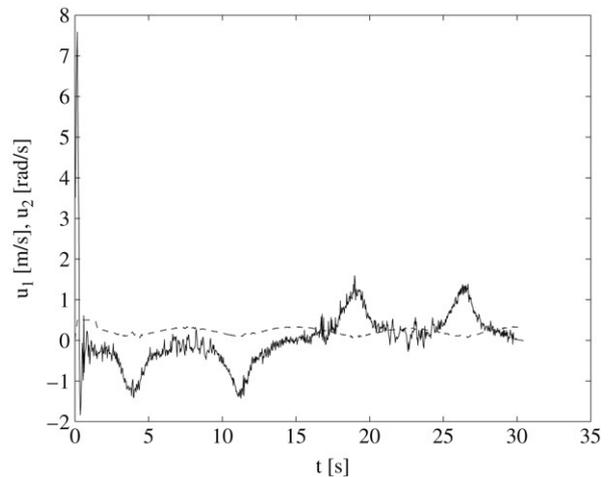


Fig. 6. Trajectory tracking with the model-predictive controller: tangential velocity v (---) and angular velocity ω (—).

the system, i.e.,

$$\bar{q}(k) = q(k) + q_m(k) - q_m(k - D)$$

where $q_m(k)$ is the output of the simulated system model without delay and $q_m(k - D)$ is the output of the simulated system model including the estimated system delay.

The reference trajectory for the experiments in Figs. 5 and 7 is defined by

$$x_r(t) = 1.1 + 0.7 \sin\left(\frac{2\pi t}{30}\right),$$

$$y_r(t) = 0.9 + 0.7 \sin\left(\frac{4\pi t}{30}\right)$$

where $t \in [0, 30] \text{ s}$. In both experiments the robot starts with an initial state error according to the reference trajectory (it does not start with the correct orientation and position).

In the first experiment the mobile robot is controlled with the proposed model-predictive controller (Figs. 5 and 6), where the controller properties and dynamics are defined by the parameters of the control. The reference model matrix is $A_r = I_{3 \times 3} \cdot 0.65$, and the weighting matrices are

$$Q = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 40 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}, \quad R = I_{2 \times 2} \times 10^{-3}.$$

The ratio of the diagonal elements in Q determines the sensitivity of the resulting controller to a certain error. A higher value of the diagonal element increases the sensitivity (gain) to the corresponding error. In the present case the control for the error in the lateral direction of driving has the highest weight, while the control for the orientation error has the lowest sensitivity. Similarly, the diagonal elements in R define the energy of the input-velocity signals; the lowest value of the elements results in more energy-consuming control. The coincidence horizon h influences the gain of the predictive control. A smaller value of h results in faster control dynamics, which consequently affects the noise propagation through the loop. In the experiment, h is chosen to be $h = 4$.

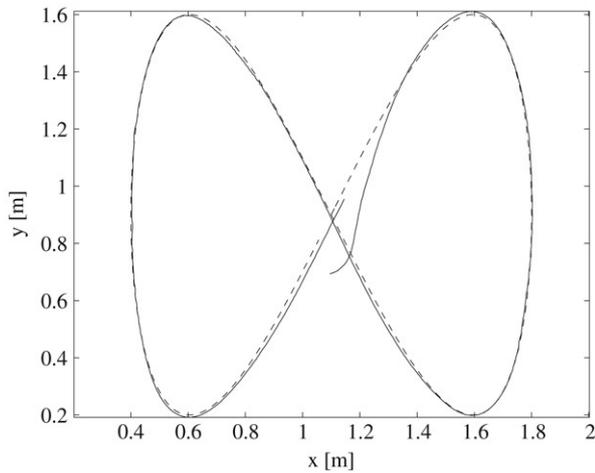


Fig. 7. Trajectory tracking with the state-tracking controller: robot path (—), reference path (- -).

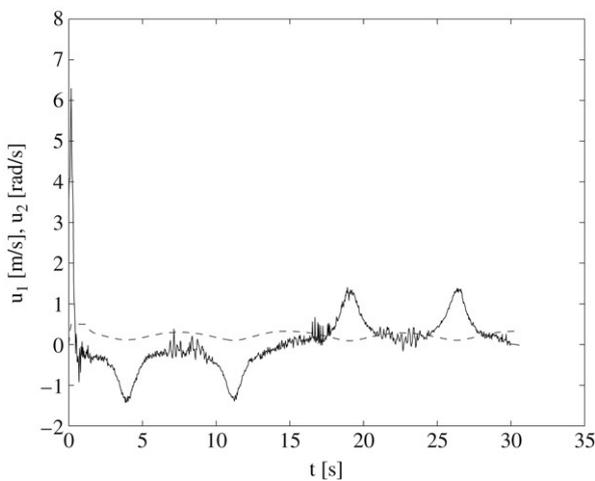


Fig. 8. Trajectory tracking with the state-tracking controller: tangential velocity v (- -) and angular velocity ω (—).

The resulting mobile-robot trajectory tracking, obtained by the proposed predictive controller, is shown in Fig. 5, while the velocity inputs are given in Fig. 6. As shown in Figs. 5 and 6, the model-predictive control exhibits good performance despite the vision-system delay (approximately compensated) and the noisy data, where the noise is mostly present in the orientation data. Other sources of disturbances the system is subjected to during experiments include sensor distortion such as wrong pose estimation (outliers; 2% of all measurements) and camera distortion corrections (perspective and radial distortion). Mobile robot wheel slipping disturbance is prevented by limiting allowable accelerations (see Section 2.2).

5.1. Comparison to the state-tracking controller

The proposed predictive controller is compared to the state-tracking controller (Figs. 7 and 8), with the gain matrix defined in Eq. (36), where the tuning parameters are selected as $\zeta = 0.7$ and $g = 60$. The performance of the obtained trajectory tracking in Figs. 7 and 8 with the state-tracking controller is also

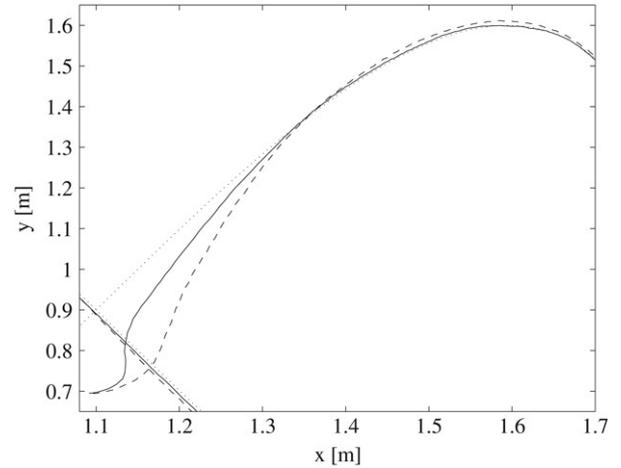


Fig. 9. Detailed view of the initial time response: model-predictive control (—), state-tracking control (- -) and reference path (· · ·).

good. The quality of the state-tracking control is comparable to that of the model-predictive control, although better results are obtained with model-predictive control, especially during the initial state-error response; the comparison is given in detail in Fig. 9.

The mobile robot converges faster to the reference trajectory in the case of model-predictive control. Better results are also obtained when the robot is driving close to the reference; tracking errors are lower during model-predictive control. The sum-square error (SSE) for each component of the state error ($q - q_r$) for model-predictive control is

$$SSE_{mpc} = (0.08 \quad 1.71 \quad 11.34)$$

and for the state-tracking controller it is

$$SSE_s = (0.05 \quad 1.86 \quad 14.11)$$

which, like for Fig. 9, favors model predictive control.

The model-predictive controller gives better control results, which is to be expected because of the more complex control structure, and taking into account future values of the reference, not only the current value, as with the state-tracking controller. As a consequence a more appropriate gain matrix (25), especially during a variable (nonconstant) reference trajectory, is obtained. Another property is the more flexible control structure relating to the desired control properties.

The model-predictive controller seems to require more tuning effort as it has more parameters than the state-tracking controller, which has only two. The smaller number of parameters appears more attractive. However, a reasonable control performance of the MPC can easily be obtained by changing the ratio of the weighting matrices Q and R . This also gives the designer the possibility of obtaining more optimized control, which is more difficult to obtain in the case of the state-tracking controller.

Both the presented and compared controllers work well when the mobile robot is close to the reference. For the case when the robot is far from the reference trajectory the controllers presented are not suitable because of large control

actions resulting from large errors. A useful upgrade in such a situation are the so-called landing curves, which allow the controller to drive the robot efficiently to the reference.

Compared with other related predictive control systems (discussed in Section 1) the main advantage of the approach presented is the analytically derived control law which significantly lowers the computational burden and enables its usage also in fast real-time implementations. However similar predictive control systems usually require some optimization technique to obtain the control vector, which results in more demanding computational algorithms less appropriate in fast real-time implementations.

6. Conclusion

The model-predictive trajectory-tracking control of a mobile robot is presented in this paper. The proposed control law minimizes the quadratic cost function consisting of tracking errors and control effort. The solution for the control is analytically derived, which enables fast real-time implementations. The proposed model-predictive control was tested on real mobile robots and the experimental results obtained were compared to a time-varying state-tracking controller. Both controllers presented work well when the mobile robot is close to the reference, although the model-predictive controller gives better control results, which is to be expected because of the more complex control structure taking into account future values of the reference, not only the current value, as with the state-tracking controller. Another property is its flexible control structure relating to the desired control properties.

The control system presented is applicable also to a large scale mobile robot for tracking planned trajectories in complex environments with obstacles. Our future plans include the usage of the controller obtained on a Pioneer 3-AT platform. Future improvements will focus on an increased robustness of the algorithm presented to larger tracking errors, mainly resulting from the wrong initial robot posture. The concept of landing curves, which guaranties an exponential convergence to the reference trajectory, will be included. Even better control would come from obtaining a more accurate model for tracking the error dynamics, which will be obtained by using a velocity linearization technique.

Acknowledgment

The authors would like to acknowledge the Slovenian Research Agency under CRP MIR M2-0116 project “Mobile Robot System for Reconnaissance, Research and Rescue” for funding this work.

References

- [1] A. Balluchi, A. Bicchi, A. Balestrino, G. Casalino, Path Tracking Control for Dubin's Cars, in: Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, 1996, pp. 3123–3128.
- [2] R.W. Brockett, Asymptotic stability and feedback stabilization, in: R.W. Brockett, R.S. Millman, H.J. Sussmann (Eds.), Differential Geometric Control Theory, Birkhäuser, Boston, MA, 1983, pp. 181–191.
- [3] C. Canudas de Wit, O.J. Sordalen, Exponential Stabilization of Mobile Robots with Nonholonomic Constraints, IEEE Transactions on Automatic Control 37 (11) (1992) 1791–1797.
- [4] D. Gu, H. Hu, Neural predictive control for a car-like mobile robot, Robotics and Autonomous Systems 39 (2) (2002) 73–86.
- [5] Y. Kanayama, Y. Kimura, F. Miyazaki, T. Noguchi, A stable tracking control method for an autonomous mobile robot, in: Proceedings of the 1990 IEEE International Conference on Robotics and Automation, vol. 1, Cincinnati, OH, 1990, pp. 384–389.
- [6] D.-H. Kim, J.-H. Oh, Tracking control of a two-wheeled mobile robot using input–output linearization, Control Engineering Practice 7 (3) (1999) 369–373.
- [7] I. Kolmanovsky, N.H. McClamroch, Developments in Nonholonomic Control Problems, IEEE Control Systems 15 (6) (1995) 20–36.
- [8] F. Kühne, W.F. Lages, J.M. Gomes da Silva Jr., Model predictive control of a mobile robot using linearization, in: Proceedings of Mechatronics and Robotics 2004, Aachen, Germany, 2004.
- [9] T.C. Lee, K.T. Song, C.H. Lee, C.C. Teng, Tracking control of unicycle-modeled mobile robots using a saturation feedback controller, IEEE Transactions on Control Systems Technology 9 (2) (2001) 305–318.
- [10] A. Luca, G. Oriolo, Modelling and control of nonholonomic mechanical systems, in: J. Angeles, A. Kecskemethy (Eds.), Kinematics and Dynamics of Multi-Body Systems, Springer-Verlag, Wien, 1995.
- [11] A. Luca, G. Oriolo, C. Samson, Feedback control of a nonholonomic car-like robot, in: J.P. Laumond (Ed.), Robot Motion Planning and Control, in: Lectures Notes in Control and Information Sciences, vol. 229, Springer-Verlag, London, 1998, pp. 171–253.
- [12] A. Luca, G. Oriolo, M. Vendittelli, Control of wheeled mobile robots: An experimental overview, in: S. Nicosia, B. Siciliano, A. Bicchi, P. Valigi, (Eds.) RAMSETE — Articulated and Mobile Robotics for Services and Technologies, Springer-Verlag, 2001.
- [13] S.P.M. Noijen, P.F. Lambrechts, H. Nijmeijer, An observer–controller combination for a unicycle mobile robot, International Journal of Control 78 (2) (2005) 81–87.
- [14] J.E. Normey-Rico, J. Gomez-Ortega, E.F. Camacho, A Smith-predictor-based generalised predictive controller for mobile robot path-tracking, Control Engineering Practice 7 (6) (1999) 729–740.
- [15] A. Ollero, O. Amidi, Predictive Path Tracking of Mobile robots. Application to the CMU Navlab, in: Proceedings of 5th International Conference on Advanced Robotics, Robots in Unstructured Environments, ICAR '91, vol. 2, Pisa, Italy, 1991, pp. 1081–1086.
- [16] G. Oriolo, A. Luca, M. Vandittelli, WMR Control Via Dynamic Feedback Linearization: Design, Implementation, and Experimental Validation, IEEE Transactions on Control Systems Technology 10 (6) (2002) 835–852.
- [17] K. Park, H. Chung, J.G. Lee, Point stabilization of mobile robots via state-space exact feedback linearization, Robotics and Computer Integrated Manufacturing 16 (2000) 353–363.
- [18] F. Pourboghrat, M.P. Karlsson, Adaptive control of dynamic mobile robots with nonholonomic constraints, Computers and Electrical Engineering 28 (2002) 241–253.
- [19] F.M. Raimondi, M. Melluso, A new fuzzy dynamics controller for autonomous vehicles with nonholonomic constraints, Robotics and Autonomous Systems 52 (2005) 115–131.
- [20] C. Samson, K. Ait-abderrahim, Feedback control of a nonholonomic wheeled cart in Cartesian space, in: Proceedings of the 1991 IEEE International Conference on Robotics and Automation, vol. 2, Sacramento, CA, 1991, pp. 1136–1141.
- [21] C. Samson, Time-varying feedback stabilization of car like wheeled mobile robot, International Journal of Robotics Research 12 (1) (1993) 55–64.
- [22] N. Sarkar, X. Yun, V. Kumar, Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robot, The International Journal of Robotic Research 13 (1) (1994) 55–69.



Gregor Klančar received his B.Sc. degree in 1999 from the Faculty of Electrical Engineering of the University of Ljubljana, Slovenia, where he is currently employed as a member of the national young researcher scheme. His research interests are in the area of fault diagnosis methods, multiple vehicle coordinated control and mobile robotics.



Igor Škrjanc received the B.Sc., M.Sc., and Ph.D. degrees, all in electrical engineering, from the Faculty of Electrical and Computer Engineering, University of Ljubljana, Slovenia, in 1988, 1991, and 1996, respectively. He is currently an Associate Professor with the same faculty. His main research interests are in adaptive, predictive, fuzzy, and fuzzy adaptive control systems.