



Collision-free control of an omni-directional vehicle

Mirosław Galicki*

Faculty of Mechanical Engineering, University of Zielona Góra, Podgórska 50, 65–246 Zielona Góra, Poland
 Institute of Medical Statistics, Computer Sciences and Documentation, Friedrich Schiller University Jena, Bachstrasse 18, D–07740 Jena, Germany

ARTICLE INFO

Article history:

Received 9 July 2007

Received in revised form

17 June 2009

Accepted 22 June 2009

Available online 26 June 2009

Keywords:

An omni-directional vehicle

Collision-free trajectory

Lyapunov stability

ABSTRACT

This study addresses the problem of controlling an omni-directional vehicle with both state and control dependent constraints. The task of the vehicle is to attain its desired final position given in the task space. The control constraints resulting from the physical abilities of actuators driving the vehicle wheels are also taken into account during the robot movement. The problem of collision avoidance is solved here based on an exterior penalty function approach which results in smooth vehicle velocities near obstacles. Provided that, a solution to the aforementioned vehicle task exists, the Lyapunov stability theory is used to derive the control scheme. The numerical simulation results carried out for the omni-directional vehicle operating in both a constraint-free task space and task space including obstacles, illustrate the performance of the proposed controllers.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Omni-directional vehicles have attracted a lot of interest recently due to useful practical tasks such as robot fork trucks handling palletes in factories or obstacle avoidance. Based on the location of the load (pallet), a trajectory of forkvehicle must be generated which ends precisely in front of, and aligned with, the fork holes (state equality constraints). Moreover, the vehicle velocity should equal zero at the terminal point. Another example is the problem of both staying on the road and avoiding an obstacle (state inequality constraints) by the robot. Finally, if a time decrease in the vehicle work cycle results in an acceleration of a technological process, then it is economically attractive to plan near time-optimal motions in such a case. By accomplishment of the aforementioned tasks, various constraints have to be taken into account, e.g. vehicle wheel velocities, collision avoidance in a task space or the simplicity of control realizations. In general, such tasks do not provide unique solutions. Consequently, some objective criteria should be specified to solve the vehicle tasks uniquely. Minimization of the performance time is mostly considered in the literature. One may distinguish several approaches in this context. In the recent work [1], a kinematic model of the vehicle and bounds on the velocities of the wheels were investigated

in the time-optimal control problem in the task space without obstacles. The authors from [1] have shown that the time-optimal trajectories between any pair of configurations in an obstacle-free task space consist of spins in place, circular arcs and tangent trajectories which are straight lines parallel to the wheel axes. The controls, as being angular wheel velocities generate usually non-zero final vehicle velocity which may be practically undesirable. Moreover, they are discontinuous and bang-bang. Therefore, there is no feedback control freedom left for these wheels to take care of disturbances or modelling discrepancies. Furthermore, discontinuities occurring at the switching points of optimal controls are unrealizable due to non-negligible motor dynamics. Nevertheless, work [1] does not address the problem of determining which of the classified trajectories is optimal for a particular pair of initial and final configurations. In work [2], an algorithm for numerical computing optimal trajectories for a bounded acceleration model of the omni-directional robot is presented. Using optimal control theory, the authors from [2] have constructed near time-optimal paths based on the bang-bang controls. Their methods generate omni-directional robot trajectories subject to complicated dynamics and actuator models. In works [3–5], an algorithm has been offered which solves the trajectory generation problem between arbitrary boundary states by iterative linearizing and inverting the equations of motion. By using a parametric trajectory model, the authors from [3–5] have converted the optimal control formulation into an equivalent nonlinear programming problem. In papers [6–8], time-optimal trajectories for both a bounded velocity [6] and bounded acceleration differential drive robots [7,8] in the unobstructed plane, have been studied. A near time-optimal control trajectory generator is presented in work [9], which solves

* Corresponding address: Institute of Medical Statistics, Computer Sciences and Documentation, Friedrich Schiller University Jena, Bachstrasse 18, D–07740 Jena, Germany. Tel.: +49 3641 934135; fax: +49 3641 933200.

E-mail addresses: galicki@imsid.uni-jena.de, miroslav.galicki@mti.uni-jena.de.

eleven first-order differential equations subject to state constraints. A real-time trajectory generation algorithm for differentially flat systems is proposed in work [10], where an approximation of non-linear constraints are replaced by linear inequality constraints. In work [11], it is shown that in the presence of obstacles, shortest paths may not exist between certain configurations of steered cars. Furthermore, in addition to the straight lines and circular arcs of minimum radius, the shortest paths may also contain segments that follow the boundaries of obstacles. Using the geometric technique, an algorithm was developed in work [12] to obtain the shortest nonholonomic distance from a car-like robot to any point on an obstacle. A numerical algorithm for finding a kinematically feasible path for a nonholonomic system in the presence of obstacles is presented in work [13]. The collision-free trajectory is yielded in [13] by solving a suitable non-linear least squares problem. An obstacle avoidance motion planning algorithm, based on a discontinuous feedback control and exploiting a navigation function given in a robot configuration space can be found in [14].

This paper presents an approach to the problem of controlling an omni-directional vehicle so that it attains a desired location (posture), and simultaneously avoids collisions with obstacles in the task space. In addition, the angular velocity limits (control inequality constraints) imposed on the vehicle wheels are taken into account. To the best of the author's knowledge, a constructive and efficient solution to this problem was not known before although it seems to be important in real-time applications. Provided that, a solution to the control problem of the omni-directional vehicle exists, the Lyapunov stability theory is used to derive the control law. Furthermore, it is also shown here how through the use of exterior penalty function method [15], the collision avoidance constraints are incorporated in our controller. Potential function methods widely used in the robotics literature [16–21] to tackle collision avoidance problems have several known disadvantages as compared to exterior penalty functions. On the other hand, exterior penalty function methods take into account only active collision avoidance constraints and generate bounded signals even on boundaries of obstacles. By a suitable choice of a non-linear mapping in our controller, vehicle controls become at least smooth functions even for collision avoidance tasks, leave room for feedback actions and retain the structure of a bang–bang solution in a non-singular case, as the numerical examples presented in Section 4 show. Although bounded velocity models capture the kinematics of a vehicle, but not its dynamics, our approach provides also bounded and at least smooth angular accelerations. Besides, the structure of the controllers proposed is much simpler as those obtained based on both the optimal control theory and vehicle dynamic equations [2]. Dealing mainly in our work with vehicle kinematics (instead of its dynamics) has also been dictated by the fact of comparing the theoretical results from recent work [1] with these presented in our study. Furthermore, we also answer the question stated for omni-directional vehicles in [1], that near optimal trajectories among obstacles consist of segments of obstacle-free trajectories and segments that follow the boundary of obstacles. On account of the fact, that the control scheme proposed here is implemented at the position level, a position controller is assumed to be available which closely tracks any (reference) desired trajectory provided by our motion controllers. The paper is organized as follows. Section 2 formulates the vehicle task to be accomplished in a task space. Section 3 describes how to employ both the Lyapunov stability theory and exterior penalty functions to determine the vehicle collision-free motion (if it exists). Section 4 provides us with a computer example of generating vehicle controls in a task space both without and with obstacles for an exemplary omni-directional vehicle. Finally, some conclusions are drawn in Section 5.

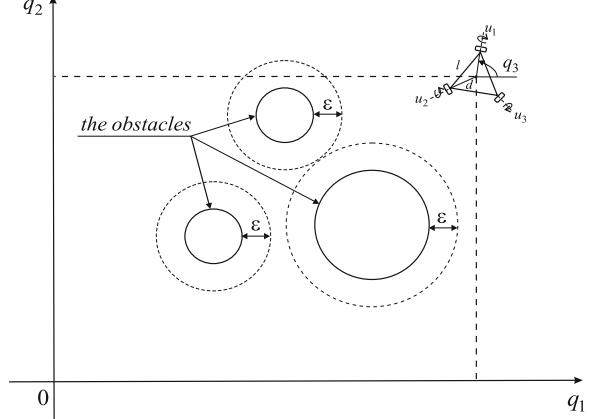


Fig. 1. A kinematic scheme of the vehicle and the task to be accomplished.

2. Problem formulation

An omni-directional vehicle belongs to a class of the common mobile-robot design whose kinematic scheme is shown in Fig. 1. The three wheels, mounted at the vertices of an equilateral triangle with wheel axes joining the center of triangle with each wheel center, not only rotate forwards and backwards, when driven by the motors in a direction perpendicular to the wheel axes, but may also slip sideways freely. Thus, the vehicle may move in any direction instantaneously. Let us consider the omni-directional vehicle described by the vector of generalized coordinates (configuration) $q = (q_1, q_2, q_3)^T \in \mathbb{R}^3$ (see Fig. 1), where (q_1, q_2) stands for the location of the center of the vehicle; q_3 is the angle made between the line joining the robot center with the first wheel center and the horizontal line; constructive parameter $d > 0$ denotes the distance from the center of the robot to each of the wheel center and l means the side length of the platform. Differential equations describing the motion of the vehicle are given in the following form [22]

$$\dot{q} = J(q)u \quad (1)$$

where $u = (u_1, u_2, u_3)^T \in \mathbb{R}^3$ stands for the vector of controls (the wheel angular velocities); $J(q) = \begin{bmatrix} -s_1 & c_1 & d \\ -s_2 & c_2 & d \\ -s_3 & c_3 & d \end{bmatrix}^{-1}$ is the Jacobian matrix that transforms velocities of the wheels into configuration-space velocities of the vehicle; $c_i = \cos(q_3 + (i-1)\pi/3)$; $s_i = \sin(q_3 + (i-1)\pi/3)$; $i = 1 : 3$. Without loss of generality, the following constant limits on controls are set

$$|u_i| \leq u_{\max}, \quad i = 1 : 3, \quad (2)$$

where u_{\max} is the upper bound on the i th control u_i .

A simple generalization of omni-directional vehicle described by Eq. (1) is a four-wheel two axes platform using omnivheels. In such a case, the vehicle becomes redundant (the number of controls which is equal to 4, is greater than the number of state variables which is equal to 3). As is easy to see, equations describing the motion of the four-wheel omni-directional vehicle take the following form

$$\dot{q} = K(q)v \quad (3)$$

where

$$K(q) = (J^T(q)J(q))^{-1}J(q);$$

$$J(q) = \begin{bmatrix} -s'_1 & c'_1 & d \\ -s'_2 & c'_2 & d \\ -s'_3 & c'_3 & d \\ -s'_4 & c'_4 & d \end{bmatrix};$$

$$c'_i = \cos(q_3 + (i-1)\pi/2);$$

$$s'_i = \sin(q_3 + (i-1)\pi/2).$$

q and d are the same as in Eq. (1); $v = (u_1, u_2, u_3, u_4)^T \in \mathbb{R}^4$; $|u_i| \leq u_{\max}$; $i = 1 : 4$. As numerical simulations carried out in Section 4 will show, the four-wheel platform described by Eq. (3) has a preferred direction of movement eliminating the undesirable control oscillations (a singular control) as compared to the three-wheel omni-directional vehicle.

In order to simplify further considerations, sizes of the wheels are also assumed to be negligibly small as compared to the vehicle sides. A task accomplished by the omni-directional vehicle consists in finding controls u satisfying (2) which drive (asymptotically) the vehicle to a desired final configuration $q_f \in \mathbb{R}^3$ with zero velocity at this configuration. Defining the error of regulation as $e = (e_1, e_2, e_3)^T = q - q_f$, the vehicle task (state equality constraints) may formally be expressed by the following equations

$$\lim_{t \rightarrow \infty} e(t) = \lim_{t \rightarrow \infty} (q(t) - q_f) = 0 \quad \lim_{t \rightarrow \infty} \dot{q}(t) = 0. \quad (4)$$

During the vehicle movement, collision avoidance (state inequality) constraints resulting from the existence of obstacles in the task space, are induced. The general form of these constraints may be written in the following manner

$$\{c^j(q) > 0\} \quad j = 1 : N \quad (5)$$

where c^j denotes either a distance function [17] between an obstacle and a vehicle side or an analytic description of the obstacle [16], and N stands for the total number of collision avoidance constraints. Due to negligibly small wheel sizes as compared to vehicle sides, the potential collisions between wheels and the obstacles are not taken for simplicity into account. We postulate further on that both initial vehicle configuration $q_0 = q(0)$ and final configuration q_f together with their small neighbourhoods do not cause collisions, i.e. $c^j(q_0) > 0, c^j(q_f) > 0$. Moreover, functions c^j from (5) are assumed to belong to a class of smooth mappings with respect to any q . In order to involve state inequality constraints (5) in the vehicle motion, suitable exterior penalty functions are introduced. The idea of exterior penalty function approach is to construct a modified criterion by adding a function called an exterior penalty function to prior cost function and then to optimize such unconstrained criterion function. Physically, finite positive values of exterior penalty function represent a penalty imposed by violating constraints (5). Moreover, by fulfilment of state inequality constraints (5), exterior penalty function equals zero. To be more precise, we introduce the following exterior penalty function to satisfy collision avoidance constraints (5)

$$U(q) = c \sum_{j=1}^{N'} E(c^j) \quad (6)$$

where $E(c^j) = (c^j - \epsilon)^2$ for $c^j \leq \epsilon$ and $E(c^j) = 0$ otherwise; ϵ stands for a given threshold value which activates the j -th inequality constraint after exceeding by c^j this value; $N \geq N'$ is the number of only active constraints (5) (i.e. such that $c^j - \epsilon \leq 0$); c denotes positive, constant coefficient (strength of penalty). Let us note that configuration set $\{q : 0 < c^j(q) \leq \epsilon\}$ determines a safety zone around the j th active constraint, $j = 1 : N'$. The size of safety zone equals ϵ and configurations from this set are obviously collision-free. If for some $1 \leq j \leq N'$ and q we have $c^j(q) = \frac{\epsilon}{10}$ then (by definition) the vehicle is said to be $\frac{\epsilon}{10}$ algebraically distant from the j th constraint. Based on (6), the contribution to the penalty of such configuration equals $c \cdot E(\frac{\epsilon}{10})$. Moreover, a vehicle configuration q is said to be safe if its contribution to penalty terms arising from all the active constraints (5) is not greater than $c \cdot E(\frac{\epsilon}{10})$ (it is obviously collision-free). In other words, a vehicle configuration q is said to be safe if the robot of the corresponding configuration q is algebraically distant from an obstacle not closer than $\frac{\epsilon}{10}$. Parameters ϵ and c will be precisely specified in Section 3.2.

Expressions (1)–(6) formulate the omni-directional vehicle task as a control problem. The fact that there exist control and state inequality constraints makes the solution of this problem difficult. The next section will present an approach to the solution (provided that it exists) of the control problem (1)–(6) making use of the Lyapunov stability theory.

3. Generating the controls for omni-directional vehicle

3.1. Constraint-free vehicle motion

In order to determine omni-directional vehicle controls, state inequality constraints (5) are not considered in this section. They will be taken into account in a control scheme proposed in Section 3.2. Based on (1), (2) and (4), a simple control law, solving the vehicle task, is proposed as follows

$$u = -u_{\max} f(J^T(q)\lambda e) \quad (7)$$

where $f(\cdot)$ is a strictly increasing, analytic and non-linear (saturating) function which fulfils the following co-ordinatewise inequalities $f(x) \cdot x \geq 0$ and $|f(x)| \leq 1$ for arbitrary $x \in \mathbb{R}$ ($f(x)x = 0$ iff $x = 0$); $\lambda = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$; λ_i positive gain coefficient, $i = 1 : 3$.

In the case of the four-wheel omni-directional vehicle, the corresponding control law takes the following similar form

$$v = -u_{\max} f(K^T(q)\lambda e). \quad (8)$$

Due to analogous form of controllers (7) and (8), our further analysis will focus only on control law (7). Hence, the closed-loop vehicle error dynamics may be expressed by the following equations

$$\dot{e} = -Ju_{\max} f(J^T\lambda e). \quad (9)$$

Applying the Lyapunov stability theory, we derive the following result.

Theorem 1. *The control scheme (7) generates the controllably admissible solution based on (9), which (asymptotically) converges to the origin $e = (0, 0, 0)^T \in \mathbb{R}^3$.*

Proof. Consider a Lyapunov function candidate

$$V = \frac{1}{2} \langle \lambda e, e \rangle.$$

The time derivative of V is given by

$$\dot{V} = \langle \lambda e, Ju \rangle$$

or equivalently

$$\dot{V} = \langle J^T \lambda e, u \rangle.$$

Substituting u from the above dependence for the right-hand side of Eq. (7), we obtain, that

$$\dot{V} = -u_{\max} \langle J^T \lambda e, f(J^T \lambda e) \rangle.$$

As can be seen, \dot{V} is negative for all $e \neq 0$ ($\text{sign}(J^T \lambda e) = \text{sign}(f(J^T \lambda e))$ and matrix J is non-singular) and is zero if and only if $e = 0$. This implies, using the La Salle–Yoshizawa invariant theorem [23], that e tends asymptotically to zero, i.e. $e(t) \rightarrow 0$ as $t \rightarrow \infty$. On account of (9) and definition of $e, \dot{q}(t) \rightarrow 0$ as $t \rightarrow \infty$. Consequently, state equality constraints (4) are fulfilled. \square

An important remark may be derived from the proof performed. Namely, for sufficiently large matrix gain coefficients λ , (near) time-optimal trajectories, widely investigated in work [1], may be

obtained as numerical simulations carried out in the next section show. Moreover, both vehicle controls and trajectories are analytic mappings. Let us note that time derivative of (7) is also a bounded mapping. Hence, control law (7) provides us both bounded and analytic mappings which may serve as reference trajectories for a trajectory tracking control problem with vehicle dynamics.

Furthermore, using similar derivation technique, as above, we can show, in addition, asymptotic stability of angular accelerations controller for omni-directional vehicle by accomplishment of robot task (4). For this purpose, let us differentiate (1) with respect to time. Consequently, we obtain error dynamics equations with drift of the general form

$$\ddot{e} = J(q)a + g(q, \dot{e})$$

where $a = \dot{u}$ denotes the vector of wheel angular accelerations; $g(\cdot, \cdot)$ stands for a mapping obtained based on Eq. (1) which is analytic and bounded for any q and bounded \dot{e} . Considering a Lyapunov function candidate

$$V' = \int_0^e \langle f(\lambda_1 e), de \rangle + \frac{1}{2} \langle \lambda_2 \dot{e}, \dot{e} \rangle$$

where λ_1, λ_2 denote diagonal matrix positive gain coefficients, it is easy to derive the control law of the form

$$a = -J^{-1}(\lambda_2^{-1}f(\lambda_1 e) + f(\lambda_2 \dot{e}) + g)$$

for which the origin is asymptotically stable (the time derivative of V' is given after performing simple calculations as follows $\dot{V}' = -\langle J^T \lambda_2 \dot{e}, f(J^T \lambda_2 \dot{e}) \rangle$). As $V'(t) \leq V'(0)$ for $t \geq 0$, we conclude that $\|\dot{e}\|$ is bounded. Consequently, angular accelerations a are bounded, too (J^{-1} is also bounded). In order to both discuss and compare the results proposed herein with those presented in recently published work [1], in what follows, control law (7) obtained at the angular velocity level will be studied. It is sometimes very desirable from the practical point of view to start the motion with vehicle velocity equal to zero, i.e. $\dot{q}(0) = 0$. For this purpose, control vector (7) is modified as follows

$$\begin{aligned} u &= -v u_{\max} f(J^T(q) \lambda e) \\ \dot{v} &= -c_1 \omega^\alpha \end{aligned} \quad (10)$$

where $v(0) = 0$; $\omega = v - 1$; c_1 is a positive constant gain; α denotes a positive coefficient which takes for our purpose the following form $\alpha = \frac{n'}{n''}$; $n' < n''$; n', n'' are odd natural numbers, ensuring the stability of controller (10). The lower equation of (10) leads to a terminal attractor [24] whose application for mobile robots was extensively studied in our works [25,26]. Let us note that v stably attains 1 in a finite time. The stability of system (1) and (10) may easily be derived for Lyapunov function candidate equal to $V + \frac{1}{2}\omega^2$.

3.2. Collision-free control

The solutions to the problem of omni-directional vehicle motion in the presence of obstacles can be generally divided into the two classes: global and local methods. Global methods of motion planning which may be divided into discretized and continuous. The discretized technique is a sequential search of a graph whose edges are generated based on a discretized control space. Graph-search methods proposed in [27–29] generate the globally optimal motion. The drawback of using graph-search techniques for trajectory generation is the resolution lost due to discretization of state and/or control space. Global continuous methods may be categorized into two classes. The first class uses an optimal control theory to generate robot trajectories. The Pontryagin Maximum Principle has been used in [6] to determine the

time-optimal trajectory in the unobstructed work space. The resulting controls are discontinuous and bang-bang. Using the calculus of variations and parameterization of controls, works [4,3] convert the continuous optimal control formulation into an equivalent nonlinear programming problem. The second class of non-holonomic motion planning techniques offered in works [30,31] is based on the use of Newton's method. Its adaptation to collision avoidance contains [13]. However, it requires a time-consuming iterative computational procedure. Moreover, only local optimization of a performance index is carried out when searching for the robot trajectory. A suitable transformation of nonholonomic constraints and trajectory parameterization has been proposed in [32] to avoid obstacles. Nevertheless, this method does not take into account control constraints. In the context of mobile manipulators with nonholonomic platform, motion planning algorithms at the control feedback level have been proposed in works [25,26]. Summarizing, global methods are (by their nature) computationally expensive. Furthermore, they are, in fact, inapplicable when there are unmodelled or moving obstacles in the task space. Because of these limitations, global collision avoidance is implemented in an off-line mode. Since our aim is to avoid collisions of the omni-directional vehicle with obstacles (and not to globally plan collision-free trajectory) during its movement to desired position and orientation q_f , we are interested in utilizing local methods, which are less computationally involved than global techniques. Moreover, local methods can be applied for real-time sensor based trajectory modification in a neighbourhood of an obstacle. These attributes make them implementable in an on-line collision avoidance. Among local methods, a method based on a potential field approach and on a Lie algebra of system evaluated at a given point, are the most representative. They have been developed in works [16,33]. However, the computation of the GCBHD formula [34] in work [33] seems to be time consuming and control signals in an obstacle neighbourhood may take arbitrarily large values in [16] which is undesirable property.

In order to propose a technique of local collision avoidance eliminating aforementioned shortcomings, (non-zero) vector λe is assumed further on to be linearly independent on $\frac{\partial U(q)}{\partial q}$. This assumption is not restrictive from the practical point of view and will be discussed further on in this section.

Our aim is to control the omni-directional vehicle such that it does not collide with obstacles, angular velocity limits (2) are not violated and the vehicle simultaneously reaches desired configuration q_f with zero velocity at this configuration. Based on (1)–(4) and (6), we propose a vehicle controller of the form

$$u = -u_{\max} f\left(J^T(q) \left(\lambda e + \frac{\partial U(q)}{\partial q}\right)\right). \quad (11)$$

Inserting regulation error e into vehicle motion equations and putting the right-hand side of (11) into (1) yields the closed-loop error dynamics

$$\dot{e} = -Ju_{\max} f\left(J^T\left(\lambda e + \frac{\partial U}{\partial q}\right)\right). \quad (12)$$

Applying the Lyapunov stability theory, we derive the following result.

Theorem 2. *If there exists a solution to the problem (2)–(5) and (non-zero) vector λe is linearly independent on $\frac{\partial U}{\partial q}$ along the vehicle trajectory, then control scheme (11) generates a collision-free controllably admissible trajectory whose equilibrium point $e = (0, 0, 0)^T \in \mathbb{R}^3$ is asymptotically stable.*

Proof. Consider a Lyapunov function candidate

$$V_c = V + U(q) = \frac{1}{2} \langle \lambda e, e \rangle + U(q). \quad (13)$$

The time derivative of V_c is given by

$$\dot{V}_c = \langle \lambda e, \dot{e} \rangle + \left\langle \frac{\partial U}{\partial q}, \dot{e} \right\rangle.$$

Substituting \dot{e} from the above dependence for the right-hand side of closed-loop error dynamics (12), we obtain

$$\dot{V}_c = -u_{\max} \left\langle J^T \left(\lambda e + \frac{\partial U}{\partial q} \right), f \left(J^T \left(\lambda e + \frac{\partial U}{\partial q} \right) \right) \right\rangle.$$

As is easy to see, \dot{V}_c is negative for $\lambda e + \frac{\partial U}{\partial q} \neq 0$ and is zero iff $\lambda e + \frac{\partial U}{\partial q} = 0$ which implies using the La Salle–Yoshizawa invariant theorem [23], that $\lambda e + \frac{\partial U}{\partial q}$ tends asymptotically to zero, i.e. $\lambda e + \frac{\partial U}{\partial q} \rightarrow 0$ as $t \rightarrow \infty$. As (non-zero) vector λe and $\frac{\partial U}{\partial q}$ are (by assumption) linearly independent, we finally obtain that $e \rightarrow 0$ and $\frac{\partial U}{\partial q} \rightarrow 0$ as $t \rightarrow \infty$. On account of the fact that q_0 is (by assumption) a collision-free configuration and (13), we have $V_c(0) = \frac{1}{2} \langle \lambda e(0), e(0) \rangle$. Since \dot{V}_c is not positive, function V_c fulfills the inequality ($\forall t \geq 0$) $V_c(0) \geq V_c(t)$. Furthermore, the choice of parameter c as follows

$$c > \frac{V_c(0)}{E(\frac{\epsilon}{10})} = \frac{\frac{1}{2} \langle \lambda e(0), e(0) \rangle}{E(\frac{\epsilon}{10})} \quad (14)$$

ensures us, that ($\forall t \geq 0$) $U > V_c(0) \geq V_c(t)$ for the vehicle distant less than $\frac{\epsilon}{10}$ from obstacles. On the other hand due to inequality ($\forall t \geq 0$) $V_c(0) \geq V_c(t)$, we have $U \leq V_c(0)$. Consequently for c satisfying (14), controller (11) generates collision-free trajectory of the omni-directional vehicle. \square

Several important remarks may be derived from the proof carried out. First, note that controller (11) guarantees both asymptotic stability of the closed-loop error dynamics (12) and fulfilment of control limits (2). Second, the choice of penalty parameter c according to inequality (14) guarantees that the vehicle is at least $\epsilon/10$ algebraically distant from obstacles during its movement along trajectory generated by controller (11). It follows from the analysis of controller (11), that the robot will collision-freely attain q_f provided that (non-zero) vector λe is linearly independent on $\frac{\partial U(q)}{\partial q}$. However, in the regular case (the

Jacobian matrix $\frac{\partial(\lambda e + \frac{\partial U}{\partial q})}{\partial q}$ is full rank at $\lambda e + \frac{\partial U}{\partial q} = 0$) at most a finite number of configurations may only lead to linear dependence of λe on $\frac{\partial U}{\partial q}$ and therefore it is unlikely to meet such configurations during the vehicle movement. Let us note that if this is the case and for strictly convex obstacles, these configurations are unstable equilibria (mobile platform is a convex set). Hence, a small disturbance in any direction will result in further collision-free platform movement to configuration q_f . Nevertheless, if (non-zero) vector λe is linearly dependent on $\frac{\partial U}{\partial q}$ and the corresponding configuration is a stable equilibrium, then the robot stops before desired configuration q_f is reached which means that controller (11) fails. Thus, global methods must be utilized to achieve q_f (see e.g. [35] for a real time version of A^* used to a redundant manipulator). Let us also note, that it is difficult (or not possible, in general) to obtain a result for global stability of controller (11) in the presence of obstacles. Involving a Filippov solution [36] results in discontinuous right-hand side of motion equations, i.e. discontinuity of vehicle velocity, which induces the undesirable effect of chattering. Furthermore, it is also very hard to prove the existence of a solution for controller (11) due to non-convex (in general) geometry of the obstacles. In such a case, controller (11) may also fail and global algorithms proposed e.g. in work [35] have to be applied to attain configuration q_f .

On the other hand, the main advantage of the (local) solution proposed here is the smooth property of vehicle velocity. Moreover, velocity constraints (2) are not violated during the vehicle motion. Furthermore, the robot explores the environment in such a way that Lyapunov function (13) asymptotically decreases to zero. It is also worth emphasizing that the knowledge of obstacles shapes is not required by generating the vehicle controls. However, for simplicity of our considerations, the robot equipped with a vision system (e.g. video camera) is assumed to provide controller (11) with an analytic description of the obstacles. Nevertheless, control scheme (11) may be applicable to unknown environments. If this is the case, the vehicle may be equipped with e.g. laser range finder or sonars. These robot proximity sensors may continually provide locations of the obstacle's points nearest to the vehicle. Consequently, knowing location of the two nearest points between obstacle and the mobile platform, and the robot configuration, the partial derivative of U with respect to q , required by controller (11) may be computed provided that c^j are assumed to be the Euclidean distances. Moreover, an important remark follows from dependencies (7) and (11). Namely, the vehicle task (2)–(5) has been solved at the feedback control level. Consequently, it is possible to control the vehicle in a real time provided that a position controller is available which closely tracks trajectories obtained from control schemes (7) and (11). Let us note, that control law (11) increases somewhat computational complexity of the collision avoidance task. Nevertheless, a very few active inequality constraints (5) are expected to occur in practical applications which significantly reduces the computational burden. The user dependent parameter ϵ is responsible for the size of safety zones which enclose the obstacles. Consequently, enclosing the obstacles in safety zones becomes conservative and may not yield a solution in the case of a cluttered task space. In order to overcome this inconvenience, gain coefficients λ and c related with ϵ by inequality (14) should be chosen as small as it is possible which corresponds to a slower platform motion (parameter λ) and a deeper penetration into safety zones (parameter c). Consequently, the omni-directional vehicle may move very close to obstacles yielding thus a solution, as the simulation results carried out in the next section show. It also follows from inequality (14), that increasing parameter ϵ results in decreasing an admissible lower bound on c . Due to usually relative large value $\|e(0)\|^2$, estimation of c from (14) (for a prescribed ϵ and λ) may be conservative as the numerical simulations carried out in Section 4 show. Finally, due to real-time nature of vehicle controller (11), we shall estimate the number of arithmetic operations required to implement the control law presented in this section. As is seen from formula (11), the partial derivative of penalty function U with respect to q is the most time-consuming term to be computed. Its estimation requires discretization, say M sample points, of mobile platform. Operations required for computation of \sin , \cos , J^T and f in (11) are not taken into account. Moreover, estimations are carried out at any time instant of the robot task accomplishment. It is difficult, in general, to estimate the number of operations required for computing term $\frac{\partial U}{\partial q}$ in controller (11). It depends on the number of points on the mobile platform that activate inequality constraints (5). If we assume for simplicity that in the worst case all the sample points on the platform activate N' collision-free constraints (5) at any time instant of the platform movement, then computational complexity for $\frac{\partial U}{\partial q}$ is of the order of $O(MN')$. Term $J^T \lambda e$ in (11) requires $O(1)$ operations. Consequently, computational complexity of the whole controller (11) is of the order of $O(MN')$.

In this context, it is interesting to compare our control scheme with a behavior-based control strategy proposed e.g. in works [37,38]. A finite set of robot turn command options (path arcs) is available in the behavior-based control. Hence, each path arc must be tested for both collision avoidance and goal seeking. The

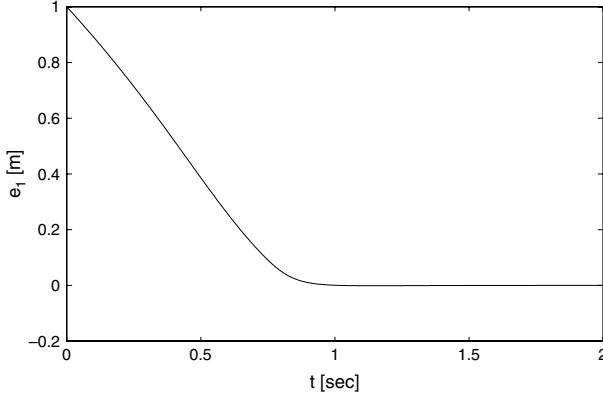


Fig. 2. Vehicle position error e_1 – constraint-free and small displacement.

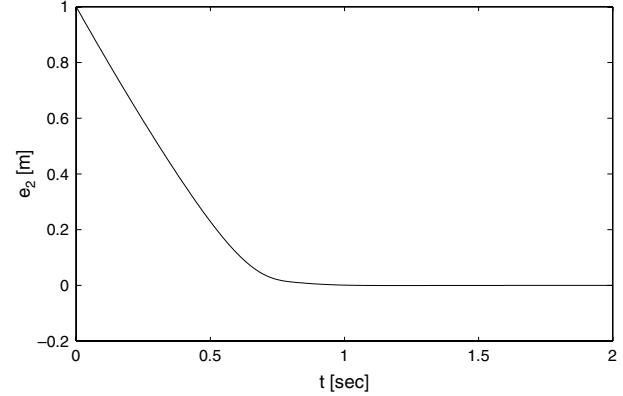


Fig. 3. Vehicle position error e_2 – constraint-free and small displacement.

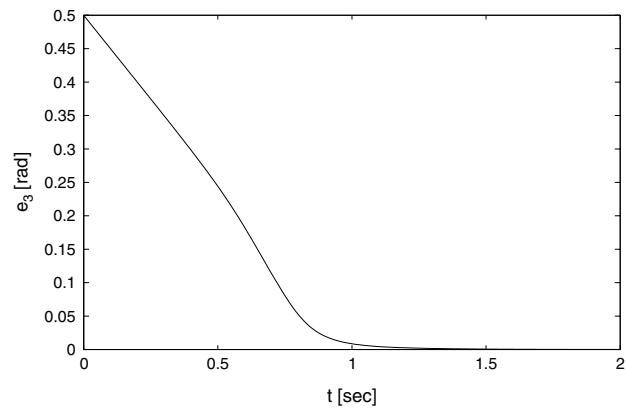


Fig. 4. Vehicle position error e_3 – constraint-free and small displacement.

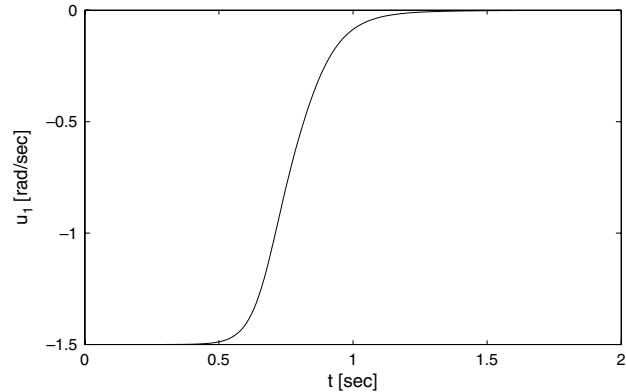


Fig. 5. Vehicle control u_1 – constraint-free and small displacement.

command choice to the controller depends on the vote weights (specified by the user) of each behavior. Consequently, a finite set of arcs and arbitrary choice of behavior weights may not be sufficient to yield a solution (although it may exist) to the problem of both avoiding the obstacles and attaining the goal. On the other hand, controller (11) generates only one proper robot action (an arc) from a set of infinitely many actions which control law (11) is able to generate. Provided that the assumptions of Theorem 2 hold true, the robot is guaranteed to attain the desired configuration and to never be in collision with obstacles when moving in the work space. Furthermore, behavior-based architecture steers the robot to a small neighbourhood of a desired location (only stability of the robot is attained) whereas control scheme (11) provides asymptotically stable robot motion to the goal configuration. Moreover, the important property of controller (11) is, that it does not violate control constraints even in a very close neighbourhood of obstacles. However, it may not be the case of behavior-based architectures. Finally, it is important to stress that our control law belongs, in fact, to a class of local methods (quickly generating the robot trajectory) whereas behavior-based architectures are usually global (time-consuming) techniques provided that they are equipped with global path planner.

4. Computer example

This section demonstrates on the five chosen vehicle tasks, the performance of controllers given by Eqs. (7), (8) and (11). For this purpose, an omni-directional vehicle, schematically shown in Fig. 1, is considered. In all numerical simulations, SI units are used. The side length l of the vehicle taken for numerical simulations equals $l = \sqrt{3}$ [m] and $d = 1$ [m]. The upper bounds on angular wheel velocities equal

$$u_{\max} = 1.5 \text{ [rad/s].}$$

The saturating (sigmoidal) function f chosen in such a way as not to violate angular wheel velocity bounds (2) and to produce smooth platform velocities, takes the form

$$f(\cdot) = 2 \cdot \left(\frac{1}{1 + e^{(\cdot)}} - \frac{1}{2} \right).$$

The task of the vehicle is to attain the final location $q_f = (0, 0, 0)^T$. In order to show the robustness of controllers (7), (8) and (11) against boundary conditions and different number of obstacles, five computer simulations have been carried out.

In the first simulation, the task of the vehicle in a constraint-free task space is to move to the origin from the initial configuration equal to $q_0 = (1, 1, 0.5)^T$ – a small displacement. Matrix gain coefficients equal $\lambda = \text{diag}(33, 12, 40)$. The results of computer

simulations are presented in Figs. 2–8 which indicate that accurate positioning of the vehicle is achieved (see Figs. 2–4). As is seen from Figs. 5–7 controls u satisfy constraints (2) and are approximately bang-bang. These types of controls were also theoretically investigated in recent work [1] where the authors have called it a circular arc type of control. Fig. 8 presents the vehicle movement. According to the nomenclature proposed in [1], controls from Figs. 5–7 generate (near) time-optimal trajectory called a rollcw trajectory. In general, such an optimal trajectory is a sequence of circular arcs and spins in place.

In the second simulation, the initial vehicle configuration in a constraint-free task space equals $q_0 = (15, 10, 3)^T$ – a large displacement. Matrix gain coefficients chosen in such a case are

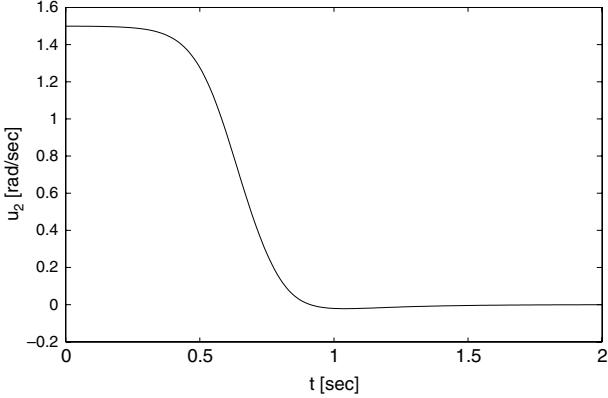
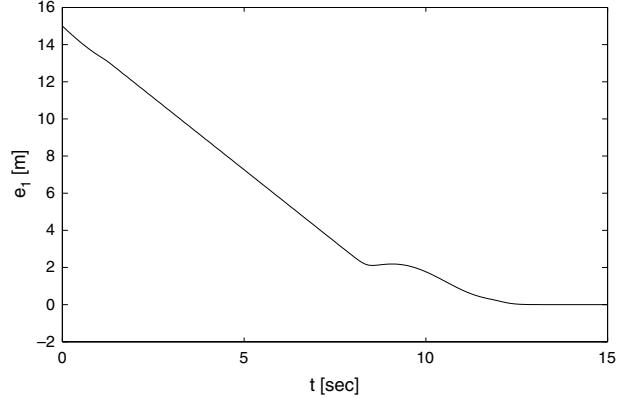
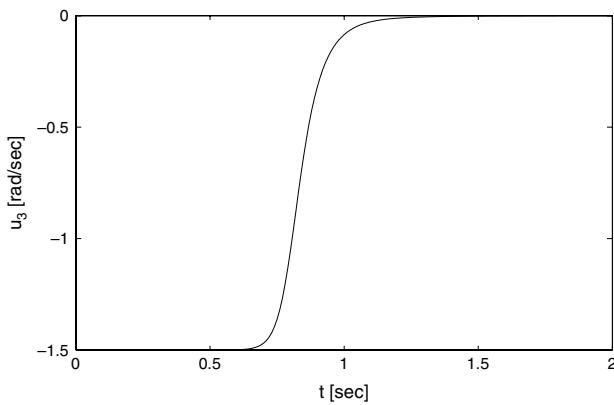
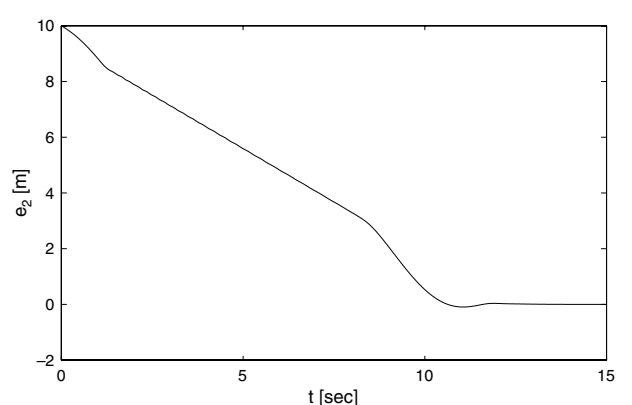
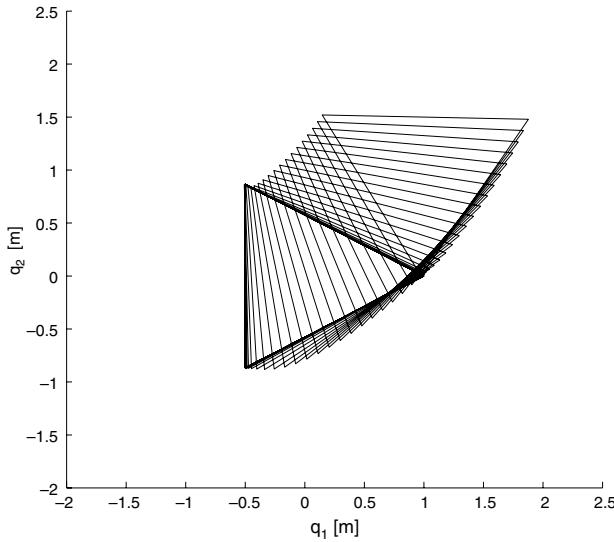
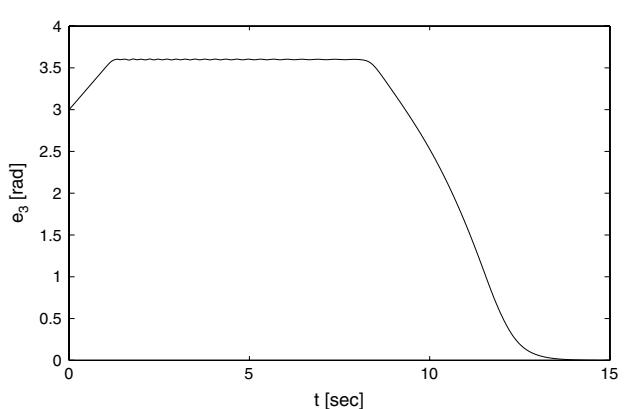
Fig. 6. Vehicle control u_2 – constraint-free and small displacement.Fig. 9. Vehicle position error e_1 – constraint-free and large displacement.Fig. 7. Vehicle control u_3 – constraint-free and small displacement.Fig. 10. Vehicle position error e_2 – constraint-free and large displacement.

Fig. 8. Vehicle movement – constraint-free and small displacement.

equal to $\lambda = \text{diag}(8, 10, 10)$. The results of computer simulations are presented in Figs. 9–15 which indicate that also accurate positioning of the vehicle is achieved (see Figs. 9–11). As is seen from Figs. 12–14, all controls u are (approximately) bang-bang in the time interval $[0, 1.1]$. Then, control u_1 starts to oscillate (see Fig. 12). Since formulas (7) and (9) provide an analytic solution, control u_1 cannot be equal to zero in an open time subinterval (otherwise such control would be identically equal to zero).

Fig. 11. Vehicle position error e_3 – constraint-free and large displacement.

Nevertheless, theoretical investigations carried out in [1], which are based on the Pontryagin Maximum Principle, provide non-analytic solutions which may be zero in an open time subinterval (a singular control). In this context, control u_1 from Fig. 12 is singular for $t > 1.1$. Moreover, our controls enforce additional (practically desirable) constraints of (asymptotic) zero final velocity. Hence, all the controls (asymptotically) equal zero at configuration q_f (see Figs. 12–14). Consequently, at the end phase of time histories, controls do not usually fulfil the Pontryagin Maximum Principle. The corresponding vehicle trajectory, called in work [1], a tangent (singular) trajectory is depicted in Fig. 15. As is seen from this figure, near time-optimal trajectory consists, in such a case, of circle arcs,

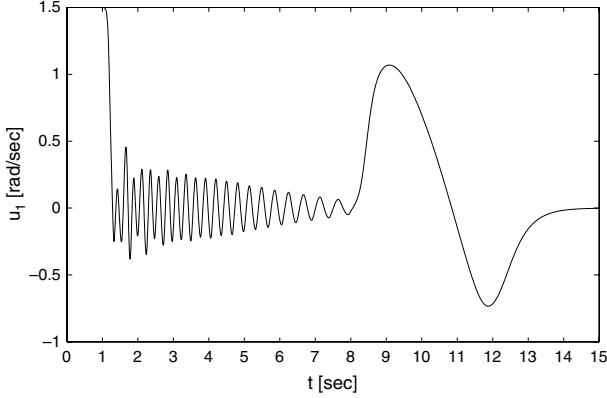


Fig. 12. Vehicle control u_1 – constraint-free and large displacement.

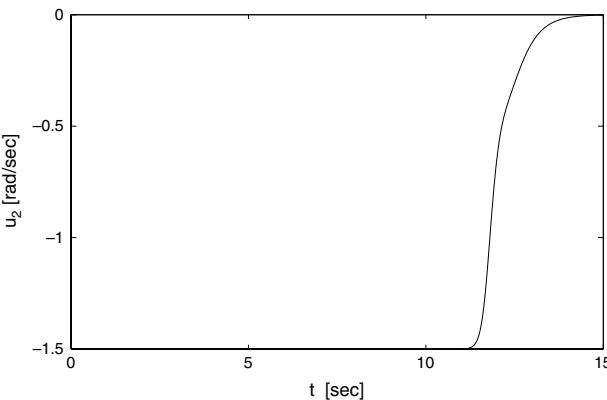


Fig. 13. Vehicle control u_2 – constraint-free and large displacement.

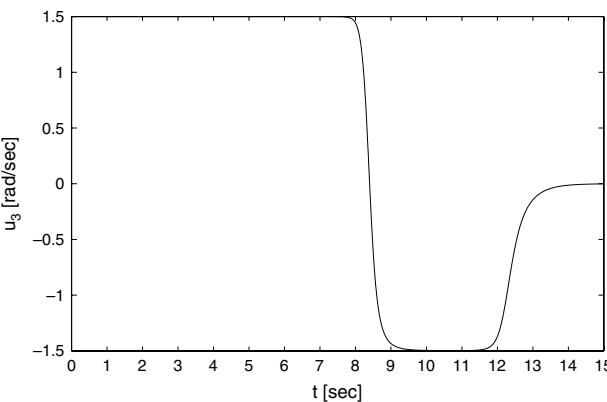


Fig. 14. Vehicle control u_3 – constraint-free and large displacement.

a singular translation and a spin. Due to enforced condition of zero final velocity, this spin does not result in all bang–bang controls.

In the third simulation, the four-wheel omni-directional vehicle was used to solve the same task as in the second experiment. In order to compare the performance of this vehicle with that analyzed in the previous experiment, initial configurations of both vehicles and gain coefficients of controllers (7) and (8) are assumed to be equal. They take numerical values from the second experiment. The results of computer simulations are depicted in Figs. 16–20. As compared to control variable u_1 from Fig. 12, controller (8) eliminates the undesirable oscillations of controls (see Figs. 17–20) which may be explained by a preferred direction of movement of the four-wheel vehicle.

In the fourth simulation, the vehicle considered in this section was used to solve the same task as in the second experiment,

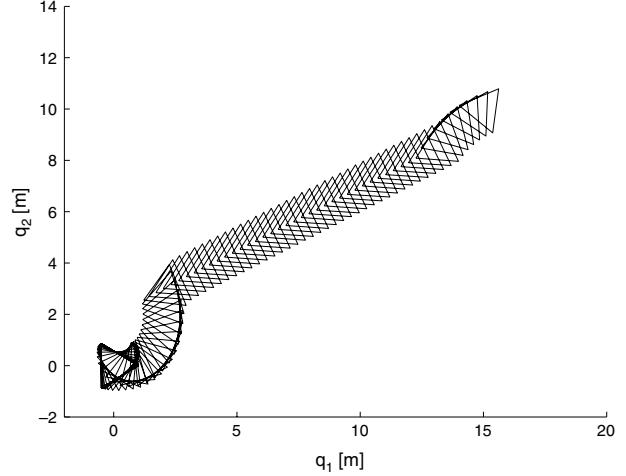


Fig. 15. Vehicle movement – constraint-free and large displacement.

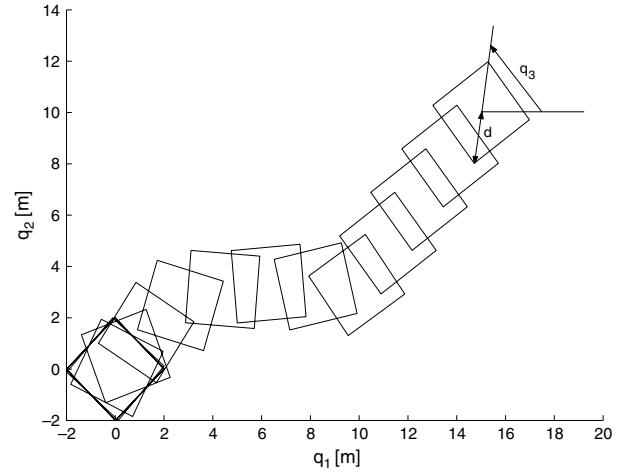


Fig. 16. Vehicle movement – constraint-free and large displacement-four-wheel vehicle.

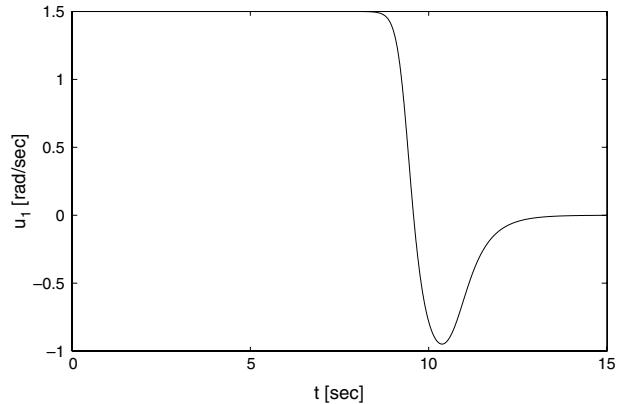


Fig. 17. Vehicle control u_1 – constraint-free and large displacement-four-wheel vehicle.

however by the assumption that there are now three obstacles—circles (provided for simplicity of calculations in analytical form by the robot vision system), schematically presented in Fig. 1, in the cluttered task space. The robot planned the route such that Lyapunov function V_c asymptotically decreased its value during the platform movement. The total number of collision avoidance constraints, which are assumed herein to be distance functions, equals $N = 9$. In order to calculate numerically the values of

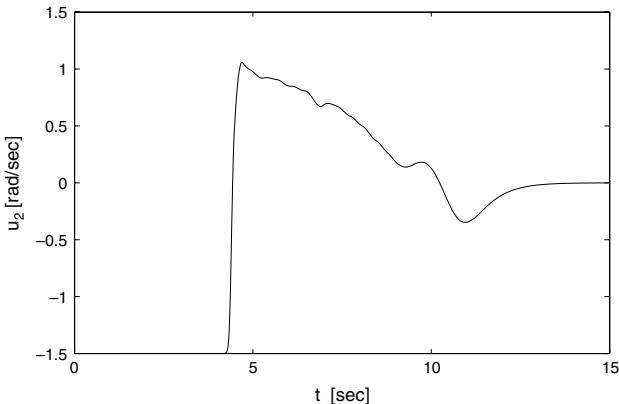


Fig. 18. Vehicle control u_2 – constraint-free and large displacement-four-wheel vehicle.

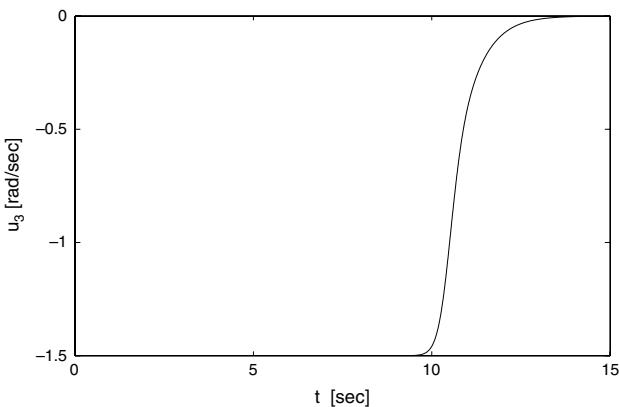


Fig. 19. Vehicle control u_3 – constraint-free and large displacement-four-wheel vehicle.

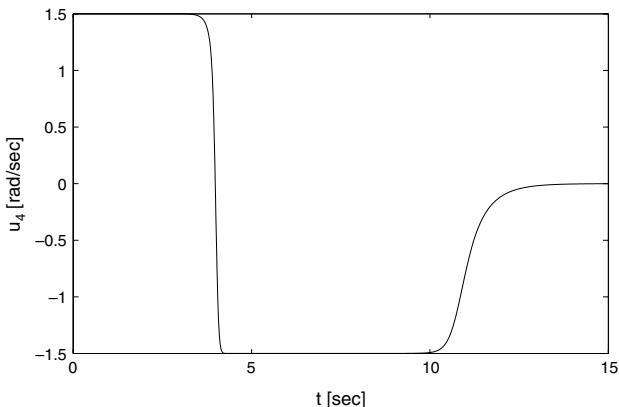


Fig. 20. Vehicle control u_4 – constraint-free and large displacement-four-wheel vehicle.

functions $c^j, j = 1 : 9$, each vehicle side has been discretized into 20 points. The threshold value ϵ taken for computations is equal to $\epsilon = 1.0$. Gain matrix λ is the same as in the second simulation. The coefficients of matrix gain λ should be chosen to fulfil inequality (14). However, their smaller values for collision-free movement as compared to the first experiment, are dictated by the fact of obtaining a mild platform motion in small neighbourhoods of obstacles. Gain parameter c takes the following value $c = 10$. Although, inequality (14) provides coefficient c such that collision avoidance is ensured, the quality of movement (mild, impetuous) is, in fact, user dependent. The chosen value of c (which is smaller than that obtained from inequality (14))

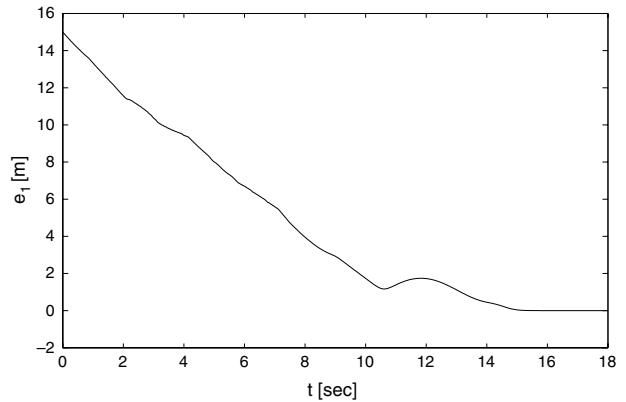


Fig. 21. Vehicle position error e_1 – collision-free displacement in cluttered task space.

results in a deep penetration of safety zones (see Fig. 27) but provides both a solution to the vehicle task and mild movement which is a desirable property. The choice of c (for a relatively cluttered task space presented in Fig. 1) according to formula (14) led us to stopping the vehicle before configuration q_f was attained. Figs. 21–27 present the results of computer simulations which indicate that also accurate positioning of the vehicle, operating among obstacles, is achieved (see Figs. 21–23). As is seen from Figs. 24–26, in the time interval [1.1 7.5], vehicle controls oscillate. It is a consequence of activating collision avoidance constraints (see Fig. 27). As is also seen from Fig. 27, the collision-free vehicle trajectory includes a singular translation. Compared to the second experiment, performance time T in the collision avoidance task has increased approximately by 3 s. This increase is the result of active collision avoidance constraints (see Fig. 27) since the vehicle has to decrease its velocity in neighbourhoods of obstacles. Moreover, Fig. 27 shows, that only at most $N' = 6$, from the total number of $N = 9$, collision avoidance constraints were active (two constraints arising from the obstacle with center $(q_1, q_2) = (12, 6)$, two constraints arising from the obstacle with center $(q_1, q_2) = (6, 5.5)$ and two constraints generated by the existence of the obstacle with center $(q_1, q_2) = (9, 10)$) during the planning session which decreases the computational burden.

The last experiment is to show the structure of (approximately) time-optimal controls and the trajectory of the omni-directional vehicle which should not enter an obstacle safety zone. As opposed to the fourth simulation, we now permit only the vehicle movement along the safety zone boundary. For this purpose, we locate in the task space only one obstacle (circle) with $\epsilon = 1$ such that it touches the vehicle at the initial configuration (see Fig. 31) to enforce near time-optimal trajectory. In order to prevent safety zone penetration, gain coefficient c is chosen sufficiently large (it fulfils inequality (14)) and equals $c = 312$. Thus, the topological sum of the obstacle and its safety zone may now be treated as a new enlarged obstacle. The controller gain matrix λ takes the following values $\lambda = \text{diag}(13, 10, 10)$. The simulation results are presented in Figs. 28–31. As is seen from Figs. 28–30, controls u_1, u_3 are of (approximately) bang–bang type for $t \in [0, 4.1]$ and then for $t > 4.1$ control u_2 becomes singular. The corresponding near time-optimal vehicle trajectory, consisting of circular arcs, a singular translation and a spin, is depicted in Fig. 31. As was to expect, the omni-directional vehicle moves at the beginning of task performance along the obstacle boundary which confirms the hypothesis stated in work [1].

5. Conclusions

In this paper, the positioning vehicle task subject to both control and state constraints, has been discussed. The control

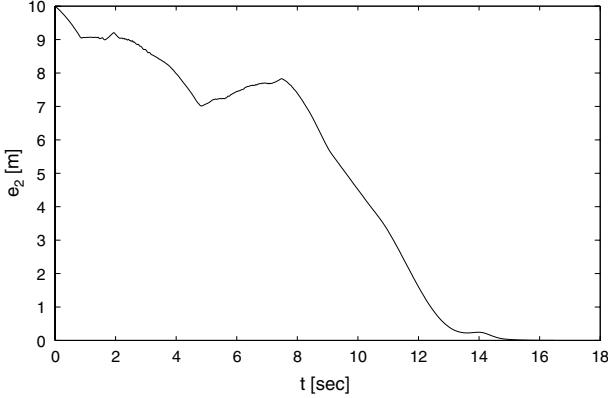


Fig. 22. Vehicle position error e_2 – collision-free displacement in cluttered task space.

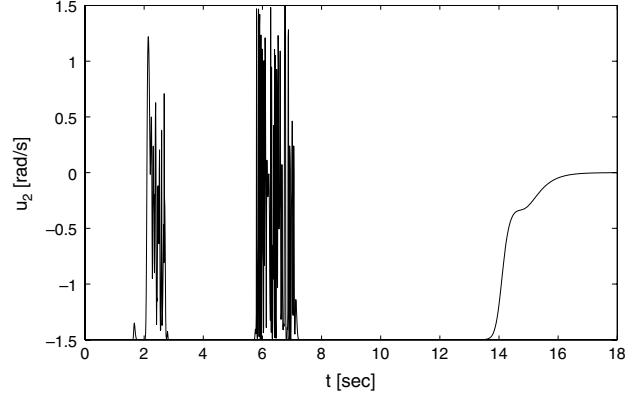


Fig. 25. Vehicle control u_2 – collision-free displacement in cluttered task space.

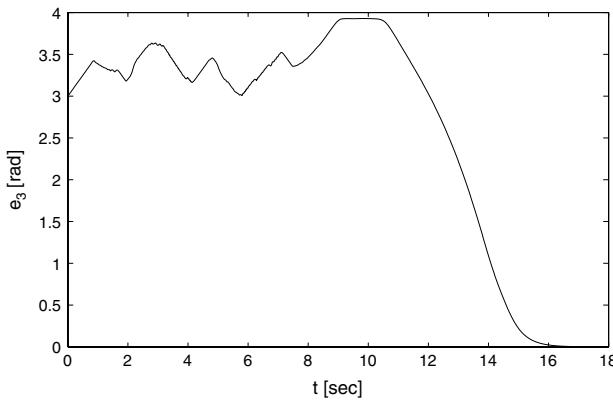


Fig. 23. Vehicle position error e_3 – collision-free displacement in cluttered task space.

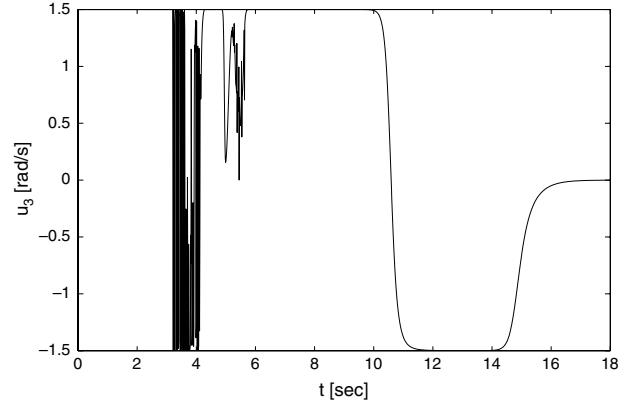


Fig. 26. Vehicle control u_3 – collision-free displacement in cluttered task space.

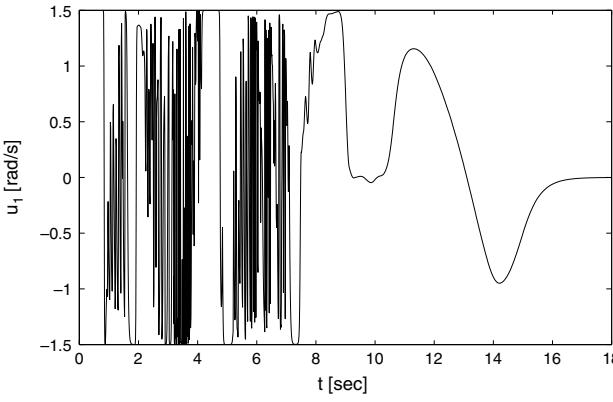


Fig. 24. Vehicle control u_1 – collision-free displacement in cluttered task space.

strategies (7), (8) and (11), taking into account both vehicle velocity limits (2) and collision avoidance constraints (5), are shown to be asymptotically stable (by fulfilment of practically reasonable assumptions). The control scheme proposed provides the user with the capability to vary the level of information needed by controller (11) depending on the form of functions c^i . That is, the approach presented is equally applicable to analytical descriptions of obstacles in the task space or distances (provided by the vehicle sensors) between the omni-directional vehicle and obstacles. Moreover, the control algorithms proposed in this paper generate

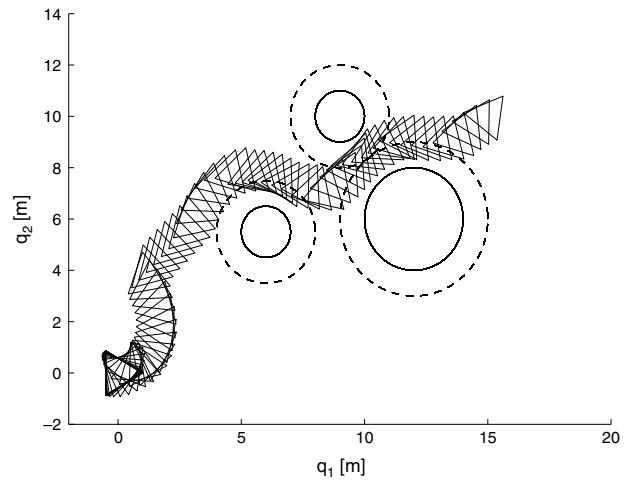


Fig. 27. Collision-free vehicle movement in cluttered task space.

at least smooth velocities (even for collision avoidance tasks) which is a desirable property in an on-line control. Numerical simulations carried out on an exemplary omni-directional vehicle have confirmed both theoretical results obtained in Section 3 and those presented in works [1]. The advantage of using the method proposed is the possibility to implement it in an on-line collision avoidance. Moreover, controller (11) does not require analytical description of obstacles, which are not often known in practice.

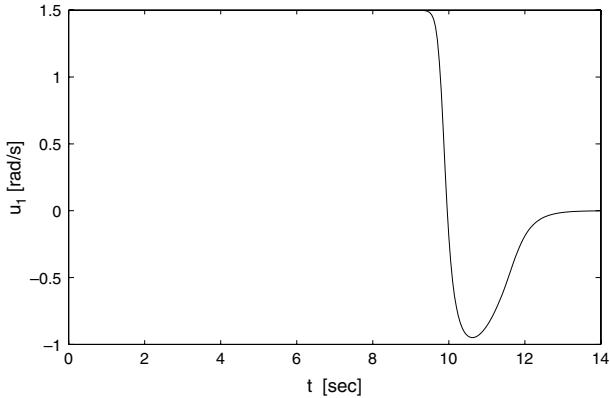


Fig. 28. Vehicle control u_1 – one obstacle in the task space.

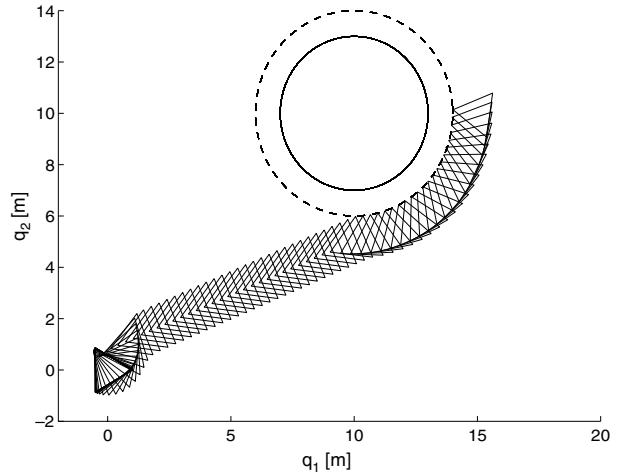


Fig. 31. Collision-free vehicle movement with one obstacle.

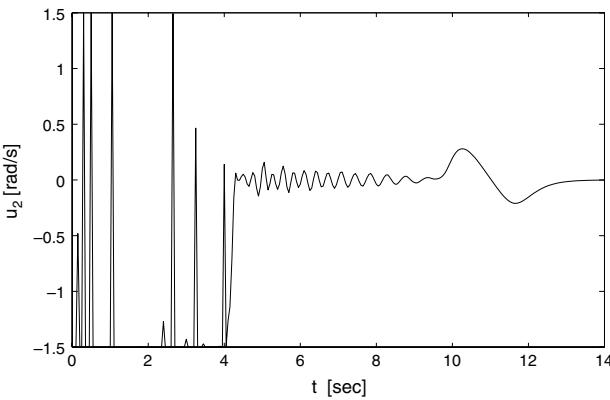


Fig. 29. Vehicle control u_2 – one obstacle in the task space.

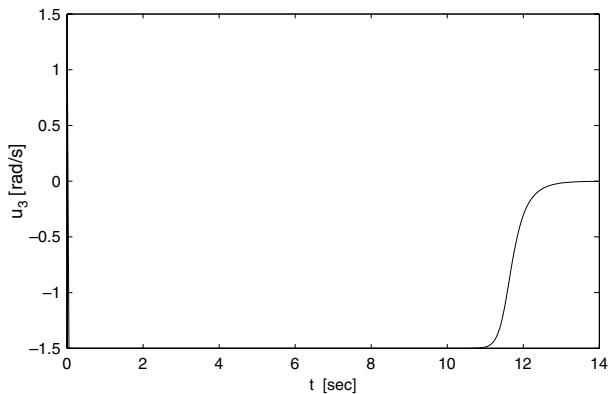


Fig. 30. Vehicle control u_3 – one obstacle in the task space.

Therefore, control scheme (11) may be applicable to unknown environments. Due to a local nature of the controller proposed, there is no guarantee to obtain a solution (even if it exists) in a complex task space including many obstacles. In such a case, computationally involved methods proposed e.g. in [35] are more suitable to generate collision-free motions. The approach proposed may be directly applicable to multiple vehicles operating in task spaces including also (moving) obstacles.

Acknowledgement

This work was supported by the DFG Ga 652/1-1, 2.

References

- [1] D.J. Balkcom, P.A. Kavathekar, M.T. Mason, Time-optimal trajectories for an omni-directional vehicle, *International Journal of Robotics Research* 25 (10) (2006) 985–999.
- [2] T. Kalmar-Nagy, R. D’Andrea, P. Ganguly, Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle, *Robotics and Autonomous Systems* 46 (2004) 47–64.
- [3] T.M. Howard, A. Kelly, Optimal rough terrain trajectory generation for wheeled mobile robots, *International Journal of Robotics Research* 26 (2) (2007) 141–166.
- [4] A. Kelly, B. Nagy, Reactive nonholonomic trajectory generation via parametric optimal control, *International Journal of Robotics Research* 22 (7–8) (2003) 583–601.
- [5] B. Nagy, A. Kelly, Trajectory generation for car-like robots using cubic curvature polynomials, in: Proc. Field and Service Robots, Helsinki, Finland, 2001.
- [6] D.J. Balkcom, M.T. Mason, Time optimal trajectories for bounded velocity differential drive vehicles, *International Journal of Robotics Research* 21 (3) (2002) 199–217.
- [7] M. Renaud, J.-Y. Fourquet, Minimum time motion of a mobile robot with two independent acceleration-driven wheels, in: Proc. IEEE Intern. Conf. Robotics and Automation, 1997.
- [8] D.B. Reister, F.G. Pin, Time-optimal trajectories for mobile robots with two independently driven wheels, *International Journal of Robotics Research* 13 (1) (1994) 38–54.
- [9] J. Reuter, Mobile robot trajectories with continuously differentiable curvature: An optimal control approach, in: Proc. the IEEE/RSI Conf. on Intelligent Robots and Systems, 1998.
- [10] N. Faiz, S. Agrawal, R. Murray, Differentially flat systems with inequality constraints: An approach to real-time feasible trajectory generation, *Journal of Guidance, Control, and Dynamics* 24 (2) (2001) 219–227.
- [11] G. Desaulniers, On shortest paths for a car-like robot maneuvering around obstacles, *Robotics and Autonomous Systems* 17 (1996) 139–148.
- [12] M. Venditelli, J.P. Laumond, C. Nissoix, Obstacle distance for car-like robots, *IEEE Transactions Robotics and Automation*, 15, 4 678–691.
- [13] A.W. Davelbiss, J.T. Wen, A path space approach to nonholonomic motion planning in the presence of obstacles, *IEEE Transactions Robotics Automation* 13 (3) (1998) 443–451.
- [14] H.G. Tanner, K.J. Kyriakopoulos, Nonholonomic motion planning for mobile manipulators, in: Proc. Int. Conf. Robotics and Automation, 2000, pp. 1233–1238.
- [15] A.V. Fiacco, G.P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Wiley, New York, 1968.
- [16] O. Khatib, Real-time obstacle avoidance for manipulators and mobile manipulators, *International Journal of Robotics Research* 5 (1) (1986) 90–98.
- [17] E.G. Gilbert, D.W. Johnson, Distance functions and their application to robot path planning, *IEEE Journal of Robotics and Automation* 1 (1985) 21–30.
- [18] J.-H. Chuang, Potential-based modelling of three-dimensional workspace for obstacle avoidance, *IEEE Transactions Robotics Automation* 14 (5) (1998) 778–785.
- [19] J.-O. Kim, P.K. Khosla, Real-time obstacle avoidance using harmonic potential functions, *IEEE Transactions Robotics Automation* 8 (3) (1992) 338–349.
- [20] E. Rimon, D.E. Koditschek, Exact robot navigation using artificial potential functions, *IEEE Transactions Robotics Automation* 8 (5) (1992) 501–518.
- [21] R.A. Conn, M. Kam, Robot motion planning on N -dimensional star worlds among moving obstacles, *IEEE Transactions Robotics Automation* 14 (2) (1998) 320–325.
- [22] I. Williams, L. Cartwer, B.E. Gallina, G. Rosati, Dynamic model with slip for wheeled omnidirectional robots, *IEEE Trans. Rob. Aut.* 18 (3) (2002) 285–293.

- [23] M. Krstic, I. Kanellakopoulos, P. Kokotovic, Nonlinear and Adaptive Control Design, J. Wiley and Sons, New York, 1995.
- [24] M. Zak, Terminal attractors for addressable memory in neural networks, Physics Letters A 133 (1988) 218–222.
- [25] M. Galicki, Inverse kinematics solution to mobile manipulators, International Journal of Robotics Research 22 (12) (2003) 1041–1064.
- [26] M. Galicki, Control-based solution to inverse kinematics for mobile manipulators using penalty functions, Journal of Intelligent Robotic Systems 42 (2005) 213–238.
- [27] J. Barraquand, J.C. Latombe, On nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles, Algorithmica 10 (1993) 121–155.
- [28] P. Ferbach, J.F. Rit, Planning nonholonomic motions for manipulated objects, in: Proc. IEEE Int. Conf. Robotics Automat., 1996, pp. 2949–2955.
- [29] P. Ferbach, A method of progressive constraints for nonholonomic motion planning, in: Proc. IEEE Int. Conf. Robotics Automat., 1996, pp. 2935–2942.
- [30] F. Lizarralde, J.T. Wen, Feedback stabilization of nonholonomic systems based on path space iteration, in: Proc. MMAR Symposium, 1995, pp. 485–490.
- [31] E. Sontag, Gradient technique for systems with nodrift: A classical idea revisited, in: Proc. IEEE Conf. Decision Control, 1993, pp. 2706–2711.
- [32] E. Papadopoulos, I. Poulakakis, I. Papadimitriou, On path planning and obstacle avoidance for nonholonomic platforms with manipulators: A polynomial approach, International Journal of Robotics Research 21 (4) (2002) 367–383.
- [33] I. Duleba, Locally optimal motion planning of nonholonomic systems, Journal of Robotic Systems 14 (11) (1997) 767–788.
- [34] R.S. Strichartz, The Campbell–Baker–Hausdorff–Dynkin formula and solutions of differential equations, Journal of Functional Analysis 72 (1987) 320–345.
- [35] M. Galicki, A. Morecki, Finding collision-free trajectory for redundant manipulator by local information available, in: Proc. the Ninth CISM-IFToMM Symposium on Theory and Practice of Robots and Manipulators, 1992, pp. 61–71.
- [36] D. Shevitz, B. Paden, Lyapunov stability theory of nonsmooth systems, IEEE Transactions on Automatic Control 30 (9) (1994) 1910–1914.
- [37] J.K. Rosenblatt, DAMN: A distributed architecture for mobile navigation, Journal of Experimental & Theoretical Artificial Intelligence 9 (1997) 339–360.
- [38] M.J. Mataric, Behaviour-based control: Examples from navigation, learning, and group behaviour, Journal of Experimental & Theoretical Artificial Intelligence 9 (1997) 323–336.



Miroslaw Galicki received the M.Sc. degree from the Technical University Wroclaw, Poland, in 1980, and the Ph.D. degree in Electrical Engineering and Computer Sciences from the Technical University of Wroclaw in 1984.

He was with the Institute of Technical Cybernetics at the Technical University Wroclaw from 1981 to 1985. Since 1985 he has been at the Technical University, Zielona Gora, Poland and since 1992 he has also been with the Institute of Medical Statistics, Computer Sciences, and Documentation, Friedrich Schiller University, Jena, Germany. His research interests include structural and stochastic optimization, both holonomic and nonholonomic systems, and optimal control problems.