Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Motion Planning through Symbols and Lattices

S. Pancanti, L. Pallottino, D. Salvadorini, A. Bicchi

Abstract— **In this paper we propose a new approach to motion planning, based on the introduction of a lattice structure in the workspace of the robot, leading to efficient computations of plans for rather complex vehicles, and allowing for the implementation of optimization procedures in a rather straightforward way. The basic idea is the purposeful restriction of the set of possible input functions to the vehicle to a finite set of symbols, or *control quanta*, which, under suitable conditions, generate a regular lattice of reachable points. Once the lattice is generated and a convenient description computed, standard techniques in integer linear programming can be used to find a plan very efficiently. We also provide a correct and complete algorithm to the problem of finding an optimized plan (with respect e.g. to length minimization) consisting in a sequence of graph searches.**

I. Introduction

In this paper, the problem of steering complex systems (such as wheeled vehicles with an arbitrary number of trailers) among obstacles, is approached. The basic idea is to introduce in the robot's workspace a particular structure, consisting of a lattice, on which computations can be very efficient. This can be obtained in some cases by suitably discretizing the space of acceptable commands to the robot, thus reducing it to a finite set of *control quanta* associated to symbols of an alphabet, and describing robot motions through the generated language.

The use of symbolic languages to plan complex motions of large systems capable of complex behaviours, and to hierarchically abstract levels of decision, planning and supervision, is an approach that has been recently advocated [1], [2], [3], [4], [5], [6]. We present here a set of tools which, in certain cases, allow a very effective use of symbolic planning.

A lattice Λ is an additive group which can be generated by integer combinations of a finite number of linearly independent vectors. If the m generators h_i are rational n-dimensional vectors (which will always be the case for us), and are arranged as the columns of a matrix $H \in \mathbb{Q}^{n\times m}$, then the generated group is always a lattice, denoted as $\Lambda = \{H\lambda | \lambda \in \mathbf{Z}^m\}$.

The crucial observation from which our proposed method departs from is that, under suitable conditions, the set of reachable configurations of a mobile robot under sequences of control quanta, is a lattice. The planning problem is in this case reduced to solving the linear integer equation

$$y = H\lambda \tag{1}$$

where y represents the desired configuration (or its approximation on the lattice), and $\lambda \in \mathbf{Z}^m$ represents the number of times certain control words, i.e. sequences of control quanta, are to be used. This is a standard problem in linear integer programming, which can be solved very efficiently in polynomial time, by e.g. using the Hermite normal form of H, $H = [B\ 0]\ U$, where $B \in \mathbb{Q}^{n\times n}$ is a nonnegative, lower triangular, nonsingular matrix, and $U \in \mathbb{Q}^{m\times m}$ is unimodular (i.e., obtained from the identity matrix through elementary column operations).

Clearly, once the generating matrix H and its Hermite normal form have been computed (which can be done in polynomial time [7], [8], and off-line), all possible plans to reach any desired configuration y are obtained at once as

$$\lambda = U^{-1} \begin{bmatrix} B^{-1}y \\ \mu \end{bmatrix},\ \forall \mu \in \mathbf{Z}^{m-n}.$$

A lattice structure hence allows to solve different planning instances in free space in practically negligible time. It also proves very useful in planning amid obstacles, and in computing shortest paths, as it will be discussed in this paper.

With such motivations, questions are in order as to which systems can be planned on lattices, and by which means. Although a general answer to this question is not known at present, the theory of quantized control systems (QCS), a topic of recent research, can provide very useful results. It is known, in particular, that the reachable set of nonholonomic systems in chained form ([9]) with piecewise constant controls taking values in a discrete set, is a lattice ([10]). It is also true that, by suitably choosing the control set, the lattice mesh can be made arbitrarily fine.

In this paper we exploit these results and ideas to propose a planner for the N-trailer vehicle model, which is known to be feedback-equivalent to chained form ([11]).

A. Method outline

The basic steps of the proposed method can be summarized as follows (see fig. 1):

1) write the kinematics of the N-trailer system in the usual coordinates and with velocity inputs as $\dot{q} = T(q)v(t)$, $q \in \mathbf{R}^{3+N}$, $v(\cdot) : \mathbb{R}_+ \to \mathbb{R}^2$ (see (6));
2) use a continuous feedback $v(\cdot) = f(q(\cdot), u)$, and a coordinate change $x = \Phi(q)$, as specified in [11], to obtain an equivalent system $\dot{x} = C(x)u(t)$ in chained-form (see (2));
3) restrict the new input $u(t)$ to piecewise constant functions over a sampling time T, and compute the exact discrete–time model $x(k+1) = \bar{C}(x(k), u(k))$ (see (3));
4) choose a finite, symmetric set of input values $U = \{0, \pm u_1, \pm u_2, \ldots, \pm u_m\}$, and impose $u(k) \in U$;
5) compute the reachable lattice for this system. If the reachable lattice mesh is too rough, change or add elements in U, and recompute;
6) solve the (optimal) planning problem in terms of a finite-length sequence of discrete inputs;
7) apply the computed sequence as a piecewise constant input $u(t)$ to the chained form system, and extract the corresponding path $q(t)$ in the original coordinates,

Work partially supported by RECSYS European Project number IST-2001-37170, FIRB contract 2003-111.

Authors are with the Interdepartmental Research Center "E. Piaggio", University of Pisa, Via Diotisalvi 2, Pisa, Italy.

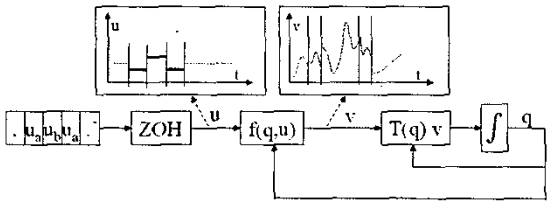

Fig. 1. Symbolic inputs are encoded by feedback into piecewise constant functins, which make the reachable set a lattice.

and the piecewise continuos inputs $v(t)$ that solve the steering problem.

II. Quantized Control Systems and Lattices

Some basic definitions and ideas on QCS and lattices that will be used in the rest of the paper are briefly reported here for reader convenience. Details can be found in [10].

Chained-form systems, introduced by Sastry and Murray in [9] as a canonical form for some continuous-time, driftless nonholonomic systems, are described by the ordinary differential equation

$$\begin{cases} \dot{x}_1 &= u_1, \\ \dot{x}_2 &= u_2, \\ \dot{x}_3 &= x_2 u_1, \\ \vdots &= \vdots \\ \dot{x}_n &= x_{n-1} u_1. \end{cases} \tag{2}$$

While several steering methods for chained-form systems have been provided in literature, optimal control for these systems is still an open problem (a characterization of extremals has been provided by [12]).

Consider the case where system inputs, rather than being allowed to change continuously in time, are bound to switch among a finite set of different levels at given switching times, which are multiples of a given time interval. Assuming such sampling interval to be of unit length, an exactly sampled model of chained-form systems can be easily obtained in discrete time from (2) by integration as

$$\begin{cases} x_1^+ &= x_1 + u_1, \\ x_2^+ &= x_2 + u_2, \\ x_3^+ &= x_3 + x_2 u_1 + \frac{1}{2} u_1 u_2, \\ \vdots &= \vdots \\ x_n^+ &= x_n + \sum_{j=1}^{n-2} x_{n-j} \frac{u_1^j}{j!} + u_1^{n-2} u_2 \frac{1}{(n-1)!}. \end{cases} \tag{3}$$

In this paper we assume that inputs $u = (u_1, u_2)$ can take values within a state-independent set of input symbols U, which is symmetric (i.e., if $u \in U$, then also $\bar{u} = -u \in U$). The set Ω of admissible control words (i.e. strings of admissible input symbols) is endowed with a composition law given by concatenation of strings. Because of the symmetry of U, every element $\omega \in \Omega$ has an inverse $\omega^{-1} \in \Omega$, simply defined as $(u_1 u_2 \cdots u_p)^{-1} = -u_p \cdots - u_2 - u_1, \pm u_i \in U, \forall i$.

In the state manifold of chained-form systems (2-3), it is customary to distinguish a *base* subsystem, consisting of the first two state variables $x_b = (x_1, x_2)^T$, and a *fiber* subsystem with coordinates $x_f = (x_3, \ldots, x_n)^T$. Observe that the restriction of chained-form systems to the base variables is linear, and indeed trivial to control. On the other hand, the difficulty in controlling fiber variables increases with the dimension of the state space. A typical example of such situation is in parking maneuvers of tractor-trailer systems, where base variables are associated with the steering tractor, and fiber variables correspond to the configurations of the trailers.

Accordingly, the reachability problem for discrete-time chained-form systems can be decoupled in the analysis of reachability of the base space, and of the fiber space associated with a reachable base point $(\overline{x}_1, \overline{x}_2)$. On the base space, system (3) has the simple form

$$x^+ = x + u, \; x \in \mathbb{R}^2, u \in U. \tag{4}$$

For such linear driftless systems, the analysis of the reachable set has been characterized in [10]. It can be observed that the reachable set R_x from a generic point x is obtained by translation of R_0. Therefore, if the control set U is quantized, symmetric and rational (as it almost always is in cases of interest, and as we assume in the rest of this paper), the reachable set is a lattice. In particular, let $U = \{0, \pm u_1, \ldots, \pm u_c\}$ and associate to U a matrix containing its positive vectors, $W = \begin{bmatrix} u_1 & \cdots & u_c \end{bmatrix}$. Then, the reachable set for (4) is the lattice generated by W. For instance, for a control set U with $c = 3$ and with

$$W = \begin{bmatrix} 0 & 2 & 2 \\ 2 & 0 & -4 \end{bmatrix}$$

the reachable lattice on the base is

$$\Lambda = \left\{ \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \lambda \,\middle|\, \lambda \in \mathbb{Z}^2 \right\}$$

(see fig. 2). Fixed a base point $(\overline{x}_1, \overline{x}_2)$ on the lattice, consider

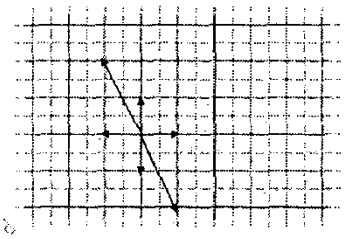

Fig. 2. Reachable lattice on the base space for an example in the text.

all possible closed paths on the base space. The control words ω that generate such closed paths form a subgroup $\tilde{\Omega} \subset \Omega$.

Upon application of a control word ω to system (3), the system moves in general from a state x to a new state $x^+(x, \omega) = x + \Delta(\omega) + \Gamma(\omega, x)$. A most useful result of [10] is that $\Gamma(\tilde{\omega}, x) = 0$, $\forall \tilde{\omega} \in \tilde{\Omega}$. The action of such control words in $\tilde{\Omega}$ on the fiber subsystem can therefore be described by [10]

$$z^+ = z + v, \quad z = (x_3, x_4, \cdots, x_n) \in \mathbb{R}^{n-2}, v \in \tilde{U} \tag{5}$$

where $\tilde{U} = \{\Delta^f(\omega), \omega \in \tilde{\Omega}\}$ and where $\Delta^f(\omega)$ denotes the $(n-2)$-dimensional projection of Δ on the fiber space. Clearly, $\tilde{U}$ is itself symmetric: indeed if $\omega \in \tilde{\Omega}$ then also $\omega^{-1} \in \tilde{\Omega}$ and $\Delta^f(\omega^{-1}) = -\Delta^f(\omega)$. The action of the subgroup $\tilde{\Omega}$ on the

fiber is thus additive and commutative, and the structure of the reachable set in the fiber is the same over every (reachable) base point.

To show that the reachable set on the fiber is itself a lattice, we need to show that the set obtained by integer combinations of all the infinite possible elements in $\tilde{U}$, can also be generated by an integer combination of only a finite number of rational vectors. This indeed holds true, as discussed in [13], where an algorithm to compute a finite set of generators is provided.

In conclusion, we have an algorithm to compute a generator matrix $H \in \mathbb{Q}^{n\times m}$ and a corresponding set of generating cyclic control words $\tilde{\Omega}^H = \{\tilde{\omega}_1, \ldots \tilde{\omega}_m\}$, such that any reachable point for the chained–form system can be written as

$$x = \left[\frac{x_b}{x_f}\right] = \left[\frac{W\Sigma^W}{H\Sigma^H}\right]$$

with arbitrary integer vectors Σ^W, Σ^H. Controls that take the system to x_b are concatenations of symbols in U (in any order) such that the symbol u_i is used Σ_i^W-times (counting the use of $-u_i$ as -1). Controls that subsequently take the system to x_f are concatenations of words in $\tilde{\Omega}^H$ (in any order) such that the word $\tilde{\omega}_i$ is used Σ_i^H-times (counting the use of $(\tilde{\omega}_i)^{-1}$ as -1).

III. Steering an N-trailer vehicle on the lattice

As mentioned in the introduction, among the nonlinear systems which can be converted in chained-form (2), wheeled vehicles represent a particularly interesting class. The kinematic model of a tractor with N trailers is given by

$$\begin{cases} \dot{x} &= \cos\theta_N v_N \\ \dot{y} &= \sin\theta_N v_N \\ \dot{\theta}_N &= \frac{1}{d_N}\sin(\theta_{N-1}-\theta_N)v_{N-1} \\ &\vdots \\ \dot{\theta}_i &= \frac{1}{d_i}\sin(\theta_{i-1}-\theta_i)v_{i-1} \quad i=1,\ldots,N \\ &\vdots \\ \dot{\theta}_1 &= \frac{1}{d_1}\sin(\theta_0-\theta_1)v_0 \\ \dot{\theta}_0 &= \omega \end{cases} \tag{6}$$

where: (x, y) is the absolute position of the center of the axle between the two wheels of the rear-most trailer; θ_i is the orientation angle of trailer i with respect to the x-axis, with $i \in \{1, \ldots, N\}$; θ_0 is the orientation angle of the tractor axle with respect to the x-axis, and d_i is the distance from the center of trailer i to the center of trailer $i-1$, $i \in \{2, \ldots, N\}$; d_1 is the the distance from the wheels of trailer 1 to the wheels of the tractor. The two inputs of the systems are v_0 and ω, the tangential velocity of the car and the angular velocity of the tractor respectively. The tangential velocity of a trailer i, v_i, is given by

$$v_i = \cos(\theta_{i-1}-\theta_i)v_{i-1} = \prod_{j=1}^{i}\cos(\theta_{j-1}-\theta_j)v_0,$$

where $i \in \{1, \ldots, N\}$. Incidentally, this model is identical to the model of a four-wheeled car pulling $N-1$ trailers, provided $\theta_0 - \theta_1$ denotes the angle of the front wheels relative to the orientation θ_1 of the rear axle of the four-wheeled car.

As described in the method outline, we proceed by using the feedback and coordinate transform suggested by Sørdalen in [11] so that (6) is put in the chained-form (2), and use piecewise constant control to obtain a discrete-time model as in (3). We then take a finite control set, and compute the generator matrices W and H. Solve first $W\Sigma^W = x_b^{goal} - x_b^{start}$, thus finding a control sequence bringing the system to the desired configuration in the base space.

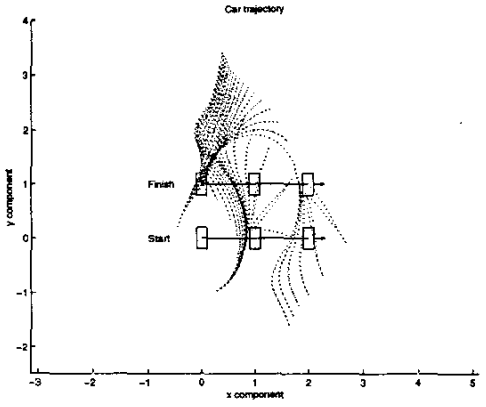

Fig. 3. Steering solutions a 2-trailer (108 steps) without obstacles can be computed very rapidly but may result in complex maneuvering.

Without loss of generality, up to rescaling the fiber state space, we take H to be an integer matrix. We then solve

$$H\Sigma^H = x_f^{goal} - x_f^{start}. \tag{7}$$

Notice that an integer solutions $\Sigma^H \in \mathbb{Z}^m$ of this equation exists if an only if the initial and goal points differ by a vector belonging to the fiber lattice. In other cases, integer truncations of a real solution x will provide approximated steering to the goal, within a tolerance dictated by the lattice mesh. As discussed previously, such solution can be obtained very rapidly once the matrix H of generators has been computed and pre-processed. This can be done in polynomial time. Once a solution of the steering problem (7) has been obtained, the continuous time trajectory of system (6) is computed through integrating the feedback system equations by numeric simulation (see fig. 3).

When the vehicle environment contains polyhedral obstacles O_i described by

$$O_i = \{x \in \mathbb{R}^n | A_i x \leq b_i\},$$

the planner can be simply modified by introducing a *collision test* function $\mathcal{CT} : \mathcal{X} \mapsto \{0, 1\}$, suitably taking into account security margins. Given a solution Σ^H to the unconstrained problem, there exist M possible different paths of minimal length $L = \sum_{i=1}^m \Sigma_i^H$, with

$$M = \frac{(\sum_{i=1}^m |\Sigma_i^H|)!}{\prod_{i=1}^m |\Sigma_i^H|!}.$$

This set of possible paths can be organized in a tree, and efficiently searched for a collision-free trajectory. An example of steering for a car-like vehicle with an obstacle is reported in fig. 4 This algorithm is not complete, however, as there are possible solutions to the planning problem that have not minimal length, which are outside the M count. Furthermore, there are infinitely many possible homogeneous solutions $H\Sigma_0^H = 0$, which provide new possible paths. This drawback

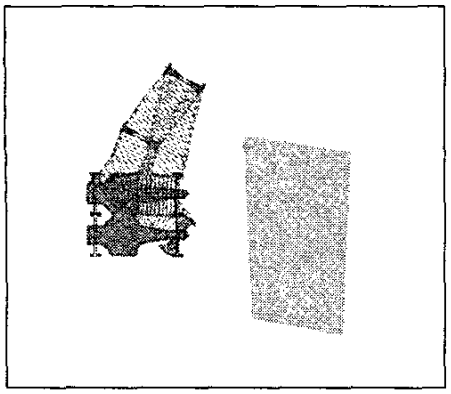

Fig. 4. A car-like system parking in parallel with a polyhedral obstacle. The non-optimized solution requires a control word of length 12.

will be solved by the optimizing algorithm to be illustrated in the following section.

IV. Optimal Steering on Lattices

Optimal steering strategies among solutions of (7) will be considered introducing a cost p_i associated to the control symbol $u_i \in U$. The corresponding cost for a word $\omega = (u_1, u_2, \cdots, u_p), u_i \in U$ is defined as $C(\omega) = \|P x\|$, where x_i stands for the number of appearances of the symbol u_i in ω (with negative sign if $-u_i$ appears), and $P = \mathsf{diag}(p_i)$.

A constrained minimization problem can be considered at this point, i.e.

$$\begin{array}{ll} \min_{\Sigma^H} & \|P\Sigma^H\| \\ \text{s. t.} & \begin{cases} H\Sigma^H = x_f^{goal} - x_f^{start} \\ \Sigma^H \in \mathbf{Z}^m \end{cases} \end{array} \tag{8}$$

leading to a linear integer program if a one-norm is considered, while using a two-norm would result in an integer quadratic program. Efficient algorithms do exist for both these problems: however, unfortunately, such formulation does not reflect the reality of our optimal control problem.

Indeed, in combining control words by concatenation cancellations of symbols may occur. To obtain the sum of two control actions $\Delta(\tilde{\omega}_i), \Delta(\tilde{\omega}_j)$ on the fiber, corresponding to control words $\tilde{\omega}_i, \tilde{\omega}_j$ whose costs are $C_i = C(\tilde{\omega}_i)$ and $C_j = C(\tilde{\omega}_j)$, respectively, the sum $C_i + C_j$ is only an upper bound to the actual cost of the corresponding control. Indeed, cancellations of one or more trailing symbols in $\tilde{\omega}_i$ with an equal number of symbols leading in $\tilde{\omega}_j$ is possible. We will denote by $\hat{C}(\tilde{\omega}_i, \tilde{\omega}_j)$ the actual cost of the word pair $(\tilde{\omega}_i, \tilde{\omega}_j)$.

For example, if $\tilde{\omega}_i = u_1 u_2 u_3 u_4$ and $\tilde{\omega}_j = -u_4 u_5 - u_2 - u_1$, in a minimum time problem we have $C_i = 4$ and $C_j = 4$. However, the concatenation of $\tilde{\omega}_i$ with $\tilde{\omega}_j$ leads, by cancellations, to the control word $u_1 u_2 u_3 u_5 - u_2 - u_1$, so that $\hat{C}(\tilde{\omega}_i, \tilde{\omega}_j) = 6 < 8$. Obviously, cancellations are crucial in minimizing unnecessary maneuvers in the steering problem, and motivate the following reformulation of the optimal control problem.

Consider an oriented graph $G_0 = (N_0, A_0)$ with a set N_0 of $l+2$ nodes, l of which are associated with the contributions $\Delta(\omega_i)$ on the fiber given by generators $b_i = \tilde{\omega}_i$, and where a start node S and a goal node F are additionally considered. In the arc set A_0 of G_0, all arcs connecting the start and goal nodes S, F with all other nodes are included, i.e. $(S, i) \in A_0, i = 1, \ldots, l$ and $(i, F) \in A_0, i = 1, \ldots, l$. An arc (i, j) is included in A_0 only if $\tilde{\omega}_i$ and $\tilde{\omega}_j$ are not the inverse of each other. In particular, for every node $i \neq S, F$, the arc (i, i) is included in A_0.

To the arc $(i, j) \in A_0$ we associate the cost $\hat{C}_{i,j} = \hat{C}(\tilde{\omega}_i, \tilde{\omega}_j) - C(\tilde{\omega}_i) \geq 0$ of the control sequence $(\tilde{\omega}_i, \tilde{\omega}_j)$ (so that the cost of path (S, i), (i, j) on the graph is $C(\tilde{\omega}_i, \tilde{\omega}_j)$), taking into account all possible cancellations. Notice that in general $\hat{C}_{i,j}$ is not equal to $\hat{C}_{j,i}$ (in the example above, for instance, $\hat{C}_{i,j} = 6 - 4 = 2$ while $\hat{C}_{j,i} = 4 - 4 = 0$). In figure 5 an example of graph is represented. A refinement

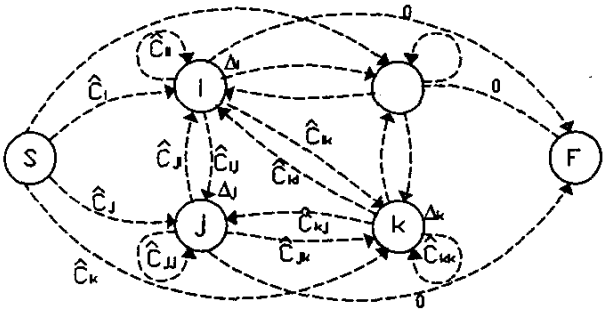

Fig. 5. Graph associated with generators of the fiber displacements.

step is necessary to finalize the graph construction for pairs $(\tilde{\omega}_i, \tilde{\omega}_j)$ where the number of cancellations is larger than the half-length of the shortest of the two words. Indeed, in this case it may happen that the cost of a triplet $(\tilde{\omega}_i, \tilde{\omega}_j, \tilde{\omega}_k)$ is underestimated by $\hat{C}_{ij} + \hat{C}_{jk}$. For example, if $\tilde{\omega}_i = u\,v - u\,v\,u$, $\tilde{\omega}_j = -u - v\,u\,v$ and $\tilde{\omega}_k = -v - u - u$, we have $\tilde{\omega}_i \tilde{\omega}_j = u\,v\,v$ ($\hat{C}_{ij} = 3 - 5 = -2$) and $\tilde{\omega}_j \tilde{\omega}_k = -u - v - u$ ($\hat{C}_{jk} = 3 - 4 = -1$) while the triplet is $\omega_i \omega_j \omega_k = u\,v - u - u$ whose cost is $4 - 5 = -1$ whereas on the graph the path (S, i), (i, j), (j, k) would cost $5 - 2 - 1 = 2$. To avoid this problem, we remove in the graph the arc (i, j) corresponding to such pairs, and add a new node associated to $\Delta(\tilde{\omega}_i) + \Delta(\tilde{\omega}_j)$ with cost $\hat{C}_{ij}$. These new nodes are connected to all other nodes by arcs whose cost is evaluated as usual, with the exception of arcs corresponding again to cancellations of more than the half-length of either words, which are not considered in the new graph.

On the graph G_0, all possible combinations of the generating control words $\tilde{\omega}_i$ are represented by connected paths from S to F. The optimal control problem on the fiber space can hence be formulated as follows:

Given the oriented graph G_0, determine the minimum-cost path from S to F with the constraint that the sum of all Δ_i of visited nodes equals the desired fiber displacement $x_f^{goal} - x_f^{start}$.

Thus, the optimal control problem can be regarded as a minimum-cost path search on a graph, with a constraint on the sum of "tokens" collected at each visited node. Notice that G_0 contains cyclic arcs of type (i, i), allowing to collect an arbitrary integer number of the corresponding token $\Delta(\tilde{\omega}_i)$. The search problem is a $\mathcal{NP}$-complete linear integer programming problem ([7],[8]), and differs substantially from standard shortest path searches on a graph because of the constraint and of the presence of cycles (cyclic paths are obviously never

considered in unconstrained path searches). The following section proposes a correct and complete algorithm to solve this optimal control problem.

V. A SOLUTION ALGORITHM

The non-standard nature of the optimization problem described above is such that even rather general solution techniques, as e.g. branch and bound, and commercial software tools for integer programming, cannot be used directly to solve the problem. We propose a procedure for the solution of this problem which basically consists of solving a sequence of problems of increasing complexity.

Consider first that an upper limit U on the optimal control cost can be easily obtained by evaluating the cost U_0 of *any* solution of the integer linear system (7) – for instance, a solution to problem (8), in the following will be referred to as *starting solution*.

At the first stage of the proposed algorithm, a new graph $G_1 = (N_1, A_1)$ is built by setting $N_1 = N_0$ and by removing all cyclic arcs from A_0, namely $A_1 = A_0 \setminus \{(i,i), \forall i\}$. Let now formalize the optimization problem obtained with the formulation given in previous section. Consider the incidence matrix $E \in \mathbb{R}^{s \times t}$ associated with the graph G_1: given an order to the elements of set A_1 (cardinality t) and of set N_1 (cardinality s), the element $E_{ij} = -1$ if the $i-th$ node is the first node of arc j, $E_{ij} = 1$ if the $i-th$ node is the second node of arc j, $E_{ij} = 0$ otherwise. Let $x \in \mathbb{R}^t$ be the vector variables taking values in $\{0, 1\}^t$ and representing the ordered arcs of the graph. Let $q \in \mathbb{R}^s$ such that $q_S = -1$, $q_F = 1$ and $q_i = 0$ for $i \neq S, F$. Finally, let $C^T \in \mathbb{R}^t$ be the vector in which the cost of the arcs are reported, the optimization problem is then

$$\min \quad Cx \qquad s.t. \quad \begin{cases} Ex = q \\ \tilde{H}x = d \\ x \in \{0, 1\}^t \end{cases} \tag{9}$$

where the set of constraints $\tilde{H}x = d$ (in the following will be referred to as set of *token constraints*) represents the constraints given in (7) where $\tilde{H} \in \mathbb{R}^{n-2 \times t}$ and the column $\tilde{H}_j$ is associated to the arc $j = (i,k)$ and represent the "token" payed at node k that is $\Delta(b_k)$ (where b_k is the generator associated with node k). The vector D represent the total displacement we intend to achieve on the fiber.

A branch-and-bound algorithm is applied to search minimum cost, token-constrained paths on G_1. Within such branch-and-bound subprocedure, the token constraint is relaxed, hence a number of classical minimum cost path search problems are obtained (solvable by the Dijkstra algorithm [14]) in each of which an arc is forced to be ($x_i = 1$) or not ($x_i = 0$) in the optimal solution. If the forced condition $x_i = 1$ or $x_i = 0$ brings to a shortest path of cost larger that U then the relative branch is cut and not further explored. Otherwise, another arc is forced to be or not in the optimal solution. If all branch are cut then no solution with cost less than U has been found. Otherwise, an optimal solution is found with cost $U_1 < U$. This solution is the shortest path from node S to F but in order to be an admissible solution of problem (9) it has to verify the token constraint. In this case the upper bound U on the optimal cost is updated, $U = U_1$.

At the $i+1$–th step of the algorithm, a graph $G_{i+1} = (N_{i+1}, A_{i+1})$ is built such that $N_{i+1} = N_i + N_0 \setminus \{S, F\}$, and A_{i+1} contains all connecting arcs between different nodes in N_{i+1} (without cyclic arcs). In other words, each node j with a cycle arc is split into two nodes (see figure 6) so that at step i, path with i cycles can be considered. A branch-and-bound algorithm is used again to find the constrained minimum cost U_{i+1}, and the upper bound is updated if $U_{i+1} < U$ and if the solution verifies the token constraint.

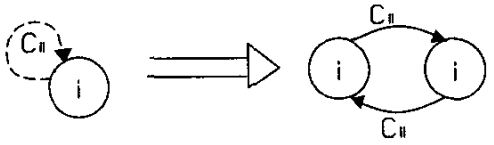

Fig. 6. The node i with a cycle arc is split into two nodes and two arcs.

A stopping condition for the procedure can be provided as follows. A lower bound on the optimal control cost solution L is initially set equal to the cheapest cost $L_0 = C_i$ of arcs of type (S,i) in the G_1 graph, since the cost of arc (i,F) is zero. At each step, the lower bound is updated as $L = L_{i+1} = L_i + \hat{C}_c$, where $\hat{C}_c$ denotes the minimum cost of a closed cycle in the graph G_1. The value of $\hat{C}_c$ is determined once and for all at the beginning of the procedure, by solving a standard (unconstrained) minimum-cost path problem on G_1.

The overall procedure is stopped whenever $L \geq U$.

Theorem 1: The solution algorithm is correct and complete.

Proof: Because initial and goal configurations are assumed to belong to the lattice, the optimum exists. Also, because the action on the fiber of the whole group $\tilde{\Omega}$ of control inputs that correspond to the desired final value of the base variables, is generated by the finite set of generators $\Delta(\tilde{\omega}_i), i = 1, \ldots, m$, and this set is (implicitly, but completely) searched by the branch-and-bound algorithm at successive stages of the algorithm, the algorithm is correct. On the other hand, the two sequences $\{L_i\}_{i \geq 0}$ and $\{U_i\}_{i \geq 0}$ are strictly uniformly increasing and non-increasing, respectively, and at any stage it holds $L_i \leq U_i$. Hence the algorithm stops in a finite number of stages, all of which consist of an implicit search on a finite graph, i.e. of a finite number of operations. ∎

The proposed algorithm has exponentially increasing complexity with the number of generators, as it uses a number of instances of a branch and bound procedure: this is hardly a surprise, as we are after all dealing with a nontrivial optimal control problem. However, performance can be improved by providing good initial estimates of the upper bound U_0. Some preprocessing of generators to facilitate the algorithm convergence can also help, and work is currently ongoing in this direction. The next section will provide some numerical examples of application of the proposed algorithm.

The problem of optimally steering N-trailers in an environment with polyhedral obstacles can be approached directly by essentially the same algorithm. At each step of the algorithm, the current candidate solution is tested for collisions through CT. If collisions are detected, the solution is discarded, and the current best solution is not updated. Notice that the collision test does not modify neither the structure of the algorithm, nor its completeness and correctness properties.

In figure 8, an optimal collision free trajectory is reported for the problem considered already in fig. 4. The maneuver length is reduced from 12 to 8 steps.

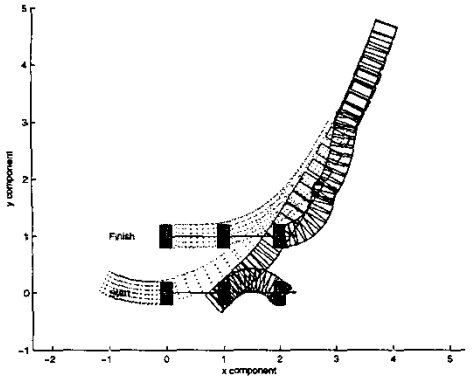

Fig. 7. Optimal steering solutions for a 2-trailer (12 steps) without obstacles.

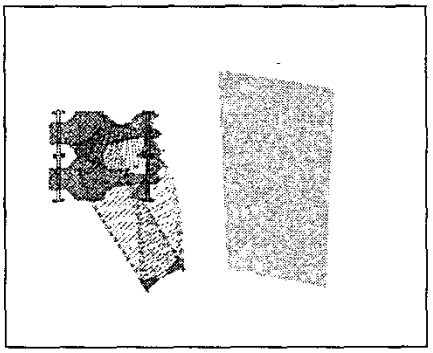

Fig. 8. Optimal solution for the problem in fig. 4, achieving the desired configuration in 8 steps.

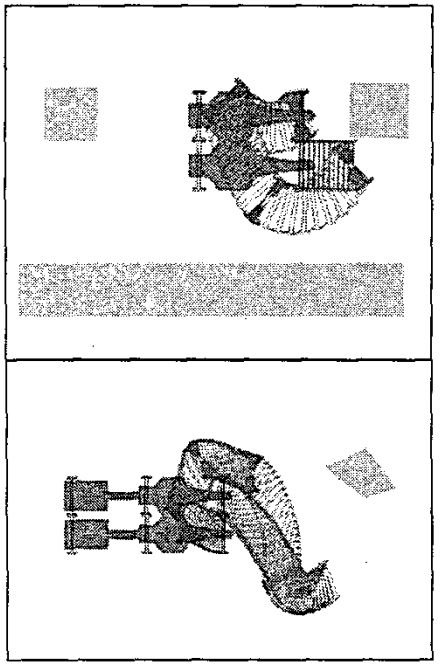

Fig. 9. Two further examples of optimized motion planning among obstacles for a parking problem of a car-like vehicle (top) and for a 2-trailer system (bottom).

VI. Conclusions

In this paper, a new approach to steering complex nonlinear systems has been proposed, which exploits the particular structure that input quantization can impose on the reachable set of some classes of systems. In particular, we have applied it to the steering problem with obstacles for wheeled vehicles with trailers.

The approach introduces some new ideas in the domain of motion planning, which may have further applications and uses. Several questions are open, as to e.g. what systems admit an input encoding that achieves a lattice structure in the workspace. A question of theoretical interest is whether the methopd provides a solution that converges to the optimum under continuous controls, when the number of input symbols and the resolution of the time discretization are increased. Furthermore, while we have only focused on open–loop optimal steering, it is possible to conceive applications where the lattice structure is conveniently used to simplify the computation of closed-loop dynamic programming problems, so as to find navigation functions for nontrivial systems.

References

[1] R. Brockett, "On the computer control of movement," in *Proc. IEEE Conf. on Robotics and Automation*, April 1988, pp. 534–540.

[2] M. Egerstedt, "Motion description languages for multimodal control in robotics," in *Control Problems in Robotics*, ser. STAR, A. Bicchi, H. Christensen, and D. Prattichizzo, Eds. Springer-Verlag, 2003, no. 4, pp. 75–89.

[3] V. Manikonda, P. S. Krishnaprasad, and J. Hendler, "A motion description language and hybrid architecture for motion planning with nonholonomic robots," in *Proc. Int. Conf. on Robotics and Automation*, 1995.

[4] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," in *Proc. AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 2000, pp. 1–11.

[5] M. Okada and Y. Nakamura, "Polynomial design of dynamics-based infomration processing system," in *Control Problems in Robotics*, ser. STAR, A. Bicchi, H. Christensen, and D. Prattichizzo, Eds. Springer-Verlag, 2003, no. 4, pp. 91–104.

[6] A. Marigo and A. Bicchi, "Steering driftless nonholonomic systems by control quanta," in *Proc. IEEE Int. Conf. on Decision and Control*, 1998.

[7] A. Schrijver, *Theory of Linear and Integer Programming*. Wiley Interscience Publ., 1986.

[8] L. A. Wolsey, *Integer Programming*. Wiley Interscience Publ., 1998.

[9] S. S. S. R. M. Murray, "Nonholonomic motion planning: Steering using sinusoids," *IEEE Trans. on Automatic Control*, vol. 38, pp. 700–716, 1993.

[10] A. Bicchi, A. Marigo, and B. Piccoli, "On the reachability of quantized control systems," *IEEE Trans. on Automatic Control*, vol. 47, no. 4, pp. 546–563, April 2002.

[11] O. Sordalen, "Conversion of the kinematics of a car with n trailers into a chained form," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1993, pp. 382–387.

[12] A. Sarichev and H. Nijmeier, "Extremal controls for chained systems," *Journal of Dynamical Control Systems*, vol. 2, pp. 503–527, 1996.

[13] S. Pancanti, L. Leonardi, L. Pallottino, and A. Bicchi, "Optimal control of quantized input systems," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, C. Tomlin and M. Greenstreet, Eds. Heidelberg, Germany: Springer-Verlag, 2002, vol. LNCS 2289, pp. 351–363.

[14] E. V. Denardo, *Dynamic Programming: Models and Applications*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey,, 1982.