MOTION PLANNING AND CONTROL FOR NON-HOLONOMIC MOBILE ROBOTS

R. Zulli*, R. Fierro, G. Conte* and F.L. Lewis****

*Dipartimento di Elettronica ed Automatica, Università di Ancona
Via Brecce Bianche, 60131, Ancona, Italia
**Automation and Robotics Research Institute, The University of Texas at Arlington, 7300 Jack Newell Blvd. South, Forth Worth, TX 76118, USA

Abstract:
The design of autonomous robots relies heavily on the development of feasible and efficient motion planning and control algorithms. Such a design must consider the non-holonomic constraints which characterize the existing wheeled mobile robots. In this paper the integration of reliable procedures for motion planning and for control is performed by means of an appropriate interface, taking into account the geometric, kinematic and dynamic models of the mobile base. Simulation results show the effectiveness of the proposed approach.

1. Introduction

The construction of fast, efficient procedures for motion planning is a meaningful step towards the construction of autonomous robots. Motion planning procedures rely on the development of feasible control algorithms. The consideration of non-holonomic constraints, that is constraints on the velocity of the system that cannot be transformed into position constraints by means of integration, is an important topic for the construction of practical trajectories for mobile robots. Motion planning and control procedures must deal explicitly with non-holonomic constraints that characterize the existing mobile robots (see (Pin and Killough, 1994) for a recent attempt to construct wheeled holonomic platforms).

In the last years much work has been done on motion planning (see Latombe,1991; Hwang and Ahuja, 1992 for a comprehensive review of existing motion planning approaches). However, most of this work does not explicitly take into consideration non-holonomic constraints, with a few notable exceptions (see Li and Canny, 1993; Barraquand and Latombe, 1989). The consideration of non-holonomic constraints makes the motion planning problem considerably more difficult and it is not trivial even in absence of obstacles (see Dubins, 1957; Reeds and Shepp, 1990).

At the same time great effort has been devoted to nonlinear control design for wheeled mobile robots. Recent control literature (Canudas de Wit et.al., 1993) addresses three different navigation problems, known as trajectory tracking, path following and point stabilization. From the theoretical point of view the interest in the study of control properties of non-holonomic systems is derived from the difficulties related to the solution of these problems, due to the peculiar structure of the kinematic equations of a wheeled robot subject to a non-holonomic constraint. In fact a result due to (Brockett, 1983) states the non-existence of a stabilizing smooth state-feedback law. Three alternatives have been considered to address the point-stabilization problem, namely the use of smooth time-varying nonlinear feedbacks, the use of piecewise continuous control laws and the use of time-varying piecewise continuous control laws. However, independently from the adopted solution, control algorithms do not take into account the presence of obstacles in the environment and thus they must rely on a previously planned collision-free path.

In this paper a feasible motion planning procedure, developed taking into consideration the existing non-holonomic constraint, is integrated with a non-linear tracking controller that takes into account the complete dynamics of the mobile base. Both the procedures constitute the result of recent research work and will be described in detail in forthcoming papers. In particular, the motion

0-7803-2722-5/95 $4.00 © 1995 IEEE

planning procedure is based on the theoretical work of (Dubins, 1957; Reeds and Shepp, 1990). Successively the Vgraph methodology is extended in order to consider the non-holonomic constraint and a classical A^* algorithm is used for finding a collision-free path. The heuristic function used for the A^* search takes into account the non-holonomic constraint. On the other hand the control algorithm is based on a dynamical extension that makes possible the integration of a kinematic controller and a torque controller for non-holonomic mobile robots. Although the structure of the controller is quite general, in this case the algorithm has been applied only for tracking a desired trajectory. Furthermore an appropriate interface has been designed for linking the motion planning and the control design procedures. This interface transforms the purely geometric output of the motion planner, given as a sequence of arcs and segments, into a time-indexed sequence of points, assigning a desired linear and angular velocity to each arc or segment.

The paper is organized as follows. In Section 2 the overall design strategy is illustrated. The non-holonomic motion planning and control algorithms are described respectively in Section 3 and Section 4. In Section 5 some examples of the planned trajectories for a mobile base are given and discussed. Finally, in Section 6, conclusions are drawn and further developments are discussed.

2. General design

In Fig. 1 the overall structure of the design algorithm is illustrated. Three different models of the wheeled robot are used, namely the geometric, kinematic, and dynamic models. The integration of the motion planning and control algorithms leads to a design algorithm able to produce a collision-free trajectory for the mobile base from a given start position (x_s, y_s, θ_s) to a given goal position (x_g, y_g, θ_g) and the values of the non-linear feedback control needed to follow the prescribed trajectory.

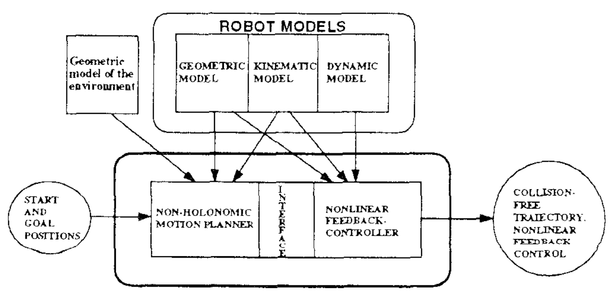

Fig. 1 - General structure of the design algorithm.

As evident in Fig. 1, only the motion planner knows the geometric model of the environment and it utilizes this knowledge to plan a collision-free path for the robot. On the other hand, only the controller knows the dynamic model of the robot and, based on this knowledge, it is able to find the appropriate commands for the wheel actuators for following the desired trajectory. Both the procedures need to consider the kinematic model of the robot, for dealing with the non-holonomic constraints. Furthermore, the motion planner needs to take into consideration the geometric model of the robot only with regard to its global size, while the control algorithm needs to know in a greater detail the geometric parameters characterizing the mobile base.

The mobile robot shown in Fig. 2 is a typical example of a non-holonomic mechanical system. It consists of a vehicle with two driving wheels mounted on the same axis, and a front free wheel. The motion and orientation are achieved by independent actuators, e.g., DC motors providing the necessary torques to the rear wheels.

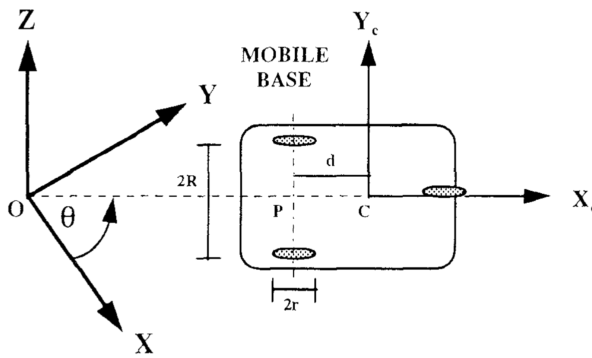

Fig. 2 - Geometric model of a non-holonomic platform.

It is known that linearization techniques fail at the intersection of the wheel axis with the axis of symmetry (point **P** in Fig. 2). A common solution to this problem is to define a new reference point on the axis of symmetry (point C in Fig. 2). The position of the robot in an inertial Cartesian frame {O,X,Y} is completely specified by the vector $q = [x_c\ y_c\ \theta]^T$ where x_c, y_c are the coordinates of the reference point **C**, and θ is the orientation of the frame {C, Xc, Yc} with respect to the inertial one.

The non-holonomic constraint states that the robot can only move in the direction normal to the axis of the driving wheels, i.e. the mobile base satisfies the conditions of pure rolling and non slipping (Yamamoto and Yun, 1993)

$$\dot{y}_c \cos\theta - \dot{x}_c \sin\theta - d\dot{\theta} = 0 . \qquad (2.1)$$

Using the above expressions is possible to derive the forward kinematics of the mobile base. Forward kinematics is used to estimate positions and velocities in Cartesian space from a set of joint variables. The kinematic equations of motion of C in terms of its linear velocity and angular velocity are

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & -d\cos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \qquad (2.2)$$

where v_1 and v_2 are the linear and angular velocities of the mobile robot. System (2.2) is called the kinematic model of the vehicle (or steering system).

The communication between the motion planner and the control algorithm is handled by an appropriate interface, illustrated in Fig. 3. Basically the interface transforms the output of the motion planner, given as a sequence of arcs and segments, into a time-indexed trajectory, considering the physical limits in the velocities achievable by the robots, i.e. considering that $|v_1| \leq v_{max}$ and $|v_2| \leq \omega_{max}$, where v_{max} and ω_{max} are the maximum linear and angular velocities of the mobile robot. The time-indexed trajectory obtained as output of the interface represents the trajectory of the reference cart used to define the tracking problem in Section 4.

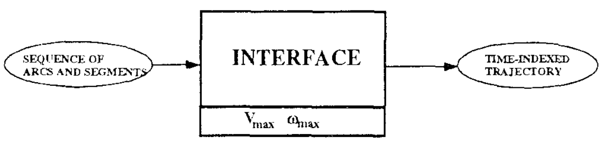

Fig. 3 - Functional view of the interface.

3. Non-holonomic motion planning

In this section the motion planning procedure will be briefly introduced and discussed. The adopted strategy for motion planning is based on the theoretical work of (Dubins, 1957) and on the successive extension made by (Reeds and Shepp, 1990) in order to consider reversals.

Simply stated, the motion planning problem consists in the calculation of a collision-free path, i.e. a sequence of collision-free positions, for a rigid object translating and rotating among static obstacles in a known environment. A widely used representation tool for stating precisely motion planning problems is the configuration space (Lozano-Perez, 1983). In the configuration space (Cspace) a robot is represented by a point and the configurations forbidden to the robot, due to the presence of obstacles in the workspace, form a region called the configuration space obstacle (CO). The complementary part of the CO with respect to the configuration space is called the free configuration space (C_{free}). In the case of wheeled mobile robots, generally the non-holonomic constraint (2.1) must be considered: this fact makes the motion planning problem considerably more difficult. A non-holonomic constraint is a constraint on the velocity of the system that cannot be transformed in a position constraint by means of integration, so it cannot be represented in the configuration space. It is represented in the tangent space of Cspace, that is the space of velocity directions at any given configuration. However, in the ideal case of a point robot moving in free-space and characterized by a bounded radius of curvature, it is possible to define a set of standard paths connecting two points in the plane with a given initial and final orientation (thus, two points in the configuration space $R^2 \times [0,2\pi[$ of the mobile robot).

A set of results due to Dubins are now recalled in a formulation suitable for our purposes. Consider a point robot pursuing a continuous differentiable path in the plane from an initial point $\mathbf{u}$ to a final point $\mathbf{v}$. Let $\mathbf{U}$ and $\mathbf{V}$ be respectively the velocity vectors of $\mathbf{u}$ and $\mathbf{v}$, and let $R>0$. Let $C=C(\mathbf{u},\mathbf{v},\mathbf{U},\mathbf{V},R)$ be the collection of all curves $\mathbf{X}$ defined on a closed interval $[0,L]$, where $L=L(\mathbf{X})$ and such that $\mathbf{X}(s) \in \mathrm{R}^2$, for $0 \leq s \leq L$. Using these definitions, it is possible to state the following Theorem:

Theorem 3.1 - If the following conditions hold:

1. $\mathbf{X}(0) = \mathbf{u}, \mathbf{X}(L) = \mathbf{v}, \dot{\mathbf{X}}(0) = \mathbf{U}, \dot{\mathbf{X}}(L) = \mathbf{V}$

2. $\left\|\dot{\mathbf{X}}(s)\right\| = 1$

3. $\left\|\dot{\mathbf{X}}(s_1) - \dot{\mathbf{X}}(s_2)\right\| \leq R^{-1}|s_1 - s_2|$

then, there exists a curve X belonging to C of minimal length. □

The curve X of minimal length will be called R-geodesic. Note that condition 3 of Theorem 3.1 is a condition on the average curvature of X in R^2.

Theorem 3.2 - Every planar R-geodesic is either:

(1) an arc of a circle of radius R, followed by a line segment, followed by an arc of circle of radius R;
(2) a sequence of three arcs of circles of radius R;
(3) a subpath of a path of type (1) or (2). □

In the sequel, the paths of the of type (1) will be called CSC (Circle,Segment,Circle) paths, while those of type (2) will be called CCC (Circle,Circle,Circle) paths. Moreover, only CSC paths will be considered, because usually it is not desired to make turns when it is not strictly necessary and, furthermore, CCC paths exist only in singular situations. In the original work of Dubins maneuvers are not allowed and it turns out that, in general, there exist four CSC paths, as illustrated in Fig. 4.

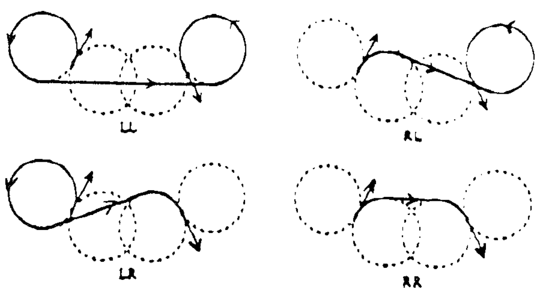

Fig. 4 - The four CSC paths for a robot that moves only forward: L and R stand for left and right.

Reeds and Shepp extended Dubins' results to the case including backward motions. They showed that in this case it is possible to define a small sufficient set of paths. This set is sufficient in the sense that it always contains the *R*-geodesic and small in the sense that these are at most 68 paths, usually considerably fewer for a given pair of points and directions. However, there are only eight CSC paths, represented in Fig. 5.

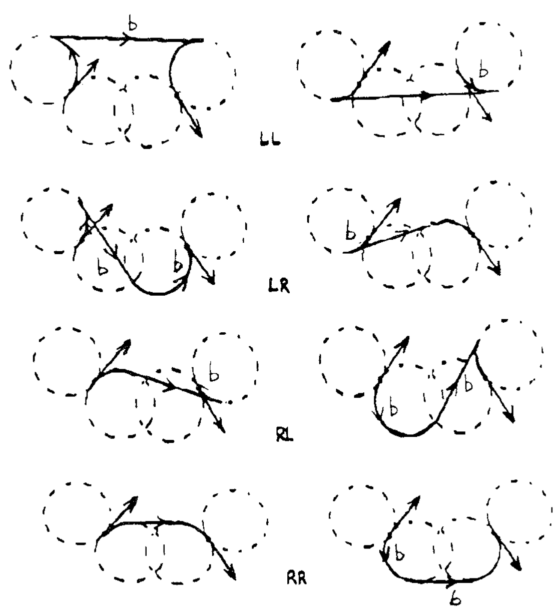

Fig. 5 - CSC paths for a robot that moves both forward and backward: the backward movements are marked with a "b".

The results just outlined apply only in free space. A new, efficient algorithm for planning collision-free paths in Cspace considering non-holonomic constraints has been developed basing on these results. In particular, the eight paths shown in Fig. 5 are considered as the set of standard paths connecting two points with given initial and final orientation. The main idea is to use the vertices of the obstacles as via-points for the calculation of a collision-free path. This idea is similar to the one used for constructing the Vgraph (Rohnert, 1986), the main difference being the use of a different concept for the visibility between vertices, based on the existence of a collision-free CSC path. The algorithm utilizes a two-dimensional representation, as in Fig. 6, where the obstacles have been appropriately enlarged for considering the dimension of the robot and the actual radius of curvature. In fact, having defined a standard set of paths in the plane, it is not necessary to construct explicitly the 3-d Cspace of the robot. At this point, an "extended" Vgraph is constructed. The nodes of the graph correspond to the vertices of the obstacles and to the start and goal positions, while each arc represents the connection between each pair of vertices. More precisely, it is possible to connect each pair of vertices of the graph with eight different CSC paths. Once that the Vgraph has been constructed, a classical A^* search algorithm (Nilsson, 1982) is used to find a feasible path using as heuristic function the distance between two vertices, evaluated as the length of the actual CSC path.

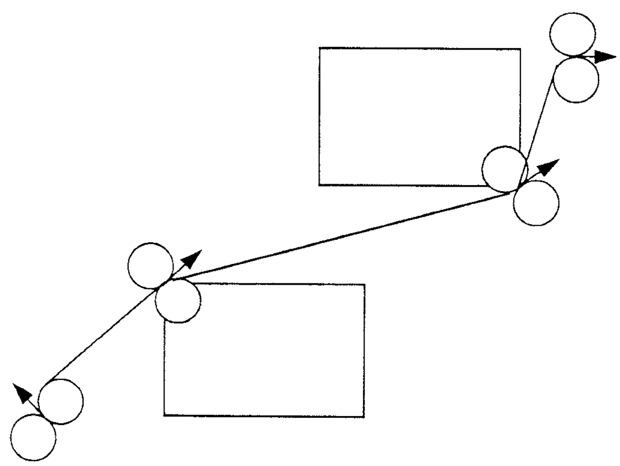

Fig. 6 - Basic idea of the algorithm: the circles are used as support for the definition of the CSC paths.

The principal feature of the algorithm, whose schematic diagram is reported in Fig. 7, is the use of an "extended" Vgraph. In fact, each arc represents a collision-free CSC path: thus, it is possible to think that each arc represents a

"CSC visibility" between two vertices, thus extending the classical visibility concept. Furthermore, the algorithm has been extended to consider the case of a variable radius of curvature and it has been successfully applied in several situations, including the car-parking problem. The motion planning procedure has been implemented in C and simulation have been performed using a PC equipped with a 80486dx/33Mhz processor. The performances are quite satisfactory. In fact, execution times are of the order of 1s on a number of examples, including realistic situations.

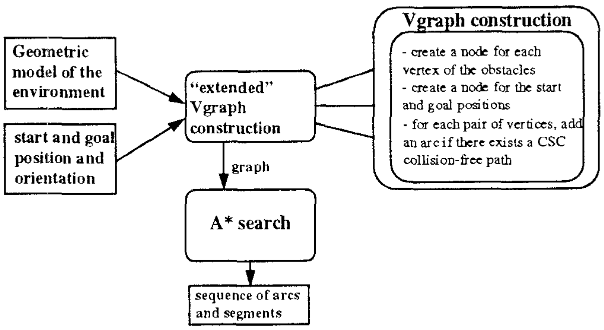

Fig. 7 - Diagram of the algorithm.

4. Control design

As stated in the Introduction, the non-holonomic control problem may be divided into three basic problems: reference-trajectory tracking, path following and point stabilization. In the sequel a stable control algorithm capable of dealing with the three basic navigation problems and that considers the complete dynamics of a non-holonomic mobile robot is briefly described. The complete description of the control procedure and the stability proofs are reported in (Fierro and Lewis, 1995).

A key point in developing intelligent systems is the reusability of the low level control algorithms and this is the case of the control structure reported in the following. In the sequel only the trajectory tracking behavior will be considered, but it is possible to generate a different stable behavior, for instance path following behavior, without changing the structure of the controller, simply by redefining the control velocity input.

The dynamic model of a mobile robot system having an n-dimensional configuration space with generalized coordinates $(q_1, q_2, \ldots, q_n)$ and subject to m constraints can be described by:

$$\mathbf{M}(q)\ddot{q} + \mathbf{V}_m(q,\dot{q})\dot{q} + \mathbf{F}(\dot{q}) + \mathbf{G}(q) = \mathbf{B}(q)\tau - \mathbf{A}^T(q)\lambda \quad (4.1)$$

where $\mathbf{M}(q) \in R^{n \times n}$ is a symmetric, positive definite inertia matrix, $\mathbf{V}_m(q,\dot{q}) \in R^{n \times n}$ is the centripetal and Coriolis matrix, $\mathbf{F}(\dot{q}) \in R^{n \times n}$ denotes the surface friction, $\mathbf{G}(q) \in R^n$ is the gravitational vector, $\mathbf{B}(q) \in R^{n \times r}$ is the input transformation matrix, $\tau \in R^r$ is the input vector, $\mathbf{A}(q) \in R^{m \times n}$ is the matrix associated with the constraints and $\lambda \in R^m$ is the vector of constraint forces.

The kinematic equalities constraints will be expressed as follows:

$$\mathbf{A}(q)\dot{q} = 0\,. \quad (4.2)$$

Let $\mathbf{S}(q)$ be a full rank matrix (n-m) formed by a set of smooth and linearly independent vector fields spanning the null space of $\mathbf{A}(q)$, i.e.

$$\mathbf{S}^T(q)\mathbf{A}^T(q) = 0\,. \quad (4.3)$$

According to (Campion et.al., 1991), it is possible to find an auxiliary vector time function $v(t) \in R^{n-m}$ for all t, such that

$$\dot{q} = \mathbf{S}(q)v(t)\,. \quad (4.4)$$

The system (4.1) is now transformed into a more appropriate representation for control purposes. Differentiating equation (4.4), substituting this result in (4.1) and then multiplying by $\mathbf{S}^T$, it is possible to eliminate the constraint matrix $\mathbf{A}^T(q)\lambda$. The complete equations of motion of the non-holonomic mobile platform are given by:

$$\dot{q} = \mathbf{S}v \quad (4.5)$$

$$\mathbf{S}^T\mathbf{M}\mathbf{S}\dot{v} + \mathbf{S}^T(\mathbf{M}\dot{\mathbf{S}} + \mathbf{V}_m\mathbf{S})v + \overline{F} + \overline{\tau}_d = \mathbf{S}^T\mathbf{B}\tau \quad (4.6)$$

where $v(t) \in R^{n-m}$ is a velocity vector. By appropriate definitions equation (4.6) can be rewritten as follows:

$$\overline{\mathbf{M}}(q)\dot{v} + \overline{\mathbf{V}}_m(q,\dot{q})v + \overline{F} + \overline{\tau}_d = \overline{\mathbf{B}}(q)\tau\,, \quad (4.7)$$

where $\overline{\mathbf{M}}(q) \in R^{r \times r}$ is a symmetric, positive definite inertia matrix, $\overline{\mathbf{V}}_m(q,\dot{q}) \in R^{r \times r}$ is the centripetal and Coriolis matrix, $\overline{\mathbf{B}}(q) \in R^{r \times r}$ is the input transformation matrix, $\tau \in R^r$ is the input vector. Equation (4.7) describes the behavior of the non-holonomic system in a new set of local coordinates, i.e. $\mathbf{S}(q)$ is a Jacobian matrix which transforms velocities v, expressed in mobile base coordinates to velocities $\dot{q}$, expressed in Cartesian coordinates. Let u be an auxiliary input, then by applying the following nonlinear feedback:

$$\tau = \overline{\mathbf{B}}^{-1}\left(\overline{\mathbf{M}}u + \overline{\mathbf{V}}_m v + \overline{F} + \overline{\tau}_d\right) \quad (4.8)$$

the dynamic control problem can be converted into the kinematic control problem:

$$\dot{q} = \mathbf{S}(q)v(t) \quad (4.9)$$

$$\dot{v} = u. \quad (4.10)$$

In performing the input transformation (4.8), it is assumed that all the dynamical quantities (e.g. $\overline{\mathbf{M}}(q)$, $\overline{F}(v)$, $\overline{\mathbf{V}}_m(q,\dot{q})$) of the vehicle are exactly known. It is straightforward to incorporate standard adaptive or robust control techniques if this is not the case (Lewis et.al.,1993). The velocity control $v(t)$ for the steering system (4.9) can be selected in several ways. In this section, a prescribed control $v(t)$ is converted into a torque control $\tau(t)$ for the actual physical cart. This allows the steering system commands $v(t)$ in the literature to be converted to torques $\tau(t)$ that take into account the parameters of the actual cart.

The tracking problem is stated as follows. Define $q_r = [x_r \quad y_r \quad \theta_r]^T$, $v = [v_r \quad \omega_r]^T$. Given a reference cart

$$\begin{cases} \dot{x}_r = v_r \cos\theta_r \\ \dot{y}_r = v_r \sin\theta_r \\ \dot{\theta}_r = \omega_r \end{cases} \quad (4.11)$$

with $v_r \neq 0, \forall t$, find a smooth velocity control input $v_C(t)$ such that $\lim_{t\to\infty}(q_r - q) = 0$. Then define an auxiliary feedback control law $u(t)$ such that $v \to v_C$ as $t \to \infty$. Finally, compute the torque $\tau(t)$ using (4.8).

A general structure for the tracking control system is presented in Fig. 8. In this figure complete knowledge of the dynamics of the cart is assumed, so that (4.8) is used to compute $\tau(t)$ given $u(t)$. To be specific, it is assumed that the solution to the steering system tracking problem in (Kananyama et.al.,1990), denoted as $v_c(t)$, is available and is given by:

$$v_c = \begin{bmatrix} v_r \cos e_3 + k_1 e_1 \\ \omega_r + k_2 v_r e_2 + k_3 v_r \sin e_3 \end{bmatrix}, \quad (4.12)$$

where

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \mathbf{T_e}(q_r - q), \mathbf{T}_e = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.13)$$

and $k_1, k_2, k_3 > 0$.

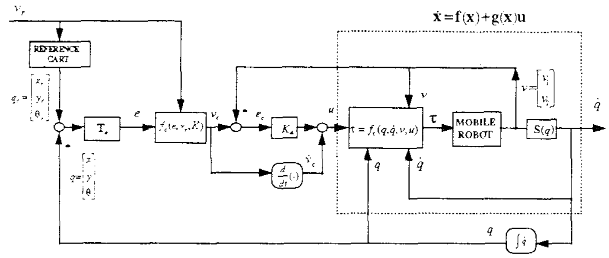

Fig. 8 - Tracking control structure.

Then the proposed nonlinear feedback acceleration control input is given by:

$$u = \dot{v}_c + \mathbf{K}_4(v_c - v) \quad (4.14)$$

where $\mathbf{K}_4$ is a positive definite, diagonal matrix. Thus, in conclusion, the overall controller is calculated as follows: first a smooth velocity control input is found by means of (4.12), successively (4.14) is used to calculate the nonlinear feedback acceleration control and finally the input transformation (4.8) is performed.

5. Simulation results

The motion planning and control design procedure developed here has been tested on several examples. In particular, the LABMATE mobile base has been considered in the simulations. In Fig. 9 an example of the results obtained is reported.

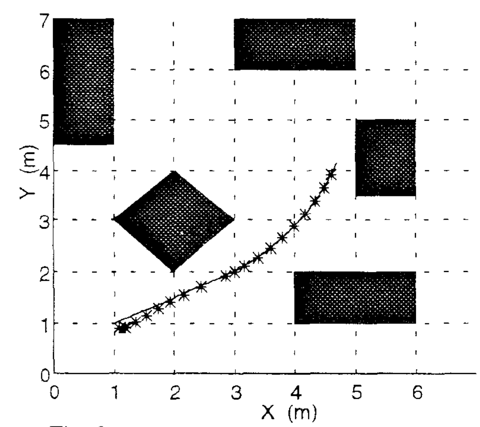

Fig. 9 - Desired (-) and actual (*) trajectories

The desired path consists of a segment and an arc, obtained as output of the motion planner described in Section 3. The output of the motion planner is then transformed in a reference trajectory by means of the interface described in Section 2. Clearly, the mobile platform is able to track the reference trajectory (Fig. 9), even in presence of discontinuities in the heading angle (shown in Fig. 10).

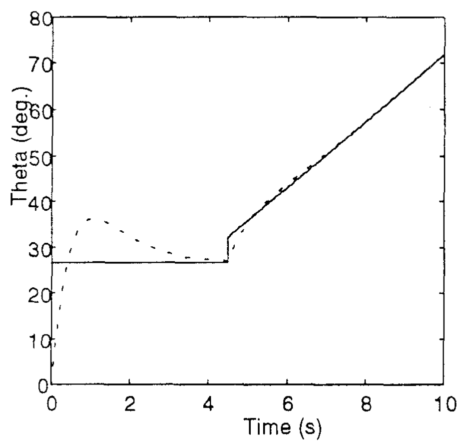

Fig. 10 - Reference (-) and desired heading Angle (..).

6. Conclusions

A complete motion planning and control procedure for mobile robots has been presented. First, a collision-free path is calculated using an efficient motion planner that takes into account the kinematics of the vehicle. The output of the motion planner is then transformed into a time-indexed data sequence which is fed into a backstepping controller, that is designed taking into account the full dynamics of the mobile base. The results of the simulations reported in Section 4 show the effectiveness of the proposed approach, which is also theoretically well founded. Further developments of this work include the testing of performances of path following and point stabilization control algorithms when used in conjunction with the motion planner described in section 3.

References

[1] Barraquand, J. and Latombe, J.C. (1989) Non-holonomic mobile robots and optimal maneuvering. *Revue d'Intelligence Artificielle*, Hermes, Paris, Vol. 3, No. 2, pp. 77-103.

[2] Brockett, R.W. (1983). Asymptotic stability and feedback stabilisation. *Differential geometric control theory*, Birkhauser, pp.181-208.

[3] Campion, G., d'Andréa-Novel, B. and Bastin, G. (1991) Controllability and state feedback stabilizability of nonholonomic mechanical systems, In C. Canudas de Wit, editor, *Lecture Notes in Control and Information Science*, pp. 106-124, Springer-Verlag.

[4] Canudas de Wit, C., Khennouf, H., Samson, C. and Sordalen, O.J. (1993) Nonlinear control design for mobile robots. In Y. F. Zheng, editor, *Recent Trends in Mobile Robots*, pp. 121-156, World Scientific.

[5] Dubins, L. E. (1957) On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal Mathematics*, Vol. 79, pp. 497-516.

[6] Fierro, R. and Lewis, F.L. (1995) Control of a nonholonomic mobile robot: backstepping kinematics into dynamics", *Submitted for publication.*

[7] Hwang, Y. K. and Ahuja, N. (1992) Gross motion planning - A survey. *ACM Computing Surveys*, Vol. 24, No.3.

[8] Kanayama, Y., Kimura, Y., Miyazaki, F. and Noguchi, T.(1990) A stable tracking control method for an autonomous mobile robot. *Proceedings IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 384-389.

[9] Latombe, J. (1991) *Robot motion planning*. Kluwer Academic Press.

[10] Lewis, F.L., Abdallah, C.T. and Dawson D.M. (1993) *Control of Robot Manipulators*, MacMillan, New York.

[11] Li, Z. and Canny, J.F. (1993) *Non-holonomic motion planning*. Kluwer Academic Press.

[12] Lozano-Perez, T. (1983) Spatial planning: a configuration space approach. *IEEE Transactions on Computers*, Vol. C-32, pp. 108-120.

[13] Nilsson, N.J. (1982) *Principles of artificial intelligence*. Springer-Verlag.

[14] Pin, F.G. and Killough, S.M. (1994) A new family of omnidirectional and holonomic wheeled platforms for mobile robots. *IEEE Transactions on Robotics and Automation*, Vol. 10, no.4, pp.480-489.

[15] Reeds, J. A. and Shepp L. A. (1990) Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, Vol. 145, No. 2, pp. 367-393.

[16] Rohnert, H. (1986) Shortest paths in the plane with convex polygonal obstacles *Information processing letters*, Vol. 23, pp. 71-76.

[17] Yamamoto, Y. and yun, X. (1993) Coordinating locomotion and manipulation of a mobile manipulator. In Y.F. Zheng, editor, Recent Trends in Mobile Robots, pp.157-181, World Scientific.