

A Nonlinear Least Squares Approach to the Numerical Optimal Control of Non-holonomic Systems

Sudhaker Samuel and S.Sathiya Keerthi
Department of Computer Science and Automation
Indian Institute of Science, Bangalore 560 012, India

Abstract This paper gives a nonlinear least squares approach for numerically finding a trajectory to transfer a non-holonomic system from one configuration to another while satisfying given point-wise configuration constraints. A car like robot is considered as an example. The car like robot is kinematically constrained and is modelled as a 2 D object translating and rotating in the horizontal plane. The configuration constraints correspond to obstacle-avoidance. A simple technique is used to set up an approximate nonlinear least squares problem that includes the various constraints. An elegant error analysis is devised to effectively deal with the approximations. Several illustrative examples are presented to demonstrate the effectiveness of the approach.

I. INTRODUCTION

Consider a system \mathcal{N} , having an n -dimensional configuration space \mathcal{C} . Any configuration $q \in \mathcal{C}$ is represented by a list of n coordinates (q_1, \dots, q_n) . Its configuration q is a differentiable function of t . Further \mathcal{N} 's motion must satisfy a scalar kinematic constraint of the form

$$F(q, \dot{q}) = 0 \quad \forall t \quad (1)$$

where $q \in R^n$ is the configuration vector, $\dot{q} = dq/dt$ is the velocity and $F : R^n \times R^n \rightarrow R^m$, $m < n$. A kinematic constraint of the form (1) is holonomic if it is integrable, i.e., if all the velocity parameters can be eliminated and (1) can be rewritten as

$$F_1(q) = 0;$$

otherwise, the constraint is a non-holonomic constraint. It has been shown that k 'independent' non-holonomic equality constraints of the form (1) restricts the space of velocities achievable by \mathcal{N} at any configuration to an $n - k$ dimensional subspace of R^n . A non-holonomic equality constraint of the form (1) is caused by a rolling contact between two rigid objects, for example between the wheels and the surface of travel of a car. It expresses the condition that the relative velocity of the two points of contact is zero. When there is no sliding, the non-holonomic constraints are linear in \dot{q} .

A non-holonomic inequality constraint of the form

$$G(q, \dot{q}) \geq 0 \quad \forall t \quad (2)$$

usually restricts the set of velocities achievable by \mathcal{N} without changing its dimension.

An important question is whether a constraint of the form (1) also restricts the set of achievable configurations since the dimension of the control space is smaller than that of the tangent space. We are interested in situations where this set of achievable configurations is not restricted as in the case of a car, where if two configurations q and q' are located in the same connected component of the car's free space, it is known from experiment that using the velocity of the rear wheels and the steering angle of the front wheels as control parameters, the car can be driven from q to q' . This concerns controllability issues [1]-[4].

Laumond was one of the early investigators of non-holonomic motion planning. In [5], he reports navigation in constrained space at the expense of a large number of manoeuvres. In [3], he introduces a new metric in the configuration space of non-holonomic systems. An iterative motion planner based on recursive subdivision of trajectories is presented in [6] and [7]. Standard results in differential geometry and nonlinear control theory have been used in [2] to present a planner with minimal number of manoeuvres. A non-holonomic planner presented in [8] takes the free path produced by a holonomic planner and transforms it to a feasible path by successively substituting feasible subpaths for portions of the input paths until the entire path is feasible. Motion planning for a platform diver, which in essence is a non-holonomic system is presented in [9]. A method of steering non-holonomic systems using sinusoids is given in [10]. A procedure based on canonical trajectories constructed from Dubin's path can be found in [11] and [12]. A practical path planner based on building a one-dimensional maximal clearance skeleton through the configuration space, using a special metric which captures the non-holonomy of the system is given in [13]. A penalty function approach which directly constructs a non-holonomic path can be found in [14].

While all of these approaches, with exception to [9] perhaps, work with (4), the approach presented in this paper deals directly with (1) and we use the example of a car to illustrate our ideas. Section 2 gives details of our approach. Section 3 deals with the formulation of the car problem. Numerical issues are addressed in section 4. Error analysis is presented in section 5 and a few illustrative examples are given in section 6.

II. APPROACH

A non-holonomic system is characterised by an under-determined differential system of the form (1). Given configuration-velocity constraints (2) and end-point constraints,

$$q(0) = q_0, \quad q(t_f) = q_f, \quad (3)$$

the non-holonomic control problem is to find a trajectory $q : [0, t_f] \rightarrow \mathcal{R}^n$ that satisfies (1)–(3). For the car problem, (2) usually corresponds to state constraints arising out of the presence of obstacles and direct constraints on velocities.

Classical optimal control deals with dynamic systems of the form

$$\dot{q} = f(q, u) \quad (4)$$

where u is a vector of control functions that can be freely chosen. A wealth of literature is available for treating such systems. It is not surprising therefore that a lot of research effort on non-holonomic control has gone into the conversion of (1) to (4). In particular if

$$F(q, \dot{q}) = A(q) \dot{q} \quad (5)$$

and A satisfies certain conditions, then differential geometric tools can be used to convert (1) to the following special instance of (4):

$$\dot{q} = g_1(q) + g_2(q) u \quad (6)$$

Many of the recent works on non-holonomic control first do this conversion of (1) to (4) and then design $u(\cdot)$ so as to satisfy the given constraints.

In this paper we take a different approach by directly working with (1) and so avoiding u altogether. If $u(\cdot)$ is required it can be easily obtained from the $q(\cdot)$ determined by our approach by the solution (pointwise in t) of (6), which is a feasible (though over-determined) linear system of equations in u . It is usual that in most non-holonomic systems, efficient methods for determining a geometric path, $q(\cdot)$ that satisfies (2) and (3) are available. Our approach consists of taking such a $q(\cdot)$ and using a nonlinear least squares method to modify it so as to include (1). Though the car has been used as the main example for illustration, our approach is quite general and is applicable to any non-holonomic system.

III. FORMULATION OF THE CAR PROBLEM

Figure 1 shows a four wheeled car, C , modelled as a two dimensional object translating and rotating in the plane. The rear wheels are aligned with the car while the front wheels are allowed to spin about their respective vertical axes. The front and rear pairs of wheels are modelled as single wheels at the mid point of the respective axles. The constraints on the system arise by allowing the wheels to roll and spin but not slip.

The configuration space of the robot is $\mathcal{D} \times S^1$, where \mathcal{D} is a compact domain of \mathcal{R}^2 . \mathcal{D} is compact since the range of positions reachable by the robot is

bounded. The robot configuration is parameterized by the co-ordinates x and y of the mid point between the two rear wheels and the angle θ between the X-axis of the cartesian frame embedded in the plane and the main axis of the car. The steering angle ϕ measures the orientation of the front wheels with reference to the main axis of the car. The control inputs of the car are the velocity $U_1^* \in \mathcal{R}$ of the front wheels in the direction in which the front wheels are pointing, and the steering velocity $U_2^* \in \mathcal{R}$.

Assuming that there is no slipping, the velocity of the point (x, y) at the midpoint of the rear wheel axle is always parallel to the main axis of the car. Hence the resultant sideways velocity of the wheels is zero. The constraints for the front and rear wheels are formed by writing the expression for the sideways velocity of the wheels and setting it equal to zero as:

$$F(q, \dot{q}) \triangleq \begin{pmatrix} \dot{x} \sin \theta - \dot{y} \cos \theta \\ \dot{x} \sin(\theta + \phi) - \dot{y} \cos(\theta + \phi) - \dot{\theta} \cos \phi \end{pmatrix} = 0 \quad \forall t \quad (7)$$

where $q = (x, y, \theta, \phi)$ denotes the configuration of the robot.

The system equations may be easily written in a traditional control-theoretic form as :

$$\begin{aligned} \dot{x} &= \cos \theta U_1 \\ \dot{y} &= \sin \theta U_1 \\ \dot{\theta} &= \frac{1}{l} \tan \phi U_1 \\ \dot{\phi} &= U_2 \end{aligned} \quad (8)$$

State constraints arise from obstacle-avoidance considerations. We assume that obstacles are expressed as the finite union of convex polygons. Each convex polygon is represented by its vertex list given in order. The work space of the car is also bounded by a finite union of convex polygons. In order to express the state constraints mathematically we need to quantify the proximity of a pair of objects represented as convex polygons.

Let P and Q be two convex polygons in \mathcal{R}^2 . Let $\{p_1, \dots, p_m\}$ and $\{q_1, \dots, q_n\}$ be the ordered vertex lists of polygons P and Q respectively. Let \bar{p} and \bar{q} , $\bar{p} \neq \bar{q}$ be specified reference points in the interiors of P and Q respectively. We define the 'expansive distance between P and Q ,' $d(P, Q)$ [15] as:

$$d(P, Q) = \min\{\lambda : \bar{p} + \lambda P \cap \bar{q} + \lambda Q \neq \emptyset\} - 1 \quad (9)$$

It is easy to see that:

- (i) $d(P, Q) + 1$ denotes the least expansion (contraction is taken as negative expansion) of P and Q about their reference points, so as to reach a 'just touching' position.
- (ii) $P \cap Q = \emptyset$ iff $d(P, Q) > 0$
- (iii) $\text{int } P \cap \text{int } Q \neq \emptyset$ iff $d(P, Q) < 0$
- (iv) P and Q are 'just touching' iff $d(P, Q) = 0$.

Thus positive $d(P, Q)$ implies that the polygons do not intersect, and negative $d(P, Q)$ implies that the polygons intersect. The condition, 'P and Q avoid collision,' can now be expressed as $d(P, Q) \geq 0$. If P and Q are represented as convex hulls of points and n denotes the total number of points then $d(P, Q)$ can be computed in time $O(n)$ using a linear programming formulation.

Now, let $P(q)$ be the polygon denoting the space occupied by the robot while at configuration q , and $Q(i)$ denote the i th obstacle. Then obstacle-avoidance is expressed as:

$$d(P(q), Q(i)) \geq 0 \quad \forall i \quad (10)$$

In addition there are velocity constraints of the form $\dot{q}_i \leq |M_i|, i = 1, \dots, n$, where M_i are suitable bounds. Collecting all such constraints we obtain (2).

IV. NUMERICAL APPROACH

We represent $q(\cdot)$ using cubic splines and work with the spline coefficients so as to have a finite dimensional problem to solve. It is easy to choose a spline formulation that explicitly enforces the end-point constraints in (3) [16]. We use a nonlinear least squares approach to enforce the remaining constraints, i.e., (1) and (2). The non-holonomic constraint, (1) is already in equality form and hence it naturally fits into a nonlinear least squares formulation. We take care of (2) by using the following error function:

$$e_1(q) = E_1 \exp(-k d(G(q, \dot{q}))) \quad (11)$$

where E_1 is a weight and k is a scalar parameter which are appropriately chosen. $e_1(q)$ is non negative and, keeping $e_1(q)$ close to zero will lead to the satisfaction of (2). As we mentioned earlier, an initial $\tilde{q}(\cdot)$ that satisfies the end point and state constraints is available. Such a geometric path can be constructed using cubic splines, following a road map which consists of a set of knot-points. Concepts of optimality of path such as 'minimum path length' can be included during the determination of \tilde{q} itself. Our aim is to modify \tilde{q} so as to include (1). While doing this it is a good idea to keep q close to \tilde{q} . To do this, we introduce another error function,

$$e_2(q, t) = E_2(q - \tilde{q}(t)), \quad (12)$$

where E_2 is an appropriately chosen weight. If the size of $e_2(q, t)$ is small then q and $\tilde{q}(t)$ are close to each other. In this way, all the constraints nicely fit into a nonlinear least squares formulation.

Let us define

$$h(q, \dot{q}, t) = \begin{pmatrix} F(q, \dot{q}) \\ e_1(q) \\ e_2(q, t) \end{pmatrix} = 0 \quad \forall t$$

Also, let v denote the finite-dimensional vector of spline variables that describe $q(\cdot)$. We solve the nonlinear least squares problem,

$$\min_v \rho(v) \triangleq \int_0^T \|h(q, \dot{q}, t)\|^2 dt, \quad (13)$$

using the Levenberg-Marquardt trust region approach [17], which is currently the most effective method known for solving nonlinear least squares problems. We have used the popular package, MINPACK [18]. This package is devised to drive a finite number of error functions towards zero, and hence it cannot directly solve (13) which requires that a continuum of error functions be made zero, i.e., $h(q, \dot{q}, t) \approx 0 \quad \forall t$. To overcome this issue we choose a small Δ and replace (13) by the problem,

$$\min_v \tilde{\rho}(v) \triangleq \sum_{i=0}^N \|h(q(t_i), \dot{q}(t_i))\|^2 \cdot \Delta, \quad (14)$$

where $N = \lfloor T/\Delta \rfloor$ and $t_i = i\Delta$. In the next section we will give a systematic way of choosing Δ .

Let $H(v)$ denote the single vector function which is the collection of $h(q(t_i), \dot{q}(t_i), t_i) \quad \forall i$. Then (14) can be compactly rewritten as

$$\min_v \|H(v)\|^2. \quad (15)$$

Starting from an initial vector, v_0 , the Levenberg-Marquardt trust region algorithm generates a sequence of iterates, $\{v_k\}$ which, under suitable conditions, converge to a solution v^* of (15). Let H_v denote the Jacobian of H with respect to v . The trust region method generates a sequence $\{s_k\}$ of iterates where the step s_k between iterates is a solution of the subproblem,

$$\min \{ \|H(v_k) + H_v(v_k)s\| \quad s.t. \quad \|D_k s\| \leq \Delta_k \} \quad (16)$$

for some bound Δ_k and diagonal scaling matrix D_k . The motivation for the choice of step s_k is that in a neighbourhood of x_k we expect the linear model

$$L_k(s) = h_k + J_k s, \quad (17)$$

where $h_k = H(v_k), J_k = H_v(v_k)$, to provide a reasonable prediction of the behaviour of H . Given an iterate v_k , a bound Δ_k and a scaling matrix D_k , a trust region method computes the tentative step s_k . The reduction produced by the step s_k is measured by the ratio

$$\rho(k) = \|H(v_k)\|^2 - \|H(v_k + s_k)\|^2 / \|L_k(0)\|^2 - \|L_k(s_k)\|^2.$$

The trust region method attempts to keep ρ_k close to unity while keeping Δ_k relatively large. If the step is satisfactory in the sense that s_k produces a sufficient reduction, then Δ_k is increased, else it is decreased.

V. ERROR ANALYSIS

There are two levels of errors whose effects on the solution are to be addressed. The first level of errors is caused by the approximation of the integral objective function in (13) by a finite sum over a grid. The second level is caused by the fact that $F(q, \dot{q})$ cannot be made identically zero because of the tolerances used to finitely terminate the nonlinear least squares

numerical procedure. In the presence of these two levels of errors, the $q(\cdot)$ obtained via the numerical approach outlined in section 4 will, in all probability, not be a trajectory that satisfies the non-holonomic constraints. Therefore, there are two questions that need to be addressed: (i) What is a non-holonomic trajectory, $\tilde{q}(\cdot)$ that is nearly $q(\cdot)$? and, (ii) How should the various tolerances and grid sizes be chosen so that $\tilde{q}(\cdot)$ satisfies (2) and (3)? In this section we briefly describe ideas which contain answers to these questions.

We begin with an analysis of the first level of errors. Let Δ_s be the size of the spline segments and, as in (14), Δ the size of the segment used for replacing (13) by (14). A good control of the errors is obtained if we choose $\Delta = \Delta_s/K$, where K is an appropriately chosen positive integer. Let us make the following assumption.

Assumption 1. F is Lipschitz continuous and, $\|q\|$ and $\|\dot{q}\|$ are uniformly bounded.

While the Lipschitz condition has to be checked for the given problem (for instance, for the car problem this is easy to show) the bounds for $\|q\|$ and $\|\dot{q}\|$ can be chosen and enforced via (2).

If assumption 1 holds, then we show the following. Suppose the nonlinear least squares numerical method uses a tolerance, ϵ and returns a $q(\cdot)$ that satisfies

$$\|F(q(i\Delta), \dot{q}(i\Delta))\| \leq \epsilon \quad \forall i. \quad (18)$$

Then we have

$$\|F(q(t), \dot{q}(t))\| \leq \epsilon + \delta(\beta) \quad \forall t \quad (19)$$

where $\beta = 1/K$, $\delta(\beta) = a_1\beta + a_2\beta^2 + a_3\beta^3$, and, a_1, a_2, a_3 are positive constants which are simple functions of Δ_s and the Lipschitz and bounding constants of assumption 1.

Remark 1. Thus, if Δ_s has been prechosen, then $\delta(\beta)$ can be made arbitrarily small by choosing K large. It is a good idea to choose the smallest K satisfying $\delta(\beta) \leq \epsilon$, so that

$$\|F(q(t), \dot{q}(t))\| \leq 2\epsilon \quad \forall t. \quad (20)$$

Now consider the problem of finding a non-holonomic trajectory, $\tilde{q}(\cdot)$ which is near $q(\cdot)$. In doing this we only concentrate on systems of the form (5)-(6). Let $L(q) = (g_2^T g_2)^{-1} g_2^T$ (here T denotes transpose) and define

$$u(t) = L^T(q(t))(\dot{q}(t) - g_1(q(t))) \quad \forall t, \quad (21)$$

a control trajectory derived from $q(\cdot)$. Now define $\tilde{q}(\cdot)$ to be the solution of (6) with $\tilde{q}(0) = q_0$ and $u(\cdot)$ given by (21). Clearly, $\tilde{q}(\cdot)$ is a non-holonomic trajectory and, because $u(\cdot)$ is derived from $q(\cdot)$, we expect $\tilde{q}(\cdot)$ to be close to $q(\cdot)$. To obtain an a priori bound on this closeness we require the following assumption. Let

$$\alpha(q) = L(q)g_1(q) \quad \text{and} \quad P(q) = \begin{pmatrix} A(q) \\ L(q) \end{pmatrix}.$$

Assumption 2. P and α are Lipschitz continuous, and, $\|P^{-1}\|$ and $\|\dot{q}\|$ are uniformly bounded. (Note

that, if the constraints in (5) are 'independent' then the definition of $L(q)$ and the relation between $A(q)$ and $g_2(q)$ ensure that $P(q)$ is nonsingular.)

If assumption 2 and (20) hold then we show that

$$\|\tilde{q}(t) - q(t)\| \leq 2(\exp(cT) - 1)\epsilon/c \quad \forall t, \quad (22)$$

where c is a constant which is a simple function of the Lipschitz and bounding constants of Assumption 2.

The a priori bounds derived above have some limited use. Our evaluation of (22) on a number of instances of the car problem of section 3 has shown that the bound is far from tight. Hence it is not a good idea to use it for setting up the tolerances in a practical algorithm. We have found the following practical algorithm to be quite effective.

Practical Algorithm.

1. Choose initial small values for Δ_s and ϵ , and large values for E_1, k and E_2 .
2. Choose K as in Remark 1.
3. Solve (15) using a nonlinear least squares numerical method with the aim of finding a v and a corresponding $q(\cdot)$ that satisfies $\|h(q(i\Delta), \dot{q}(i\Delta))\| \leq \epsilon \quad \forall i$. Now execute one of the following substeps.
 - (a) If the method is unable to find a such a $q(\cdot)$, it means that the spline grid is not fine enough. In this case decrease Δ_s and go back to step 2.
 - (b) If the method finds a $q(\cdot)$, but $q(\cdot)$ does not satisfy (2), then increase E_1 and k , and go back to step 2.
 - (c) If the method finds a $q(\cdot)$ which satisfies (2) go to step 4.
4. Compute $\tilde{q}(\cdot)$ and check if $\|\tilde{q}(T) - q_f\|$ is small enough and $\tilde{q}(\cdot)$ satisfies (2). If so, stop. Else, decrease ϵ and go back to step 2.

If the variables in step 1 are chosen properly for the particular problem being solved, then one entrance into step 3 is sufficient to find a feasible non-holonomic path, $\tilde{q}(\cdot)$. Even if looping back to step 3 is required, we should note that the most recent $q(\cdot)$ (and the corresponding v) can be used to restart the modified nonlinear least squares solution and hence only an incremental amount of work is required in step 3.

VI. EXAMPLES

Our procedure was implemented on a Personal IRIS 4D/20 work station with interactive graphics showing the moving object, the obstacles and the path at the end of each iteration. We tested the method on a variety of examples in a simulated environment using a car with a width to length ratio of 1:4. Figure 2 shows an example with a few obstacles. Figure 3 shows the same example executed without enforcing (12). Figure 4 shows two examples with a more cluttered work

space. On all the examples tried, we found that, even though the initial solution violates (1) very badly, just a few nonlinear least squares iterations are sufficient to enforce (1) nicely.

VII. CONCLUSION

In this paper we have presented a numerical path planner which, given a geometric feasible path, generates a non-holonomic path using a nonlinear least squares approach. Several examples have been included to illustrate our approach. An elegant error analysis has been included to justify the approximations. The approach can be applied to any non-holonomic system. We hope to try our approach on trailer and fire-truck examples which have more states and control parameters.

References

- [1] Z.Li, R.M.Murray and S.S.Sastry, *Robotics: Manipulation and Planning*, Preprint, December 1990.
- [2] J.Barraquand and J.C.Latombe, "On non-holonomic mobile robots and optimal maneuvering," 4th International Symposium on Intelligent Control, Albany, 1989.
- [3] J.P.Laumond, "Non-holonomic motion planning versus controllability via the multibody car system example," Report # STAN-CS 90-1345, 1990.
- [4] J.Barraquand and J.C.Latombe, "Non-holonomic multibody mobile robots : Controllability and motion planning in the presence of obstacles," IEEE International Conference on Robotics and Automation, 1991.
- [5] J.P.Laumond, "Feasible trajectories for mobile robots with kinematic and environment constraints," International Conference on Intelligent Autonomous Systems, Amsterdam, pp. 346-354, 1986.
- [6] P.Jacobs, J.P.Laumond and M.Taix, "A complete iterative motion planner for a car like robot," Journées de Géométrie Algorithmique, INRIA, June 1990.
- [7] P.Jacobs, J.P.Laumond, M.Taix and R.Murray, "Fast and exact trajectory planning for mobile robots and other systems with non-holonomic constraints," Technical Report # 90318, LAAS, CNRS, Toulouse, France, September 1990.
- [8] J.C.Latombe, "A fast path planner for a car-like indoor mobile robot," 9th National Conference on Artificial Intelligence, AAAI, CA, 1991.
- [9] N.V.R.K.N.Murthy and S.S.Keerthi, "Optimal control of a somersaulting platform diver : A numerical approach," IEEE International Conference on Robotics and Automation, Atlanta, May 1993.
- [10] R.M.Murray and S.S.Sastry, "Steering non-holonomic systems using sinusoids," IEEE Conference on Decision and Control, pp.2097-2101, December 1990.
- [11] P.Jacobs and J.Canny, "Planning smooth paths for mobile robots," IEEE International Conference on Robotics and Automation, pp.2-7, May 1989.
- [12] P.Jacobs and J.Canny, "Robust motion planning for mobile robots," IEEE International Conference on Robotics and automation, 1990.
- [13] B.Mirtich and J.Canny, "Using skeletons for non-holonomic path planning among obstacles," International Conference on Robotics and Automation, pp. 2533-2540, May 1992.
- [14] S.Samuel and S.S.Keerthi, "Numerical Determination of Optimal Non-Holonomic Paths in the Presence of Obstacles," to be presented at the IEEE International Conference on Robotics and Automation, Atlanta, May 1993.
- [15] C.J.Ong, *Penetration Distances and their Applications to Path Planning*, Ph.D. Thesis, Mechanical Engineering Department, The University of Michigan, 1993.
- [16] W.H.Press, B.P.Flannery, S.A.Teukolsky and W.T.Vetterling, *Numerical Recipes*, Cambridge University Press, Cambridge, 1986.
- [17] W.R.Cowell, *Sources and Development of Mathematical Software*, Prentice-Hall Inc., 1984.
- [18] J.J.Moré, B.S.Garbow and K.E.Hillstom, "User guide for MINPACK-1," Argonne National Labs Report ANL-80-74, 1980.

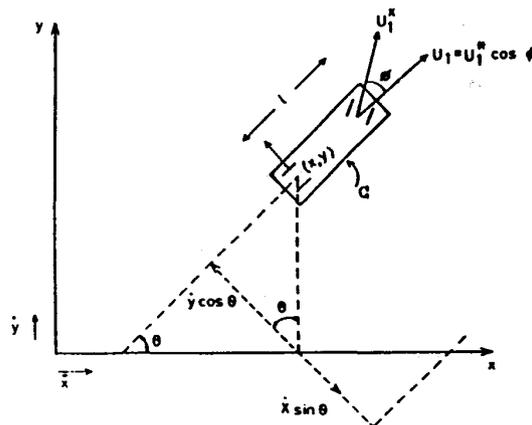


Fig. 1. Four wheeled car-like robot

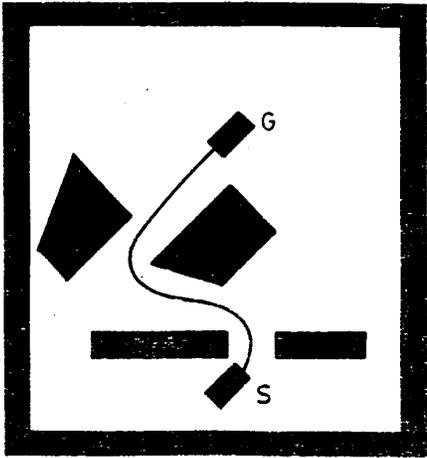


Fig. 2. Example with a few obstacles

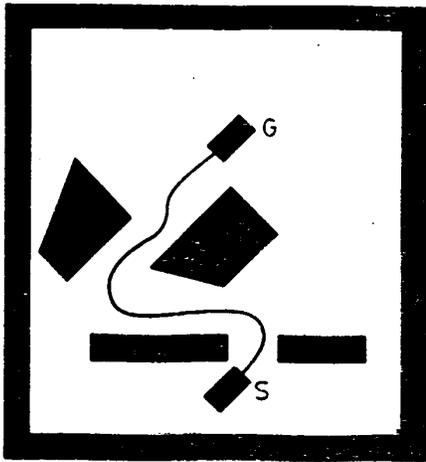


Fig. 3. Example without enforcing (12)

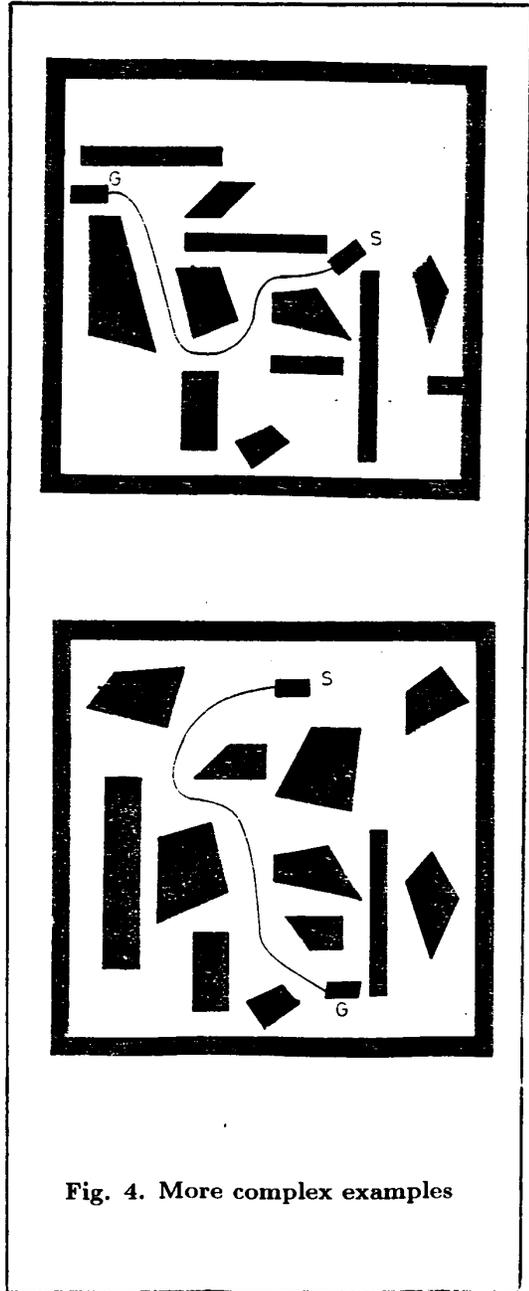


Fig. 4. More complex examples