DTD 5
ARTICLE IN PRESS

Available online at www.sciencedirect.com
SCIENCE DIRECT®

Robotics and Autonomous Systems xxx (2005) xxx–xxx

Robotics and Autonomous Systems
www.elsevier.com/locate/robot

Polynomial-based obstacle avoidance techniques for nonholonomic mobile manipulator systems

Evangelos Papadopoulos[a,*], Iakovos Papadimitriou[b,1], Ioannis Poulakakis[c,1]

[a] Department of Mechanical Engineering, National Technical University of Athens, Athens 157 80, Greece
[b] Department of Mechanical Engineering, University of California, Berkeley, CA 94720, USA
[c] Department of Mechanical Engineering, McGill University, Montreal, Que., Canada H3A 2A7

Received 12 December 2003; received in revised form 28 March 2005; accepted 30 March 2005

Abstract

A planning methodology for nonholonomic mobile manipulators in the presence of obstacles is developed. The method employs smooth and continuous functions, such as polynomials, and it is very fast, easy to use and computationally inexpensive. The core of the method is based on mapping the nonholonomic constraint to a space where it can be satisfied trivially. In this paper, the method is first extended to include polygonal obstacles of any kind, allowing for less conservative workspace representations. The algebraic nature of the methodology and its advantages are retained. To improve the performance of the method in finding collision-free paths with smaller length, two techniques are studied in detail. The first uses intermediate path points and the second exploits the periodicity of the trigonometric functions involved. The proposed methodology is also extended to the case of obstacles that are moving in the workspace with a priori known trajectories. This case is illustrated by an example of great application interest, in which the end-point follows a desired Cartesian trajectory while the platform and the manipulator follow valid and collision-free paths connecting given initial and final points. Additional illustrative examples demonstrate the planning methodologies in a variety of obstructed spaces.

Keywords: Mobile manipulators; Nonholonomic; Obstacle avoidance; Path planning; Phafians

1. Introduction

Mobile robots are of great importance to applications that involve inhospitable or remote environments, inaccessible or dangerous to humans. Typical examples can be found in mining, forestry, planetary exploration, security, etc. Although robots exist that use legs for locomotion, the most common terrestrial and space

* Corresponding author. Tel.: +30 210 772 1440; fax: +30 210 772 1455.
E-mail addresses: egpapado@central.ntua.gr (E. Papadopoulos), iakovos@mechatro2.me.berkeley.edu (I. Papadimitriou), poulakas@cim.mcgill.ca (I. Poulakakis).
[1] This work was completed while these authors were with the Department of Mechanical Engineering, at the National Technical University of Athens.

0921-8890/$ – see front matter
doi:10.1016/j.robot.2005.03.006

exploration robot platforms are wheeled. Tasks that go beyond inspection require a manipulator on-board. In this paper, planning in the presence of obstacles for mobile manipulators, i.e. mobile platforms equipped with manipulators, is studied.

Research in the area of mobile manipulators typically concentrates on point-to-point motion planning of the integrated system, mainly in obstacle-free environments, or deals with effects due to the coupling between the manipulator and its mobile platform. On the other hand, work in obstacle avoidance and navigation in cluttered environments typically deals with wheeled platforms alone.

Complete algorithms for solving the path planning problem for holonomic robots navigating in cluttered environments are available in the literature [1]. However, most of these algorithms are not directly applicable to systems that exhibit nonholonomic behavior, such as wheeled platforms. This is due to the fact that in nonholonomic systems, the number of degrees-of-freedom (d.f.) is less than the dimension of the configuration space. Therefore, a path that lies completely in the admissible space may not be realizable by the system. A comprehensive survey of developments in motion planning and control of nonholonomic systems can be found in [2], while a collection of papers concerning the open-loop motion planning problem for nonholonomic systems can be found in [3].

Most of the nonprobabilistic obstacle avoidance methods for nonholonomic wheeled mobile platforms can be roughly categorized into search-based methods, geometric approaches and artificial potential field methods. Barraquand and Latombe used an exhaustive search-based method that explores a system's configuration space by propagating step motions corresponding to some controls [4]. In most of the geometric methods, the final path computed by a planner is the concatenation of elementary paths computed by a basic procedure. Jacobs and Canny present a complete algorithm for calculating approximate collision-free trajectories with minimum turning radius and no reversals, using a set of canonical trajectories that satisfy the constraints, such as straight-line segments followed by arc segments [5]. Laumond et al. present a complete and exact path planner for wheeled platforms with lower bounded turning radius [6]. This planner uses the same families of canonical trajectories to transform a path, which was calculated by a geometric planner ignoring motion constraints, into a feasible one.

It is important to note here that most of the above methods are specific to wheeled platforms and cannot be applied to general classes of nonholonomic systems, because the admissible paths are not known a priori. To the best of our knowledge, the most general result is due to Sekhavat and Laumond [7], where the authors show how the algorithm developed in [6] can be extended for a class of nonholonomic systems that are or can be transformed into chained form. Other approaches to the obstacle avoidance problem for nonholonomic platforms without manipulators include dynamic programming techniques [8,9], progressive constraints [10] and least square approximations of paths returned by a holonomic planner based on artificial force fields [11].

Results in the area of mobile manipulators have concentrated on issues related to the coupling between the manipulator and the platform. Many of the approaches proposed exploit the kinematic redundancy of mobile manipulators using optimization techniques, so that the system attains configurations that satisfy constraints or minimize some criterion, e.g. [12,13]. Nevertheless, only limited results exist, in which obstacles, as environment-imposed constraints, are explicitly brought into play. Yamamoto and Yun proposed a method for obstacle avoidance in which they assumed that only the manipulator and not the platform may encounter the obstacle [14]. The developed controller allows the system to retain optimal or sub-optimal configurations while the manipulator avoids obstacles using potential functions. On the other hand, Ogren et al. proposed a method assuming that only the platform and not the manipulator may encounter an obstacle [15]. In their method, the end-point follows a given desired trajectory in a proven stable way, while at the same time the base motions are generated so that it will not collide with an obstacle. Perrier et al. represent the nonholonomy of the vehicle as a constrained displacement and try to make the global feasible displacement of the system correspond to the desired one [16]. Tanner et al. studied the problem of obstacle avoidance by the entire mobile manipulator system, and they proposed a general nonholonomic motion planning methodology based on a discontinuous feedback law under the influence of a potential field [17]. The method was applied to the case of many mobile manipulators handling a deformable body. A computationally inexpensive, polynomial-based planning methodology for mobile manipulators that also allows for obstacle avoidance was first introduced in [18,19].

This paper focuses on developing motion planning techniques for nonholonomic mobile manipulators operating in obstructed environments. The proposed method employs polynomial functions, although any smooth function is an equally valid candidate, to construct collision-free paths that move the mobile manipulator from any initial configuration to a final desired one. As was first shown in [19], this can be done by increasing the order of the polynomials that are used in planning trajectories, and then selecting the additional coefficients based on a systematic procedure so that the integrated system is guaranteed to avoid all the obstacles. In this paper, the obstacle avoidance principle presented in [19] is further studied and the basic methodology is expanded in two ways. First, the method is extended to include any kind of polygonal obstacles with the immediate advantage of allowing for more realistic representations of obstructed workspaces, and thus for tighter maneuvers. The algebraic nature of the method is retained and the time to compute collision-free paths increases only linearly with the number of obstacles and sides considered. Second, the method is expanded to accommodate nonstationary obstacles that move along known trajectories in the system's workspace. This situation also arises implicitly by incorporating manipulator joint limits while requiring the end-point to follow some desired pre-specified trajectories. The paper also discusses techniques that improve the performance of the method, such as the selection of intermediate points and the exploitation of the periodicity of platform orientation. These techniques result in increased flexibility and improve performance both by finding collision-free paths in cases where the original method fails and by considerably decreasing the length of the calculated paths. Finally, the method is successfully implemented to a large variety of simulated motion planning problems involving cluttered environments, including the parallel parking and crack-sealing problems.

2. System kinematics and nonholonomic constraint mapping

For simplicity, this paper focuses on a mobile system, which consists of a two degree-of-freedom manipulator mounted on a differentially driven mobile platform, see Fig. 1. However, the developed methodology can be applied to systems with N d.f. manipulators, or to car-like mobile platforms.

2.1. Manipulator subsystem

The mobile system consists of two subsystems, the holonomic manipulator and its nonholonomic base. The Cartesian coordinates of joint H and end-point E

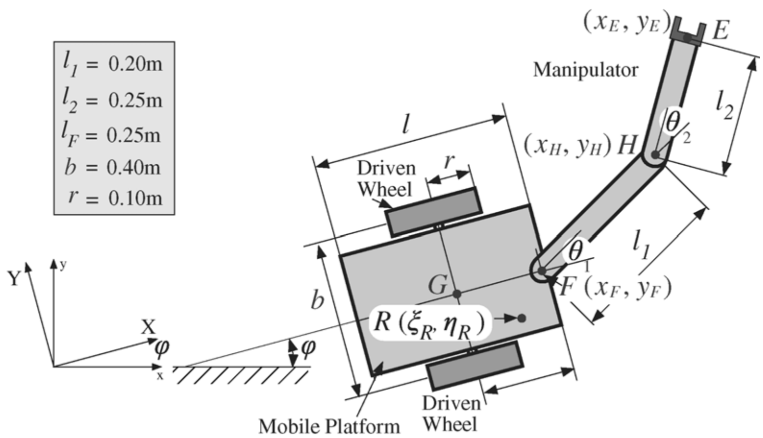

Fig. 1. Mobile manipulator system with a differentially driven platform.

with respect to the world frame, see Fig. 1, are given by,

$$\begin{aligned} x_{\mathrm{H}} &= x_{\mathrm{F}} + l_1 \cos(\varphi + \vartheta_1) \\ y_{\mathrm{H}} &= y_{\mathrm{F}} + l_1 \sin(\varphi + \vartheta_1) \end{aligned} \tag{1}$$

$$\begin{aligned} x_{\mathrm{E}} &= x_{\mathrm{F}} + l_1 \cos(\varphi + \vartheta_1) + l_2 \cos(\varphi + \vartheta_1 + \vartheta_2) \\ y_{\mathrm{E}} &= y_{\mathrm{F}} + l_1 \sin(\varphi + \vartheta_1) + l_2 \sin(\varphi + \vartheta + \vartheta_2) \end{aligned} \tag{2}$$

where $(x_{\mathrm{F}}, y_{\mathrm{F}})$ is the position of the mounting point F of the mobile platform, φ the platform orientation, ϑ_1 and ϑ_2 represent the manipulator joint angles, and l_1 and l_2 denote the manipulator link lengths. Eqs. (1) and (2) show that the end-point position depends on the position of the mounting point and on the orientation of the platform. If the configuration of the mobile platform is known, one can plan manipulator trajectories according to well-established methods. Therefore, solving the platform planning problem facilitates greatly the planning of manipulator trajectories.

2.2. Mobile platform subsystem

As shown in Fig. 1, the mobile platform is driven by two independent wheels. We assume that the speed, at which the system moves is low, and therefore the two driven wheels do not slip. This constraint, written for the manipulator mounting point F, is described by

$$\dot{x}_{\mathrm{F}} \sin\varphi - \dot{y}_{\mathrm{F}} \cos\varphi + \dot{\varphi} l = 0 \tag{3}$$

where l is the distance between points G and F, see Fig. 1. This constraint can be used for other platform points too. However, writing the constraint for F facilitates the analysis because this point appears in the manipulator kinematic equations, Eq. (2).

It has been proven that the nonholonomic constraint of the differentially driven mobile platform given by Eq. (3) can be written as [19],

$$\mathrm{d}u + v\,\mathrm{d}w = 0 \tag{4}$$

where u, v, w are properly selected functions of the platform position and orientation, x_{F}, y_{F} and φ. One set of functions that can be used is given by [19],

$$u(x_{\mathrm{F}}, y_{\mathrm{F}}, \varphi) = x_{\mathrm{F}} \sin\varphi - y_{\mathrm{F}} \cos\varphi \tag{5}$$

$$v(x_{\mathrm{F}}, y_{\mathrm{F}}, \varphi) = l - x_{\mathrm{F}} \cos\varphi - y_{\mathrm{F}} \sin\varphi \tag{6}$$

$$w(x_{\mathrm{F}}, y_{\mathrm{F}}, \varphi) = \varphi \tag{7}$$

Eqs. (5)–(7) constitute a transformation (x_{F}, y_{F}, φ) → (u, v, w), which is defined at every point of the configuration space. This transformation greatly facilitates path planning. Indeed, if we choose functions f and g as follows

$$w = f(t) \tag{8}$$

$$u = g(w) \tag{9}$$

$$v = -\frac{\mathrm{d}u}{\mathrm{d}w} = -g'(w) \tag{10}$$

then Eq. (4) is satisfied identically. Therefore, the planning problem reduces to the selection of functions f and g such that they satisfy the initial and final configuration conditions. Such functions can be polynomials, splines, or any other continuous and smooth time functions. For example, one possibility is to choose function f as a fifth-order polynomial, so that the platform initial and final angle, velocity and acceleration can be specified, and function g as a third-order polynomial, so that initial and final platform positions can be specified. Once u, v and w have been found, the platform coordinates are computed by inverting Eqs. (5)–(7). The complete methodology has been illustrated in detail in a previous paper [19].

3. Mapping to the *u*–*v*–*w* space

In this section, we study how obstacles and points on the platform and the manipulator in the Cartesian space are mapped through the transformation given by Eqs. (5)–(7). The mapping from a two- to a three-dimensional space adds one-dimension, which in this case, corresponds to the orientation of the platform. Therefore, an obstacle is mapped to a family of obstacles whose members are identified by the value of the orientation angle of the platform.

3.1. Obstacle mapping

It is assumed that the location of obstacles in the system workspace is known and fixed. However, it will be shown later that the methodology can be easily expanded to include the avoidance of moving obstacles,

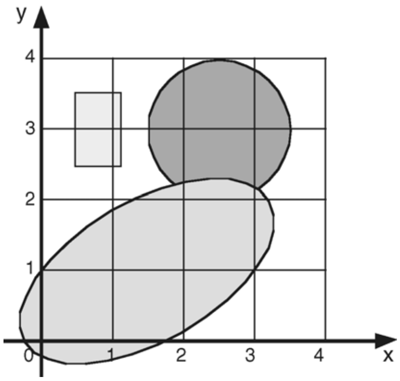

Fig. 2. Obstacles in the Cartesian x–y space.

provided that their trajectories are known or can be estimated.

The nature of the transformation given by Eqs. (5)–(7) can be understood better by decomposing it using homogeneous transformations as

$$\begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\pi/2) & -\sin(\pi/2-\varphi) & 0 & 0 \\ \sin(\pi/2-\varphi) & \cos(\pi/2-\varphi) & 0 & -l \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{\mathrm{F}} \\ y_{\mathrm{F}} \\ \varphi \\ 1 \end{bmatrix} \Rightarrow \boldsymbol{u} = \boldsymbol{T}_1\boldsymbol{T}_2\boldsymbol{x} \tag{11}$$

One can observe that matrix $\boldsymbol{T}_1$ corresponds to a reflection, while matrix $\boldsymbol{T}_2$ to a rotation by $\pi/2-\varphi$ and a translation by $-l$. Therefore, this transformation constitutes a global diffeomorphism in the configuration space that preserves both the length and the shape of an obstacle. For example, Fig. 2 depicts an elliptic, a circular and a rectangular obstacle in the Cartesian x–y space. Using the transformation described by Eqs. (5)–(7), these are transformed to the obstacles depicted in Fig. 3. It can be seen that for some $w=\varphi$, the obstacles in the u–v–w space are still an ellipse, a circle and a rectangle, while the centers of all families of obstacles lie on helicoids.

3.2. Platform and manipulator mapping

The transformation developed above refers to the manipulator mounting point F of the platform. However, when it comes to obstacle avoidance, it is obvious that other points of the platform and of the manipulator must be taken into account to ensure obstacle avoidance for the whole system. Therefore, in this section we study how a point on the platform or on the manipulator is mapped through the transformation for point F given by Eqs. (5)–(7).

To this end, we consider point R on the platform, with coordinates $(\xi_{\mathrm{R}}, \eta_{\mathrm{R}})$ expressed in the coordinate frame (ξ, η) parallel to the platform coordinate frame (X, Y) with origin at point F, see Fig. 1. Its Cartesian coordinates relative to the world frame are given by,

$$x_{\mathrm{R}} = x_{\mathrm{F}}(\xi_{\mathrm{R}}\cos\varphi - \eta_{\mathrm{R}}\sin\varphi) \tag{12a}$$

$$x_{\mathrm{R}} = x_{\mathrm{F}}(\xi_{\mathrm{R}}\sin\varphi - \eta_{\mathrm{R}}\cos\varphi) \tag{12b}$$

Substituting Eq. (12) into Eqs. (5)–(7) for both points F and R and after simple manipulations we conclude that:

$$u_{\mathrm{R}}(w) = u(w) - \eta_{\mathrm{R}} \tag{13a}$$

$$v_{\mathrm{R}}(w) = v(w) - \xi_{\mathrm{R}} \tag{13b}$$

Eq. (13) can be used to take into consideration additional points of interest when planning a collision-free path, e.g. corners of the vehicle.

In a similar way, using point F transformation coordinates and Eqs. (1) and (2), the manipulator arm is mapped in the u–v space, according to the following equations

$$\begin{aligned} u_{\mathrm{H}}(w) &= u_{\mathrm{F}}(w) - l_1\sin\vartheta_1 \\ v_{\mathrm{H}}(w) &= v_{\mathrm{F}}(w) - l_1\cos\vartheta_1 \end{aligned} \tag{14}$$

$$\begin{aligned} u_{\mathrm{E}}(w) &= u_{\mathrm{F}}(w) - l_1\sin\vartheta_1 - l_2\,\sin(\vartheta_1+\vartheta_2) \\ v_{\mathrm{E}}(w) &= v_{\mathrm{F}}(w) - l_1\cos\vartheta_1 - l_2\,\cos(\vartheta_1+\vartheta_2) \end{aligned} \tag{15}$$

4. Obstacle avoidance

As was seen in Section 2, the motion planning problem reduces to choosing two continuous and smooth functions f and g which satisfy the initial and final configuration conditions.

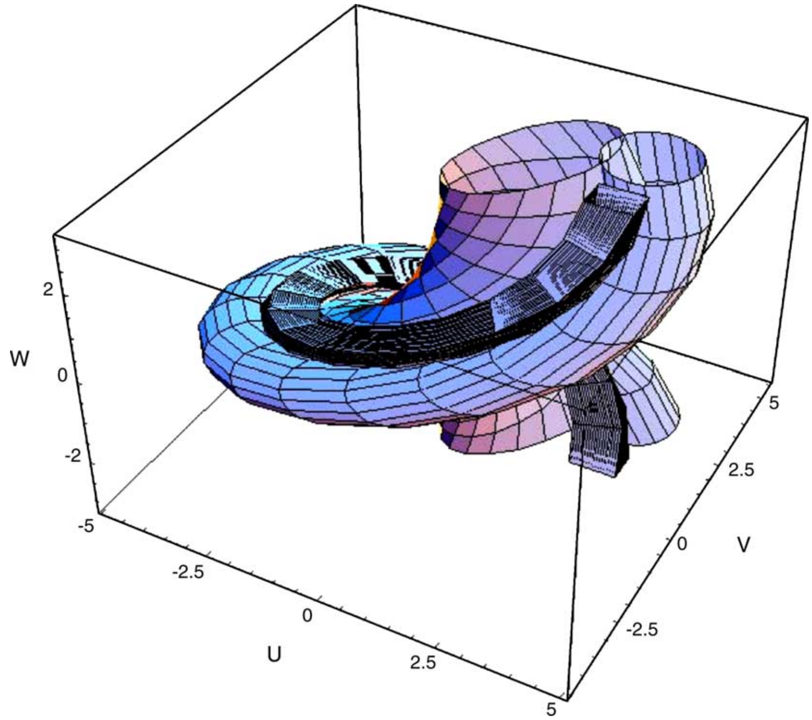

Fig. 3. The obstacles in Fig. 2 transformed in the u–v–w space.

This results to a path in the u–v–w, which satisfies the initial and final conditions of the system. This path can then be transformed to the Cartesian space by inverting the transformation given by Eqs. (5)–(7). This inverse always exists and yields the Cartesian path that the system will follow to reach the target configuration. Using point F as a reference point for motion planning one has

$$x_F(u, v, w) = u \sin w + (l - v) \cos w \tag{16a}$$

$$x_F(u, v, w) = u \cos w + (l - v) \sin w \tag{16b}$$

$$\varphi(u, v, w) = w. \tag{16c}$$

If obstacles exist in the system workspace, the path must be flexible enough to satisfy the boundary conditions and avoid the obstacles. In this case, functions f and g in Eqs. (8)–(10) are selected to be of fifth- and fourth-order polynomials, respectively

$$f(t) = a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 \tag{17}$$

$$g(w) = b_4w^4 + b_3w^3 + b_2w^2 + b_1w + b_0 \tag{18}$$

The six coefficients for $f = \varphi$ allow for setting any platform initial and final orientation, velocity and acceleration. As mentioned earlier, to satisfy the initial and final conditions for the platform position and velocity, only four coefficients are needed in $g(w)$, namely b_0–b_3. The additional coefficient b_4 allows for path shaping so that obstacles are avoided. If desired, one may add in $g(w)$ additional coefficients for additional path flexibility. Hence, the problem of avoiding Cartesian obstacles is reduced to the problem of finding an admissible region for the additional coefficient b_4 and selecting an appropriate value for it according to some criterion.

Here, for the sake of clarity, elliptic and polygonal obstacles are considered. The case of finding admissible values for the additional coefficient b_4 in the presence of an elliptic obstacle, centered at (x_0, y_0) with principal axis R_a and R_b, and rotated by an angle ψ, is addressed first. Circular obstacles constitute a special case of elliptic obstacles with $R_a = R_b = R$ and ψ, any.

The principle of the method for ellipses lies on the fact that collision is avoided when certain critical points of the robot lie outside the ellipse. Making use of the fact that an elliptic obstacle is mapped in the u–v–w space onto an ellipse with the same principal axes, then for each $w = \varphi$ the following inequality must hold,

$$R_b^2[(u(w) - u_0) \cos\psi' - (v(w) - v_0) \sin\psi']^2 + R_a^2 [(u(w) - u_0) \sin\psi' + (v(w) - v_0) \cos\psi']^2 - R_a^2 R_b^2 > 0 \quad (19)$$

where $u_0(w)$, $v_0(w)$ are the coordinates of the transformed center of the ellipse, $u(w)$, $v(w)$ are the coordinates of the point of interest, and are all functions of w as expected, and $\psi' = \psi - \varphi$ is the transformed angle of the ellipse.

For a given set of system boundary conditions, and due to Eqs. (9) and (10), $u(w)$ and $v(w)$ are linear functions of the coefficient b_4. Then, following some algebraic manipulations, Eq. (19) yields,

$$\alpha b_4^2 + \beta b_4 + \gamma > 0 \quad (20)$$

The coefficients α, β and γ are known functions of w and of the boundary conditions $u_{\rm in}$, $v_{\rm in}$, $w_{\rm in}$, $u_{\rm fin}$, $v_{\rm fin}$, $w_{\rm fin}$ computed via the transformation given by Eqs. (5)–(7) and the boundary conditions in the Cartesian space. Eq. (20) is a very practical representation of the criterion for obstacle avoidance. If this inequality holds for all $w = \varphi$, then the planned path of the system point of interest will never collide with the obstacle. If additional system points must be considered, then additional inequalities in b_4 result, possibly further limiting the range of admissible b_4.

Notice that since Eq. (20) is essentially a distance criterion and since $u(w)$ and $v(w)$ are linear functions of b_4, the left side of the resulting inequality will always be a second-order polynomial in b_4. This fact simplifies the problem of finding appropriate values of b_4 for which Eq. (20) is satisfied. Indeed, it can be shown that coefficient a is always a positive number, and therefore satisfaction of Eq. (20) requires that b_4 lies outside of the second-order polynomial roots, $b_4^a(w)$ and $b_4^b(w)$.

Next, polygonal obstacles are considered. These can be described in the Cartesian space by a closed sequence of linear segments of the form $y_{(i)} = c_{1(i)}x + c_{2(i)}$, $x \in [x_{1(i)}, x_{2(i)}]$, where $c_{1(i)}$ and $c_{2(i)}$ are constant coefficients representing the slope and position of each distinct linear segment, respectively, $x_{1(i)}$, $x_{2(i)}$ are the end-points of each distinct linear segment and i is the number of polygon linear segments. Obstacle avoidance is guaranteed when a mobile manipulator point under consideration, with transformed coordinates $u_{\rm R}(w)$ and $v_{\rm R}(w)$, does not belong to any of these linear segments. Therefore, using Eq. (16), the following must hold true for all segments

$$u_{\rm R}(w) + \frac{c_{1(i)} \cos w - \sin w}{c_{1(i)} \sin w + \cos w}(l - v_{\rm R}(w)) + \frac{c_{2(i)}}{c_{1(i)} \sin w + \cos w} \neq 0 \quad (21)$$
$$\forall w \in [w_{\rm in}, w_{\rm fin}]$$
$$\forall (u_{\rm R}(w) \sin w + (1 - v_{\rm R}(w)) \cos w) \in [x_{1(i)}, x_{2(i)}]$$

Again bearing in mind that $u_{\rm R}(w)$, $v_{\rm R}(w)$ are linear functions of b_4, Eq. (21) results in

$$b_4 \neq p_{(i)}(w) \quad (22)$$

where the quantities $p_{(i)}(w)$ are again known functions of w and of the boundary conditions.

To ensure obstacle avoidance of the multi-body structure, we need to consider the points on the platform and the manipulator, whose obstacle avoidance guarantees the obstacle avoidance of the mobile manipulator as a whole. More specifically, we first check whether the vertices of the platform as well as its edges collide with the obstacles, according to all the possible collision scenarios. This is repeated for the arm links and joints.

4.1. Platform obstacle avoidance

Fig. 4 depicts possible collision cases for the platform. These have been classified as cases (a–f). Note that the case of a linear segment being in parallel contact with the platform falls either under case (c), if at least one of the end-points of the segment is outside of the platform bounds, or under case (f), if none of the segment end-points is outside of the platform bounds. To illustrate the method, we describe the analysis for the cases of circular and polygonal obstacles in detail. Similar results hold for elliptic obstacles, but are not presented here due to their lengthy equations.

To ensure obstacle avoidance, platform vertices must be mapped in the u–v space and the admissible range of b_4 must be determined. The platform-mapping problem has been described in Section 3. Thus, making use of the results presented in that section, we conclude

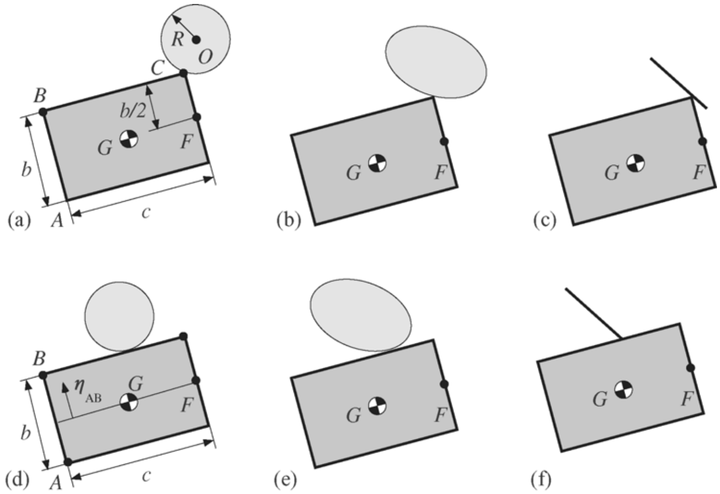

Fig. 4. Collision cases between an elliptic, a circular and a line segment object.

that for a vertex, e.g. point B in Fig. 4(a), located at a certain distance from point F, the range of admissible b_4, is calculated by the following inequality:

$$\left(u(w) - \frac{b}{2} - u_0(w)\right)^2 + (v(w) + l - v_0(w))^2 > R^2 \quad \forall w \in [w_{\text{in}}, w_{\text{fin}}] \tag{23}$$

where $u_0(w)$ and $v_0(w)$ are the transformed coordinates of the center of the obstacle for some platform orientation $w = \varphi$, R the radius of the circular object, and l and b are the dimensions of the platform. The transformed coordinates of point F, $u(w)$ and $v(w)$, are linear functions of the coefficient b_4, which implies that Eq. (23) can be algebraically manipulated to take the form of a second-order polynomial in b_4, such as the one in Eq. (20). Finding a path that prevents collision with vertex B requires a selection of b_4 outside the roots of this polynomial. Clearly, the same procedure has to be repeated for all vertices of the platform.

As far as the platform edges are concerned, we check when the linear segment, which represents a platform edge, becomes tangent to the circular obstacle. For example, let us assume that we want to find collision-free paths for the edge AB of the platform, as depicted in Fig. 4(d). According to the analysis in Section 3, the edge's transformation in the u–v space is given by

$$u_{\text{AB}}(w) = u(w) - \eta_{\text{AB}} \qquad \forall \eta_{\text{AB}} \in \left[-\frac{b}{2}, \frac{b}{2}\right] \tag{24a}$$

$$v_{\text{AB}}(w) = v(w) + l \tag{24b}$$

Thus, the inequality that must hold true in order to ensure obstacle avoidance is:

$$(u(w) - \eta_{\text{AB}} - u_0(w))^2 + (v(w) + l - v_0(w))^2 > R^2 \quad \forall\, w \in [w_{\text{in}}, w_{\text{fin}}], \qquad \eta_{\text{AB}} \in \left[-\frac{b}{2}, \frac{b}{2}\right] \tag{25}$$

After some algebraic manipulation, the above inequality is written as:

$$\alpha\eta_{\text{AB}}^2 + \beta\eta_{\text{AB}} + \gamma > 0 \tag{26}$$

where the coefficients α, β and γ are known functions of b_4, w and of the boundary conditions. Since $\alpha > 0$, Eq. (26) will hold for all $\eta_{\text{AB}} \in [-b/2, b/2]$ if it does not have any real roots. In the limit, the platform side AB becomes tangent to the circular obstacle when the following condition holds,

$$\beta^2 - 4a\gamma = 0 \text{ for } \eta_{\text{AB}} \in \left[-\frac{b}{2}, \frac{b}{2}\right] \tag{27}$$

The above equation is actually a second-order polynomial in b_4 whose coefficients are known functions of w and of the boundary conditions. Thus, its roots, if calculated for every w, yield b_4 for which the side of the platform becomes tangent to the circular obstacle. Selecting b_4 outside this range ensures a collision-free path. It must be mentioned here that the procedure described above has to be repeated for every side of the platform; however, the computational burden rises only linearly with the number of edges. The case of elliptic obstacles is similar to that of the circular obstacles described above and, therefore it is not presented here.

Finally, for polygonal obstacles, we observe that a collision occurs either when one of the vertices of the platform collides with one of the linear segments of the polygonal obstacle, or when one of the edges of the platform collides with one of the end-points of the linear segment, see Fig. 4(c and f). The former case can be tackled by the use of Eq. (22) for every platform vertex. The latter case is similar; specifically, each end-point of the polygon's linear segments is mapped onto the u–v–w space and each platform edge is parameterized as in Eq. (24). For every $w \in [w_{\mathrm{in}}, w_{\mathrm{fin}}]$, obstacle avoidance is achieved when the mapped end-points do not belong to the parameterized platform edges.

4.2. Manipulator arm obstacle avoidance

During system motion, the manipulator may be moving with respect to its base. The relative manipulator motion (path and trajectory) can be planned easily using n boundary conditions, e.g. end-point initial and final position and velocity. A simple solution to this problem is to use polynomials of order $n-1$. Then, the n polynomial coefficients, and thus the joint path, are calculated by solving a system of n linear equations, resulting from the n boundary conditions. In the case where obstacles are present, we employ the redundancy offered by the base and modify its path so that the entire system will not collide with the obstacles.

To this end, we map the moving manipulator into the u–v space, as was shown in Section 3. Note that according to Eqs. (14) and (15), the mobile manipulator changes its configuration during motion. However, the method calculates the coefficients of the second-order polynomials and their roots for each time step. This procedure results in an admissible range of b_4 that holds for all times, and allows the selection of the value to be used. In conclusion, the methodology guarantees obstacle avoidance of the entire mobile system while the required computational time increases linearly with the number of manipulator vertices that are checked. This important property of the method will be clarified in the following example.

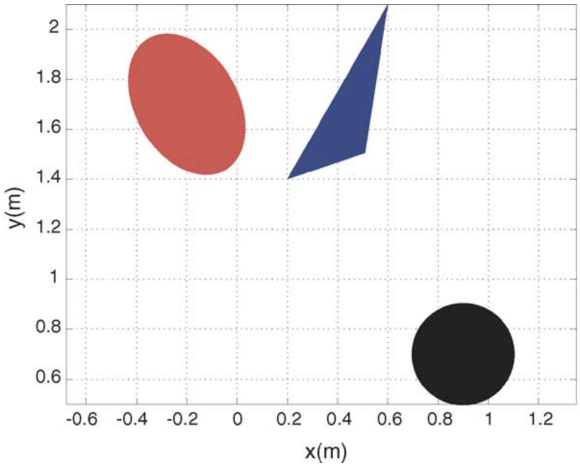

Fig. 5. Obstacles in the Cartesian Space.

Example 1. To illustrate the obstacle avoidance method described above, we consider the system depicted in Fig. 1 navigating in the workspace depicted in Fig. 5, where a circular, an elliptic and a triangular obstacle exist. For the simulation, the following values were chosen: total motion time 6 s, initial configuration $(x_{\mathrm{F}}^{\mathrm{in}}, y_{\mathrm{F}}^{\mathrm{in}}, \varphi^{\mathrm{in}}, \vartheta_1^{\mathrm{in}}, \vartheta_2^{\mathrm{in}}) = (-0.2\,\mathrm{m}, 0.5\,\mathrm{m}, -90^\circ, -40^\circ, -50^\circ)$ and final desired configuration $(x_{\mathrm{F}}^{\mathrm{fin}}, y_{\mathrm{F}}^{\mathrm{fin}}, \varphi^{\mathrm{fin}}, \vartheta_1^{\mathrm{fin}}, \vartheta_2^{\mathrm{fin}}) = (1\,\mathrm{m}, 2\,\mathrm{m}, 120^\circ, 40^\circ, -130^\circ)$. Note that choosing a different total time will make the system move faster or slower, but will have no effect on the Cartesian path of the platform.

A fifth-order polynomial is used to parameterize function f, i.e. the orientation of the platform w as a function of time and a fourth-order polynomial for function g, as given by Eqs. (17) and (18). Two fifth-order polynomials are used for the trajectories of the manipulator joint angles. Having made these calculations, the algorithm is ready to calculate the roots of the polynomials, which yield the range of admissible values for b_4.

More specifically, for the circular obstacle, the roots of 12 second-order polynomials are calculated, 8 corresponding to the platform (4 vertices and 4 edges), 2 to the first manipulator link (point H and link FH) and another 2 to the second link (point E and link HE). Similarly, for the elliptic obstacle, a set of 12 polynomials is calculated as well. Finally, for the triangle, 36 linear equations are solved, 12 corresponding to the

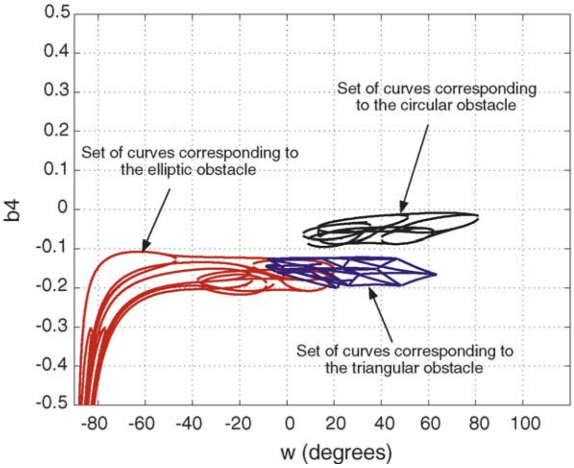

Fig. 6. Admissible range of b_4 for avoiding collisions between platform-manipulator and obstacles shown in Fig. 5.

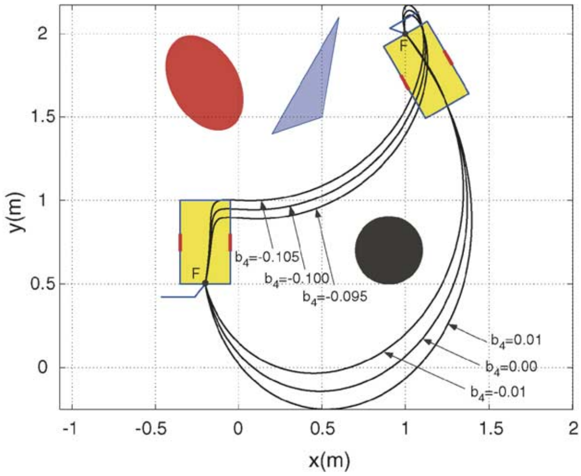

Fig. 7. Front point paths for some critical b_4 values corresponding to Fig. 6 results.

collision of the 4 platform edges with the 3 triangle vertices, another 12 corresponding to the collision of the 4 platform vertices with the 3 triangle edges, another 6 corresponding to the manipulator points, E and H, with the triangle edges and another 6 corresponding to the manipulator links with the triangle vertices.

Having completed this task, Fig. 6 is plotted. It shows that the admissible values for b_4 are $b_4 \in (-0.107, -0.095) \cup (-0.009, \infty)$.

As expected, admissible paths through the narrow passageway actually exist. Focusing on this figure, the upper isolated set of curves corresponds to constraints on b_4 induced by the circular object. In more detail, one may identify six ellipse-like curves corresponding to the polynomials of the four platform vertices and of manipulator points H and E. These ellipse-like curves are connected to each other with 12 curves that correspond to the polynomials of the 4 platform edges and 2 manipulator links (2 roots for each polynomial). The other set of curves in Fig. 6 correspond to the elliptical and triangular obstacles in Fig. 5, and are made of curves that are produced in a similar manner as explained with respect to the circular obstacle.

Fig. 7 depicts a number of point F paths for admissible values of b_4 that just miss the obstacles. Notice that the range of paths that pass between the triangular and the circular object is limited because both the triangular and the elliptical objects limit the range of backwards maneuvers available to the mobile system. However, there is basically no problem to reach the final configuration when all three obstacles are bypassed by moving initially forward and below the circular obstacle.

Fig. 8 depicts snapshots of the motion of the platform, which corresponds to the choice $b_4 = -0.1$. The mobile system goes through the passageway without colliding with the obstacles and successfully reaches its destination. Choosing b_4 at the boundaries of the admissible region would have resulted to a near hit motion of the mobile manipulator.

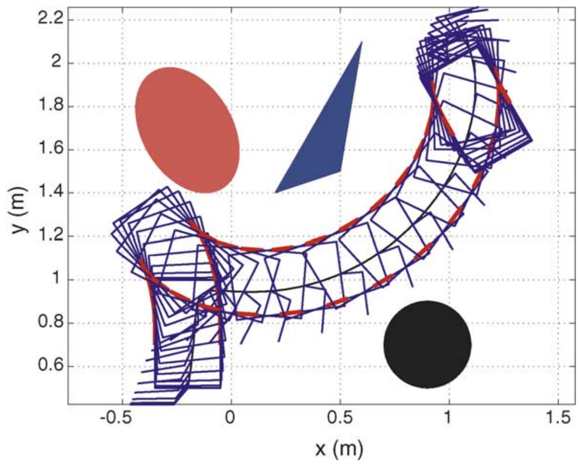

Fig. 8. Motion animation for a differentially driven mobile manipulator.

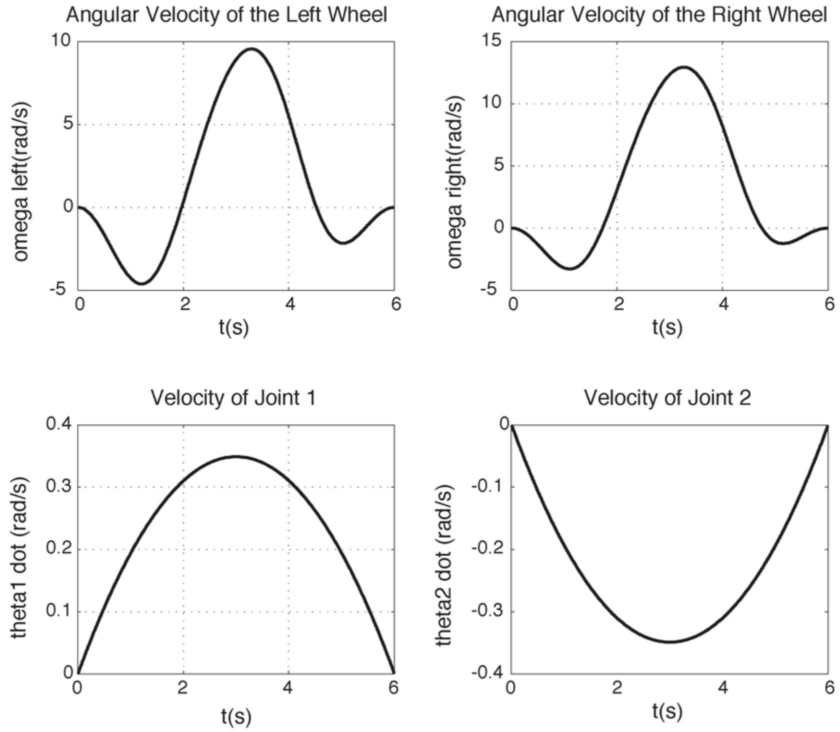

Fig. 9. Input velocities for the path shown in Fig. 8.

The resulting rotational velocities of the wheels and the manipulator joint speeds are depicted in Fig. 9. As expected, the use of polynomials results in continuous velocity profiles with smooth starts and stops and no excessive input velocities.

It should be noted at this point that since the method calculates the roots of the polynomial in b_4 at each time step, the method can account for not only stationary, but also moving obstacles, provided that we have an estimate of their future positions. Such an estimate can be calculated through machine vision and Kalman-filter-based algorithms. The details of this kind of detection system exceed the scope of this paper, however, an application of the method on moving obstacles is presented in Section 4.

It is also worth noting that more complex mobile manipulators require the solution of more equations. More specifically, a simple extension of the methodology reveals that, given a rectangular platform with a planar N-d.f. manipulator arm in a cluttered environment with j elliptical/circular obstacles and k polygonal obstacles with l vertices, requires the solution of $(j + kl)(8 + 2N)$ equations.

Finally, depending on the number of obstacles, the complexity of the platform geometry and the on-board computational power, the final selection of b_4 from the admissible range can be part of an optimization loop. For example, provided that sufficient computational power is available, the algorithm can calculate the admissible region of b_4 values, compute the system trajectory for each admissible value of b_4, and choose the one that will satisfy a criterion, such as minimum acceleration or travelled distance.

5. Implementation issues

The methodology developed above guarantees that if a solution can be found within the class of the polynomial functions used, then the integrated system of the platform and manipulator will avoid any obstacles present in its workspace. However, in some cases the path planner might yield collision-free paths that are long or it might not be able to find an admissible path at all. More specifically, the order of the polynomials used is closely related to the complexity of the path returned by the planner, i.e. the number of back and forth maneuvers. The higher the order of the polynomials is, the more complex and flexible the paths become, at the expense of added computations. For example, the use of fifth-order polynomials results in a pair of selectable parameters, b_4 and b_5, that directly affect the shape of the admissible paths. The best pair with respect to some path qualities can be an interesting optimization problem exceeding the scope of this paper.

Another issue that one must have in mind is related to the use of polynomials for function g. Indeed, the calculation of the coefficients of g is impossible when the initial orientation is equal to the final one, i.e. $\varphi_{\text{in}} = \varphi_{\text{fin}}$. Also, if these two values are very close, the method yields solutions in which the resulting path is long. Despite these issues, the application of the methodology can be extended to these cases, as well, with some simple additional modifications.

5.1. Intermediate point technique (IPT)

One way to tackle the above issues is to introduce one or more intermediate configurations, through which the platform must pass during its motion towards the final configuration. This technique is also useful when one wants to ensure that the system will pass through a specific point. The user can select an intermediate configuration according to some additional task-specific requirements. Note also that a major advantage of this technique is that introducing intermediate points leaves all calculations unchanged with the tradeoff of having the system stop momentarily at the intermediate point.

Example 2. Here, we assume the workspace depicted in Fig. 10, where there is a circular and an elliptic obstacle, forming a narrow passageway. For clarity and simplicity reasons we consider the platform only. The addition of the manipulator arm will not affect the results. For the simulation, the total move time is chosen equal to 6 s, the initial configuration is chosen as $(x_F^{\text{in}}, y_F^{\text{in}}, \varphi^{\text{in}}) = (-0.2\,\text{m}, 0.5\,\text{m}, 0°)$ and the final one as $(x_F^{\text{fin}}, y_F^{\text{fin}}, \varphi^{\text{fin}}) = (2\,\text{m}, 0.5\,\text{m}, 45°)$.

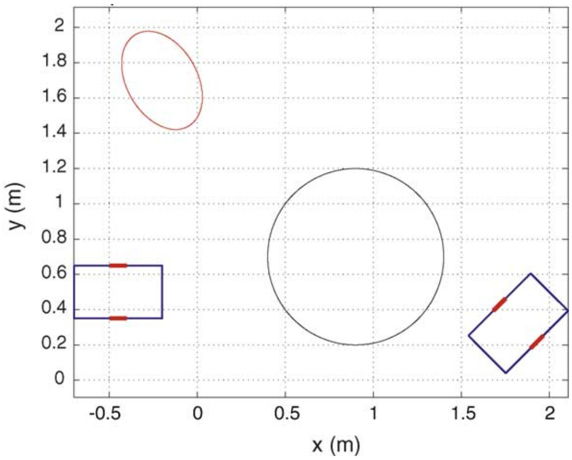

Fig. 10. Initial and final platform configuration in an obstructed workspace.

The region of admissible b_4 is shown in Fig. 11. It can be seen that a collision-free path is ensured when $b_4 \in (-\infty, -89.88) \cup (-50.86, -41.8l)$. The first region of admissible b_4 corresponds to long paths, which guide the platform above the obstacles, while the second corresponds to long paths through the narrow passageway. For instance, selecting $b_4 = -90$, yields a path whose length is $l_2 = 33.35$ m, see Fig. 12. It is evident that for this workspace and b_4 selection, the method leads to a very long path. Similar results are obtained for $b_4 \in (-50.86, -41.81)$.

Next, an intermediate point is introduced. The total move time is now 12 s and the intermediate

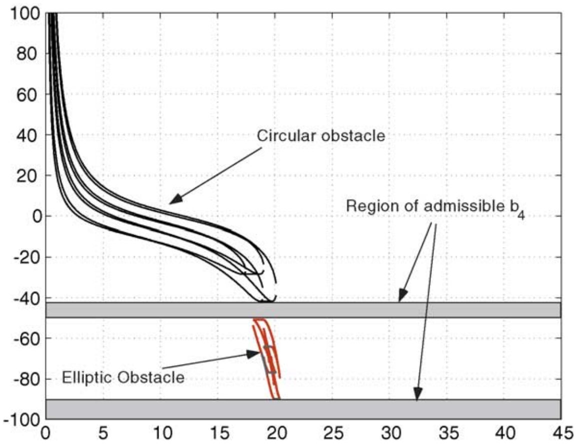

Fig. 11. Region of admissible b_4 for the workspace shown in Fig. 10.

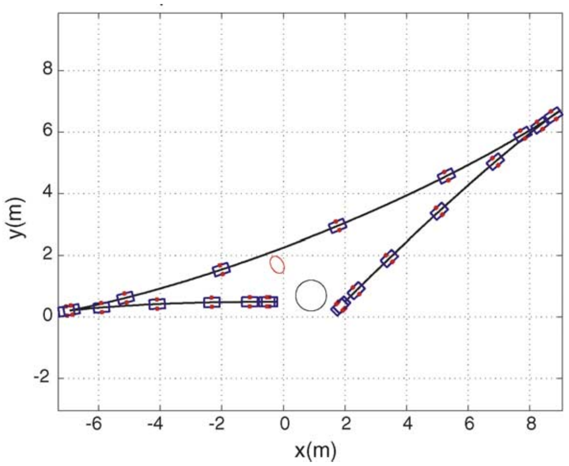

Fig. 12. Calculated path for $b_4 = -90$.

configuration is described by $(x_F^{tr}, y_F^{tr}, \varphi^{tr}) = (0.5\,\mathrm{m}, -0.5\,\mathrm{m}, 45^\circ)$. The admissible region of b_4 for each sub-path is $b_4 \in (-10.18, \infty)$ and $b_4 \in (-0.89, \infty)$. Selecting $b_4 = 0$ for both sub-paths yields the path shown in Fig. 13.

It is clear that the length of the path is now much shorter ($l = 3.87$ m). The wheels speed profiles are shown in Fig. 14. Indeed, these remain smooth while the platform stops momentarily at $t = 6$.

Example 3. The intermediate point technique can be employed to enhance the control over the path of the system. Such a capability is also important in parking or unparking problems. Let us consider the unparking problem and assume the workspace depicted in Fig. 15.

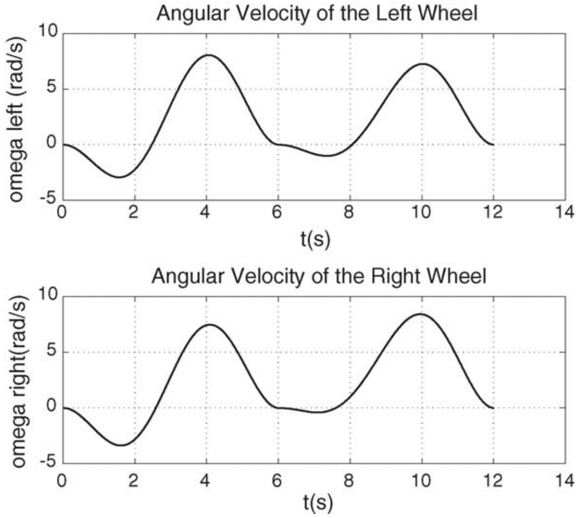

Fig. 14. Input velocities with the IPT for the path shown in Fig. 13.

For the simulation, the total move time is chosen equal to 12 s, while the initial configuration is $(x_F^{in}, y_F^{in}, \varphi^{in}) = (0.75\,\mathrm{m}, 0.2\,\mathrm{m}, 0^\circ)$. The final desired configuration of the system $(x_F^{fin}, y_F^{fin}, \varphi^{fin}) = (1.5\,\mathrm{m}, 0.8\,\mathrm{m}, 0^\circ)$. Notice that the initial and final base orientations are equal. Therefore, an intermediate point must be used. Introducing an intermediate configuration such as the one given by $(x_F^{tr}, y_F^{tr}, \varphi^{tr}) = (0.8\,\mathrm{m}, 0.6\,\mathrm{m}, 45^\circ)$, the problem is easily solved. The

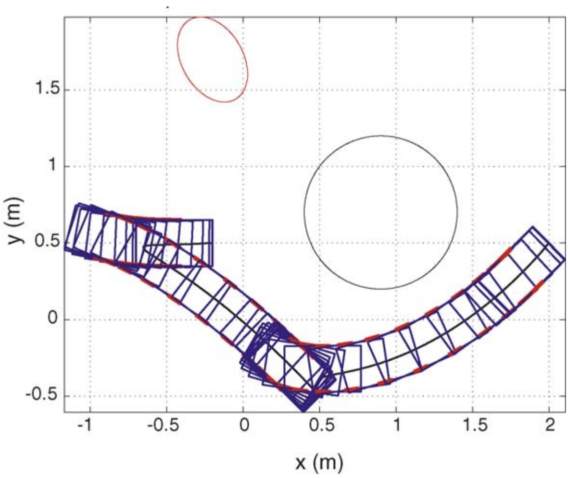

Fig. 13. Calculated path using the IPT technique.

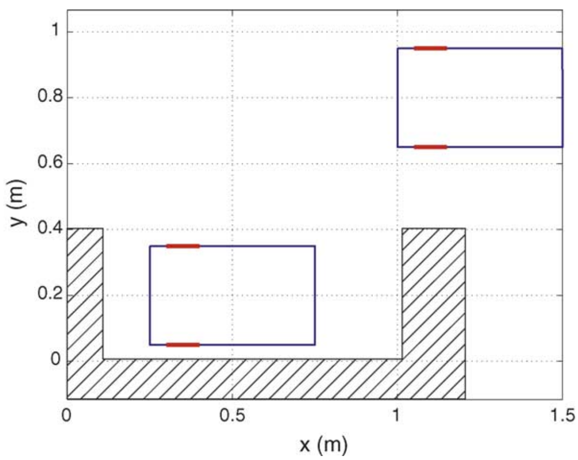

Fig. 15. Obstructed workspace simulating the unparking problem.

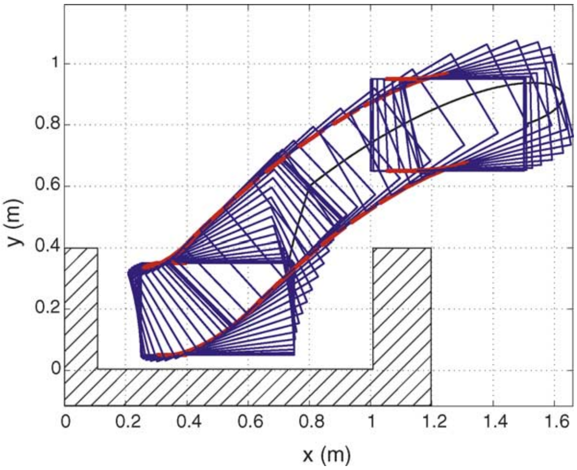

Fig. 16. Unparking using the IPT technique.

admissible regions of b_4 are $b_4 \in (-1.20, 0.91)$ and $b_4 \in (-\infty, 5.5)$ for each sub-path. Selecting $b_4 = 0$ for both sub-paths results in the successful unparking path depicted in Fig. 16.

5.2. Orientation periodicity technique (OPT)

An alternative way to expand the implementation of the method to complex environments and to cases where the initial and final orientations of the platform are the same is to take advantage of the periodicity of the trigonometric functions involved. Indeed, by adding full turn (360°) multiples, either to the initial or to the final vehicle orientation φ the boundary conditions are still respected while the algorithm yields solutions in one run (no intermediate points needed). Next, we present an example in which this technique is illustrated and compared with results obtained using the intermediate point technique.

Example 4. The workspace in this example is the one used in Example 2 and depicted in Fig. 10. The initial and final configuration are the same to the ones used in Example 2, but now the final orientation requested from the planner is $45^\circ + 360^\circ = 405^\circ$. This full turn addition does not violate the desired boundary conditions, while it allows more flexibility for the platform in moving through the obstacles and avoiding long paths. Going through the methodology as before, yields the admissible range for b_4 which is now $b_4 \in (-\infty, \ -5.86 \times 10^{-3}) \cup (2.62 \times 10^{-3}, \ 0.0178)$.

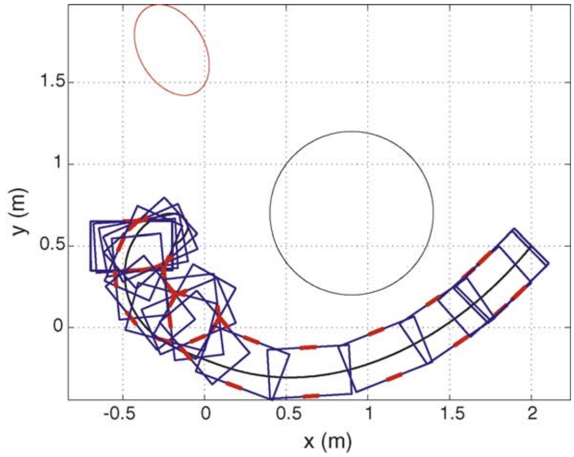

Fig. 17. Calculated path using the OPT and a single additional turn.

Selecting $b_4 = 0.01$ results in the path shown in Fig. 17, where the platform performs a full rotation in the beginning and continues until it reaches its destination. The length of the constructed path is $l = 3.86$ m.

Fig. 18 depicts the resulting wheel speed profiles, which as before are smooth. Compared to the IPT, this technique offers the benefit that the algorithm needs to be run only once, avoiding the process of re-running the algorithm when intermediate points are present. As for the length of the path, this remains small in both cases. Depending on the topology of the

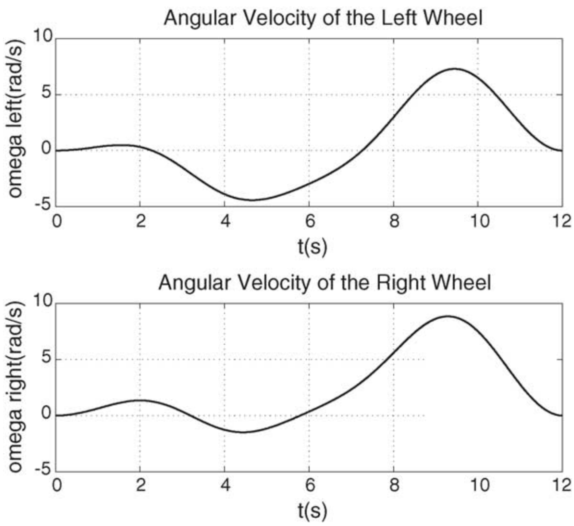

Fig. 18. Wheel velocities for the OPT path shown in Fig. 17.

workspace and the task requirements, one of these techniques may present an advantage compared to the other. However, in both techniques, the salient advantage of fast algebraic computations that scale linearly with the number of obstacles or sides taken into account is retained.

6. Moving obstacles and end-point trajectory planning

In this section, we first turn our attention to the problem of avoiding obstacles that move in the robot's workspace with known trajectories. Such information may be available, for example, by a mapping method or an off-line trajectory planner. Under these circumstances, the position of all obstacle vertices and sides are known functions of time. Therefore, all distance inequalities still have the form of Eq. (20), i.e. they result in quadratic forms with respect to b_4. In this case though, the a priori known obstacle coordinates as a function of time are used. As is the case with stationary obstacles, moving obstacles restrict the range of feasible b_4 coefficients, by producing constraint curves in the b_4–w space.

The methodology for dealing with moving obstacles can be employed in a host of situations. Among them, an important one is the planning of end-point trajectories when the manipulator is mounted on a nonholonomic platform. This particular situation arises in many practical robotic applications such as in the robotic crack-sealing, where the manipulator end-point must follow a certain crack on the pavement, see, e.g. [20]. In more detail, this problem involves constructing a path and a trajectory for the platform, so that (a) the manipulator avoids singularities at its workspace limits, (b) its end-point follows a given Cartesian trajectory and (c) the entire mobile manipulator system does not collide with proximal obstacles.

To study this problem, the end-point desired trajectory is parameterized by the following functions

$$x_{\mathrm{des}} = x_{\mathrm{des}}(t) \tag{28a}$$

$$y_{\mathrm{des}} = y(x_{\mathrm{des}}) = y(x_{\mathrm{des}}(t)). \tag{28b}$$

To keep the end-point within the manipulator workspace limits, the following equations must hold for all time t,

$$(x_{\mathrm{F}} - x_{\mathrm{des}})^2 + (y_{\mathrm{F}} - y_{\mathrm{des}})^2 < (l_1 + l_2)^2 \tag{29a}$$

$$(x_{\mathrm{F}} - x_{\mathrm{des}})^2 + (y_{\mathrm{F}} - y_{\mathrm{des}})^2 > (l_1 - l_2)^2 \tag{29b}$$

Eq. (29) imply that the manipulator mounting point F has to be inside a moving circle with radius $(l_1 + l_2)$ centered at the end-point location $(x_{\mathrm{des}}, y_{\mathrm{des}})$ and outside a co-centered circle with radius $|l_1 + l_2|$, at all times. The core of the technique lies on the fact that we can treat these two circles as moving obstacles and employ the obstacle avoidance methodology described previously, focusing on point F. In other words, the manipulator reach constraints described by Eq. (29) can be treated as "virtual" obstacles for point F. Next, the methodology for exact end-point following is outlined in detail.

Given the initial and final positions and velocities of the platform, we use the methodology described in Section 4 to map all obstacles in the b_4–w space. The same methodology is employed to map the manipulator reach constraints, i.e. the "virtual" obstacles, described by Eq. (29). Indeed, after some algebraic manipulations, the constraints given by Eq. (29) take the form of the standard second order inequality given by Eq. (20). As expected, this process further restricts the region of admissible b_4, which results in obstacle avoidance for the platform and keeps the end-point within its manipulator workspace. Selection of an appropriate b_4 yields a specific platform path. Knowing the trajectory of the end-point and that of the platform – and thus of the mounting point F – the manipulator joint trajectories are calculated using trivial inverse kinematics.

Since up to this point the manipulator itself has not been checked against collisions, the remaining task is to ensure that the computed joint trajectories and the links themselves do not interfere with the obstacles. However, this step is very simple and involves calculating the trajectory of point H using Eq. (1) and then checking if the linear segments (FH) and (HE) collide with any of the obstacles at every time step. If the manipulator motion results in collisions with obstacles, then the algorithm is repeated with a new b_4, which yields a new admissible trajectory for the platform and new manipulator joint motions. The technique iterates until a solution, which results in no collision with the obstacles, is found. The method is illustrated in the following example.

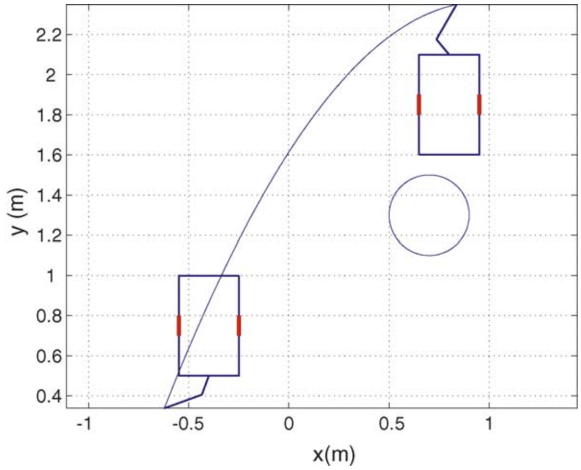

Fig. 19. Crack-sealing problem.

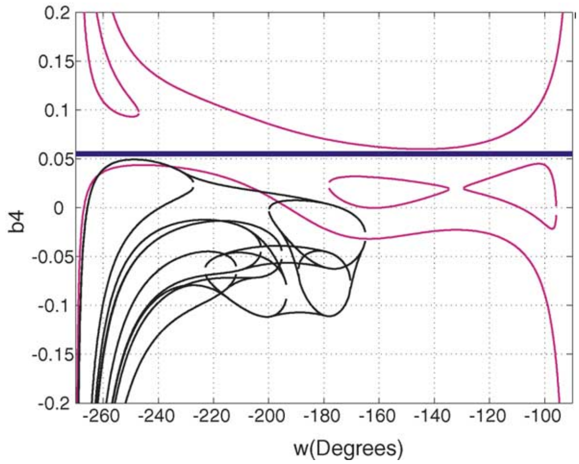

Fig. 20. Admissible b_4 for the crack-sealing problem shown in Fig. 19.

Example 5. In this example, the platform must reach a desired position and orientation while the end-point should follow a given Cartesian trajectory (e.g. representing a crack on a pavement), and collision with the circular obstacle is avoided. The desired trajectory is given by

$$x_{\text{des}}(t) = -0.0135t^3 + 0.1215t^2 - 0.622 \qquad (30a)$$

$$y_{\text{des}} = -0.80x_{\text{des}}^2 + 1.55x_{\text{des}} + 1.61 \qquad (30b)$$

The desired starting and stopping configuration for the platform and the end-point path are depicted in Fig. 19. For the simulation, the total move time is chosen equal to 6 s, the initial configuration is $(x_F^{\text{in}}, y_F^{\text{in}}, \varphi^{\text{in}}, \vartheta_1^{\text{in}}, \vartheta_2^{\text{in}}) = (-0.4\,\text{m}, 0.5\,\text{m}, -90^\circ, -20^\circ, -50^\circ)$ and the final configuration is $(x_F^{\text{in}}, y_F^{\text{in}}, \varphi^{\text{in}}, \vartheta_1^{\text{in}}, \vartheta_2^{\text{in}}) = (-0.8\,\text{m}, 2.1\,\text{m}, -270^\circ, -40^\circ, -70^\circ)$.

The first step is to calculate the roots of the polynomials concerning the circular obstacle by using the obstacle avoidance method described in Section 4. Then, we calculate the roots of the polynomials describing the collisions with the "virtual" obstacles that correspond to the manipulator reach constraints. All roots are plotted for each $w \in [w_{\text{in}}, w_{\text{fin}}]$ in Fig. 20. It is clear that the admissible values for b_4 are $b_4 \in (0.05, 0.06)$. This relatively thin region is mainly due to manipulator reach constraints, to its configuration with respect to the nonholonomic platform, and to the fact that during the sealing task, the latter has to change its orientation by 180°. Indeed, in the absence of the obstacle, this range would still be thin, $b_4 \in (0.045, 0.060)$.

Choosing $b_4 = 0.055$ and using inverse kinematics, the joint trajectories are calculated and possible collisions of the manipulator links are checked. In this case, no collisions occur. However, if collisions did occur, we would have to select another b_4 and check again. Having ensured a collision-free path, the trajectory for both the platform and the end-point results. Fig. 21 depicts snapshots of motion of the mobile manipulator system, while Fig. 22 depicts system wheel

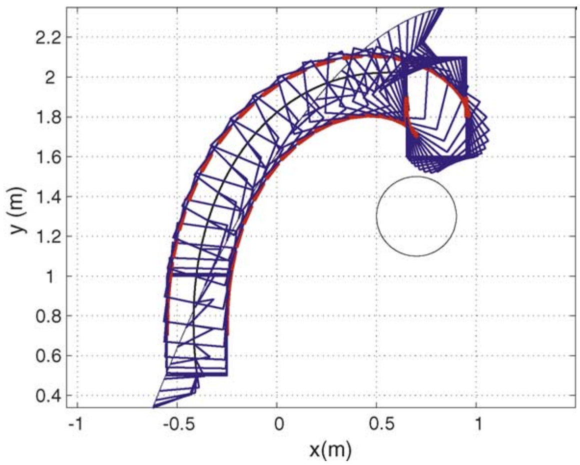

Fig. 21. Motion animation for the mobile manipulator for the crack-sealing problem.

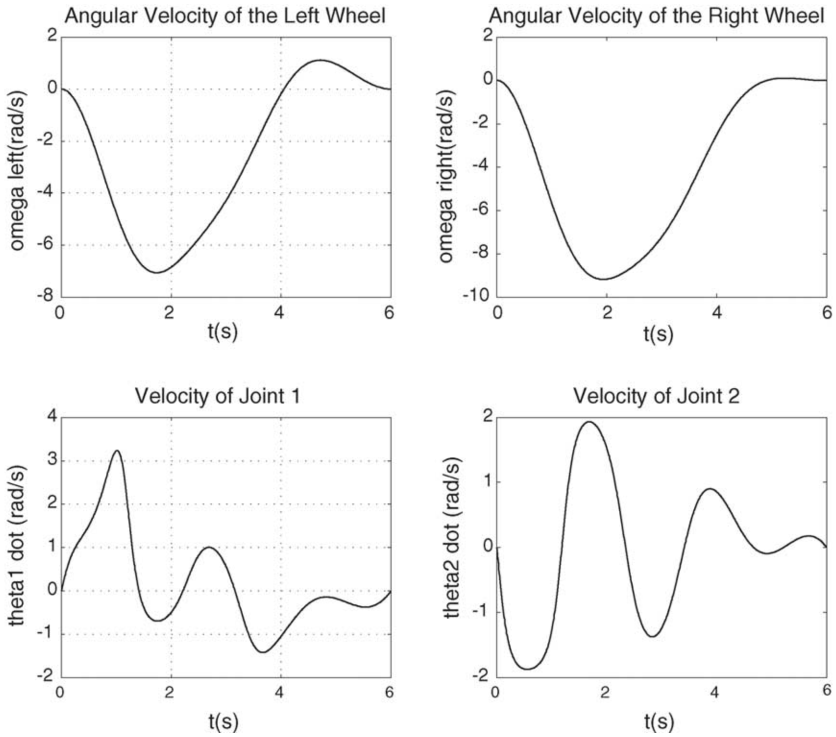

Fig. 22. Input velocities for the path shown in Fig. 21.

velocities. It is evident that the manipulator succeeds in following the specified trajectory. In addition, the platform moves without violating the nonholonomic constraint, without colliding with the obstacle and without driving the manipulator to its workspace limits. As is always the case with this methodology, wheel speeds shown in Fig. 22 are smooth, and therefore realizable.

A significant advantage of this method is that it requires very little additional time since it only has to calculate the roots of two polynomials for each of the two reach restrictions of the manipulator. It also provides the opportunity for multiple solutions by changing the value of b_4 or even by implementing one of the techniques proposed in Section 5. In general, the method provides a quick way of deriving admissible trajectories in complex cases like the crack-sealing problem illustrated here. Although in this paper a planar manipulator was assumed, in principle the method can be extended to spatial systems, taking into account the degree of conservatism that can be tolerated. However, this is an issue of current research.

7. Conclusions

In this paper, a methodology for planning the motion of nonholonomic mobile manipulators in the presence of obstacles and its application to various situations was presented in detail. The method uses smooth and continuous functions such as polynomials, and takes into account the details of the geometry of the integrated platform-manipulator system to return collision free paths or trajectories. The method was applied to a differentially driven platform equipped with a two-link manipulator; however, it can be easily extended to more complex mobile manipulators.

The concept of using appropriately shaped polynomial functions for obstacle avoidance in nonholonomic systems, initially proposed in [18], was further elaborated and extended here, and important implementation issues were discussed. The method was extended to include polygonal obstacles of any kind. The algebraic nature of the methodology is retained, and the time taken to compute valid paths increases linearly with the number of obstacles and sides considered. This paper also discusses ways to resolve implementation issues inherent in [18], which sometimes resulted in unacceptably long paths or in singularities depending on the boundary conditions and the location of the obstacles. Two efficient techniques have been exemplified to remedy these problems. In the first, intermediate points are employed, and in the second, the periodicity of the platform's orientation is exploited. Illustrative examples demonstrated the use of these techniques in various workspaces with obstacles enclosed in simple geometrical forms. Finally, the planning methodology was extended to include the case of obstacles moving in the workspace along known trajectories. This situation also arises when the manipulator end-point is required to follow a desired Cartesian trajectory, such as in the crack-sealing problem.

The techniques presented in this paper addressed initial limitations without increasing the computational effort. As a result the proposed methodology yields a fast planner, a significant advantage especially in cases where the system's computing resources are limited or the mission time is critical.

Acknowledgments

Support for this work by IRIS III, Canadian Centres of Excellence (LAW), is acknowledged by the first author. The work of I. Poulakakis has been supported by a R. Tomlinson Doctoral Fellowship Award and by the Greville Smith McGill Major Scholarship.

References

[1] J.-C. Latombe, Robot Motion Planning, vol. 15, Kluwer Academic Publishers, 1991.
[2] I. Kolmanovsky, H. McClamroch, Developments in nonholonomic control problems, IEEE Control Syst. (December 1995) 20–35.
[3] Z. Li, J. Canny, Nonholonomic Motion Planning, Kluwer Academic Publishers, 1992.
[4] J. Barraquand, J.-C. Latombe, Nonholonomic multibody mobile robots: controllability and motion planning in the presence of obstacles, in: Proceedings of the IEEE International Conference on Robotics and Automation, April, 1991, pp. 2328–2335.
[5] P. Jacobs, J. Canny, Planning smooth paths for mobile robots, in: Proceedings of the IEEE International Conference on Robotics and Automation, April, 1989, pp. 2–7.
[6] J.-P. Laumond, J. Jacobs, M. Taix, R.M. Murray, A Motion planner for nonholonomic mobile robots, IEEE Tr. Rob. Autom. 10 (5) (1994) 577–593.
[7] S. Sekhavat, J.-P. Laumond, Topological property for collision-free nonholonomic motion planning: the case of sinusoidal inputs for chained form systems, IEEE Tr. Rob. Autom. 14 (5) (1998) 671–680.
[8] A. Divelbiss, J. Wen, A path space approach to nonholonomic motion planning in the presence of obstacles, IEEE Tr. Rob. Autom. 13 (3) (1997) 443–451.
[9] M. Yamamoto, M. Iwamura, A. Mohri, Quasi-time-optimal motion planning of mobile platform in the presence of obstacles, in: Proceedings of the IEEE International Conference on Robotics and Automation, April, 1999, pp. 2958–2963.
[10] P. Ferbach, A method of progressive constraints for nonholonomic motion planning, IEEE Tr. Rob. Autom. 14 (1) (1995) 172–179.
[11] A. Bemporad, A. De Luca, G. Oriolo, Local incremental planning for a car-like robot navigating among obstacles, in: Proceedings of the IEEE International Conference on Robotics and Automation, April, 1996, pp. 1205–1211.
[12] E. Papadopoulos, Y. Gonthier, A framework for large-force task planning of mobile redundant manipulators, J. Rob. Syst. 16 (3) (1999) 151–162.
[13] H. Seraji, A unified approach to motion control of mobile manipulators, Int. J. Rob. Res. 17 (2) (1998) 107–118.
[14] Y. Yamamoto, X. Yun, Coordinated obstacle avoidance of a mobile manipulator, in: Proceedings of the IEEE International Conference on Robotics and Automation, Nagoya, Japan, May, 1995, pp. 2255–2260.
[15] P. Ogren, M. Egerstedt, X. Hu, Reactive mobile manipulation using dynamic trajectory tracking, in: Proceedings of the IEEE International Conference on Robotics and Automation, San Fransisco, CA, May, 2000, pp. 3473–3478.
[16] C. Perrier, P. Dauchez, F. Pierrot, A global approach for motion generation of nonholonomic mobile manipulators, in: Proceedings of the IEEE International Conference on Robotics and Automation, Leuven, Belgium, May, 1998, pp. 2971–2976.
[17] H.G. Tanner, S.G. Loizou, K.J. Kyriakopoulos, Nonholonomic navigation and control of cooperating mobile manipulators, IEEE Tr. Rob. Autom. 19 (1) (2003) 53–64.
[18] E. Papadopoulos, I. Poulakakis, Planning and obstacle avoidance for mobile robots, in: Proceedings of the IEEE International Conference on Robotics and Automation, Seoul, Korea, May, 2001.
[19] E. Papadopoulos, I. Poulakakis, I. Papadimitriou, On path planning and obstacle avoidance for nonholonomic mobile manipulators: a polynomial approach, Int. J. Rob. Res. 21 (4) (2002) 367–383.
[20] E. Papadopoulos, J. Poulakakis, Planning and model-based control for mobile manipulators, in: Proceedings of the International Conference on Intelligent Robots and Systems

(IROS '00), Kagawa University, Takamatsu, Japan, October 30–November 5, 2000.

Evangelos Papadopoulos received his diploma from the National Technical University of Athens (NTUA) in 1981, and his MS and PhD degrees from MIT in 1983 and 1991, respectively, all in mechanical engineering. He was an analyst with the Hellenic Navy from 1985 to 1987. In 1991, he joined McGill U. and the Centre for Intelligent Machines (CIM) as an assistant professor to be tenured in 1977. Currently, he is an associate professor with the Mechanical Engineering Department at the NTUA. He teaches courses in the areas of systems, controls, mechatronics and robotics. His research interests are in the area of robotics, modeling and control of dynamic systems, haptic devices, mechatronics and mobile robotics. He has published more than 100 articles in journals and conference proceedings. He is serving as an associate editor of the IEEE *Transactions on Robotics* and of the *Theory of Machines and Mechanisms*. Prof. Papadopoulos is a senior member of the IEEE and of the AIAA, and a member of the ASME, the Technical Chamber of Greece (TEE) and the Sigma Xi.

Iakovos Papadimitriou received his diploma in mechanical engineering from the National Technical University of Athens, Greece and his MS in mechanical engineering from the University of California, Berkeley. He has worked as a researcher at the National Technical University of Athens and at the California Partners for Advanced Transit and Highways (P.A.T.H.). His research interests include motion planning and control of autonomous vehicles, parametric and nonparametric modeling, as well as model-based control design. He is currently with Gamma Technologies Inc., specializing in software development for engine/powertrain control applications. He is a member of SAE and the recipient of the Best Student Paper Award at the 2003 American Control Conference.

Ioannis Poulakakis received a diploma in mechanical engineering and an MSc in robotics and control systems from the National Technical University of Athens, Greece. He obtained an M.Eng. in mechanical engineering from McGill University, where he was supported by a R. Tomlinson Doctoral Fellowship Award and the Greville Smith McGill Major Scholarship. He is currently working towards his PhD in the Department of Electrical Engineering and Computer Science at the University of Michigan, where he is a W. Benton Fellow. His research interests include applications of dynamical systems theory in robot locomotion with emphasis in dynamic legged robots, and also motion planning and control of systems exhibiting nonholonomic behavior. Mr. Poulakakis is a student member of IEEE and ASME, and a member of the Technical Chamber of Greece (TEE).