Journal of Theoretical
and Applied Mechanics
Sofia 1994–95, Year XXV No 1–2

PLANNING OF SMOOTH COLLISION–FREE PATHS FOR MANIPULATION ROBOTS

V. BALAVESSOV*

1. Introduction

Collision avoidance is recognized as an essential issue in current robotic research. Many reports have been devoted on path planning for robots working in cluttered environment and on trajectory generation for robots operating jointly in space. The complexity of the problem of obstacle avoidance is defined by the complexity of any of the following associated problems:

1. modelling of obstacles;
2. obtaining the distance/proximity of robot links to various objects in the working space;
3. searching for collision-free paths; and
4. obtaining optimal collision-free paths.

Obstacle modelling is actually a tool for estimation of some proximity measure, *eg* distance between robot links and objects [1], or safety clearance. This is used for construction of working space maps [2] and for obtaining artificial repelling forces [3], [4] or possible directions of motion that direct the links away from obstacles [5]. Obstacles may be represented either in task space, *eg* [4], or in configuration space, *eg* [6]. In task space, obstacles are usually approximated by unions of a limited number of *primitive forms* [4], by *polyhedra* [2], [7], or *polytopes* [2]. Typically, links are shrunk to line segments, while obstacles are extended in such a way that a safety clearance is ensured. Difficulties arise in representing concave objects, unless they are considered as unions of convex forms. Primitive forms modelling is closely related to the concept of artificial potential [3] that assumes the existence of repelling

* The support of National Research Foundation of Bulgaria under grants NIMM 44/91 and NITN 427/94 is gratefully acknowledged

forces. The polyhedral representation usually goes together with linear programming techniques for path optimization [2]. The polytope representation is well suited for obtaining of link-to-object distances [2]. Some approaches utilize the configuration space representation of the constraints, thus transforming actual obstacles into a single configuration-space obstacle, and shrinking the robot into a point [6, 8]. This approach inevitably suggests some kind of discretization of the configuration space and certain types of configuration-space maps [2]. The virtues of the approach are based on the easy and fast access to the computer memory where the configuration space map is stored prior to operation, on variety of algorithms already developed for VLSI routing, and on the convenient representation of arbitrarily shaped obstacles. Different real obstacles may overlap in the configuration space, *ie* it is hard to distinguish one from the other or to perform motion of an obstacle there, unless the configuration obstacles are represented by *space slices* [6]. The path-find problem does not regard path optimization. Usually the *artificial potential* concept [4] or the *penalty functions* approach are utilized, but there exist various kinds of euristic approaches, too. Typically, the bulky inverse kinematics computations are circumvented and the corresponding real time implementation is easy from computational viewpoint, especially for a non-complex environment. However, special efforts are required against trajectory jamming. The problems of constraint path optimization are based on the minimization of an objective function, that considers both the constraints and various costs associated with the trajectory such as length, safety (clearance), and time of completion. Constraint path optimization problems utilize complex mathematical techniques, like linear programming, numerous modifications of the visibility graph approach [9, 10], or the Pontryagin principle. The *time optimal path planning* problem [9] corresponds to minimization of the travelling time. It is associated with consideration of the complete dynamics of the robot and the saturation limits for motor torques, joint velocities and accelerations. Uncertain dynamics parameters make the problem even more complex. Generally, the problem is hard to solve. Less difficulties arise in solving the problem of time optimal trajectories, where the path is known a priori [2, 11]. The *shortest path planning* problem [10] implies that the travelling distance is subjected to minimization. The shortest path is not necessarily the time optimal path, although it suggests intuitively a short time of completion. Shortest constraint paths usually feature discontinuous path curvatures at certain points, where the robot has to stop and start again, and this increases the travelling time. It is found, that paths with continuous curvatures better meet time optimal motion requirements [8]. Thus the problem of generating smooth collision-free paths arises

as an intermediate problem between shortest path planning and path finding, from the one hand, and time optimal trajectory planning, from the other hand.

The objective of this report is an optimization approach to obstacle-free path planning that provides smooth paths at low computational costs through the choice of an appropriate criterion.

2. Problem statement

Robot motion planners are computer-based and therefore they are intrinsically discrete. Hence, it is reasonable to search for a discrete collision-free path rather than generate a continuous one and then form some discrete path on this base.

Consider a mechanical system with n degrees of freedom. Denote by $q \in \Re^n$ an arbitrary configuration, and by q_{init} and q_{fin} the initial and the destination configurations of the manipulator path, respectively. Suppose that the robot environment is modelled, and a function $h(q) : \Re^n \mapsto \Re$ is defined such that

1. $h(q) = 1$ when the robot collides with the surface of some obstacle;
2. $0 \leq h(q) < 1$ when the robot is away from the obstacles;
3. $h(q) > 1$ when the robot "gets inside" some obstacle;
4. $h(q)$ increases with the approach of the robot to obstacles, and *vice versa*;
5. $\forall q \ \exists \dfrac{\partial h(q)}{\partial q} \stackrel{\text{def}}{=} g^T(q)$, such that $||g^T(q)|| < \infty$.

It should be noted that, if the closest distance $d = d(q)$ of the robot to the environment is known, then e^{-d} is such a function, provided negative values of d are used to designate "penetration" of obstacles with robot links [5].

The objective is to find a sequence of points, that form a smooth obstacle-free path

$$q_{\text{init}} = q_1, q_2, q_3, \ldots, q_p = q_{\text{final}}, \tag{1}$$

where $q_i \in \Re^n$, $1 < i < p, p > 0$ is an integer, and q_1 and q_p are known *a priori*. The integer p defines the amount of points of the path. It has to be a trade-off, as long as it must be both large enough to ensure small distance between any two consecutive path points, and small enough to guarantee computational effectiveness of the planner. Further on the matrix

$$Q \stackrel{\text{def}}{=} [q_2, q_3, \ldots, q_{p-1}]^T \in \Re^{(p-2)\times n},$$

where the terminal configurations do not enter for convenience, is used as a compact synonym of the path (1).

The sequence of path points (1) corresponds to consecutive time samples $t_0, t_1, t_2, ..., t_p$, *ie* $q_k = q(t_k)$, and consecutive time intervals $t_{k+1} - t_k = \Delta t_k > 0$. With a single path, various robot trajectories may be generated using various samples, and various intervals Δt_k, respectively.

3. Estimation of a robot path

In order to handle the problem, estimates of the smoothness and the safeness of the path have to be considered.

Further on the cost

$$r(q_k) \stackrel{\text{def}}{=} \frac{1}{2}(q_{k+1} - 2q_k + q_{k-1})^T(q_{k+1} - 2q_k + q_{k-1}) \tag{2}$$

is referred to as *local path ruggedness* at the point q_k, $1 < k < p$.

The local path ruggedness (2) is non-negative, and it is zero everywhere only for a uniformly tabulated straight line path in joint space. If the path points in the set (1) are uniformly tabulated with respect to time, then (2) represents a generalized discrete estimate of the joint space acceleration. Therefore, the cost (2) represents a combination of the local path curvature and the local trajectory acceleration.

Robot motions usually start and terminate at zero velocity. In such cases it is reasonable to augment the set of points (1) by duplicating the terminal configurations:

$$\left| \begin{array}{rcl} q_0 & = & q_1 \\ q_{p+1} & = & q_p \end{array} \right. \tag{3}$$

The average local path ruggedness

$$r(Q) \stackrel{\text{def}}{=} \frac{1}{p}\sum_{k=1}^{p}(q_{k+1} - 2q_k + q_{k-1})^T(q_{k+1} - 2q_k + q_{k-1}) \tag{4}$$

is further on referred as *path ruggedness.* Intuitively, it is opposite to the path smoothness and correlates with the length of the path in joint space.

The path safety is estimated in a way similar to the one used for estimation of the path smoothness. The *local collision danger* estimate is represented by the function $c_{\text{loc}}(q_k) \stackrel{\text{def}}{=} h(q_k)$, $0 < k < p$. The inequality $c_{\text{loc}}(q_k) < 1$, is observed $\forall k$, along any collision-free path. The *path collision danger* estimate is defined as an average of the weighted local estimates along the path:

$$c(Q) \stackrel{\text{def}}{=} \frac{1}{p}\sum_{k=1}^{p}\sigma_k c_{\text{loc}}(q_k), \tag{5}$$

where $c(Q) : \Re^{(p-2)\times n} \mapsto \Re$ is a smooth map, and $\sigma_k > 0, \quad 0 < k < p$, are properly chosen weights. Given the weights, a safer average clearance corresponds to smaller value of the cost (5).

4. Optimization criterion and optimum paths

If (3) holds, then the path ruggedness (1) cannot be zero everywhere along the path. To obtain a smooth path, the path ruggedness (4) must be kept as small as possible. The attitude of the robot to its environment is estimated by the collision danger (5). To ensure collision-free motion, local values $c_{\text{loc}}(q_k)$ have to be kept both below 1 and as low as possible along the path, in order to obtain a safe clearance. However, the above mentioned requirements are often contradictory. Therefore, a trade-off solution has to be found.

Consider the following criterion:

$$J = r(Q) + c(Q) \longrightarrow \min, \tag{6}$$

or the equivalent one

$$J = \frac{1}{p}\left(\sum_{k=1}^{p} \frac{1}{2}(q_{k+1} - 2q_k + q_{k-1})^T(q_{k+1} - 2q_k + q_{k-1}) + \sigma_k c_{\text{loc}}(q_k)\right) \longrightarrow \min .$$

A trade-off may be obtained through subjecting the path to optimization with respect to this criterion.

The cost (6) allows at least one local minimum. The position of a local extrtemum is obtained through differentiating of the cost (6) with respect to q_k, and by equalizing the corresponding values to zero. The terminal configurations are fixed, therefore $1 < k < p$. Arranging the settings $\frac{\partial J}{\partial q_k} = 0$, the result is represented by the following system of equations:

$$M.Q = B, \tag{7}$$

where

$$M \stackrel{\text{def}}{=} [m_{i,j}] \in \Re^{(p-2)\times(p-2)}, \quad m_{i,j} = \begin{cases} 6 & \text{if } i = j \\ -4 & \text{if } |i-j| = 1 \\ 1 & \text{if } |i-j| = 2 \\ 0 & \text{if } |i-j| > 2 \end{cases};$$

$$B \stackrel{\text{def}}{=} [b_1, b_2, \ldots, b_{p-2}]^T$$

where — assuming $p > 6$,

$$\begin{aligned} b_1 &= -\sigma_2 g(q_2) + 4q_1 - q_0; \\ b_2 &= -\sigma_3 g(q_3) + 4q_2 - q_0; \\ \dots \quad &\dots \quad \dots \\ b_k &= -\sigma_{k+1} g(q_{k+1}), \forall k : \ 2 < k < p-3 \\ \dots \quad &\dots \quad \dots \\ b_{p-3} &= -\sigma_{p-2} g(q_{p-2}) \\ b_{p-2} &= -\sigma_{p-1} g(q_{p-1}) + 4q_p \end{aligned}$$

M is a constant positive definite five-diagonal matrix. Thus solving of eq. (7) with respect to the path Q is relatively simple, as much as this matrix is concerned. The use of the *Cholesky* technique is most suitable to the purpose. Problems arise however with the computation of the vectors $g(q_k)$, $2 < k < p-1$, that are included in the $B = B(Q)$ matrix, as long as path points q_k are not fixed *a priori*. Therefore some iterative computational scheme is inevitable. As long as $M > 0$, it is easy to show, that using

$$MQ^{(i+1)} = B(Q^{(i)}),$$

instead of eq. (7), where $Q^{(i)}$ refers to the i-th iteration, is equivalent to a gradient scheme for obtaining the optimum solution with respect to the cost (6). In the sequel a technique is developed in order to improve the relaxation. It is based on dividing of the path into two subpaths.

5. Obtaining of smooth paths

As already mentioned, the points q_0, q_1, q_p, and q_{p+1} can be considered as fixed ones. Suppose also that all points with odd indices q_{2k+1}, $2 < 2k+1 < p$, are fixed, too. Then the path optimum with respect to the cost (6) is obtained as a solution of the following system of equations:

$$M_{\text{even}} Q_{\text{even}} = B_{\text{even}}, \tag{8}$$

where

$$M_{\text{even}} \stackrel{\text{def}}{=} [m_{\text{even}_{i,j}}]; \quad m_{\text{even}_{i,j}} = \begin{cases} 6 & \text{if } i = j \\ 1 & \text{if } i \neq j \end{cases},$$

$$Q^T_{\text{even}} \stackrel{\text{def}}{=} [q_2, q_4, \dots, q_{2k}, \dots]$$

$$\begin{aligned} b_{\text{even}_1} &= -\sigma_2 g(q_2) + 3q_1 + 4q_3; \\ b_{\text{even}_2} &= -\sigma_4 g(q_4) + 4q_3 + 4q_5; \\ &\dots \quad \dots \quad \dots \\ b_{\text{even}_k} &= -\sigma_{2k} g(q_{2k}) + 4q_{2k-1} + 4q_{2k+1}; \\ &\dots \quad \dots \quad \dots \end{aligned}$$

If p is odd, then the last entry of B_{even} is defined by

$$b_{\text{even}_l} = -\sigma_{p-1} g(q_{p-1}) + 3q_p + 4q_{p-2}, \tag{9}$$

where $l = \frac{p-1}{2}$, else (even p) it is

$$b_{\text{even}_l} = -\sigma_{p-2} g(q_{p-2}) + 4q_{p-1} + 4q_{p-3} - q_p, \tag{10}$$

where $l = \frac{p-2}{2}$.

If, in an analogy, all points with even indices $q_{2k}, 2 \leq 2k \leq p-1$, are fixed, then the path optimum with respect to the cost (6) is obtained as a solution of the following system of equations:

$$M_{\text{odd}} Q_{\text{odd}} = B_{\text{odd}}, \tag{11}$$

where

$$M_{\text{odd}} \stackrel{\text{def}}{=} [m_{\text{odd}_{i,j}}; \quad m_{\text{odd}_{i,j}} = \begin{cases} 6 & \text{if } i = j \\ 1 & \text{if } i \neq j \end{cases},$$

$$Q_{\text{odd}}^T = [q_3, q_5, \dots, q_{2k+1}, \dots], B_{\text{odd}}^T \stackrel{\text{def}}{=} [b_{\text{odd}_1}, b_{\text{odd}_2}, \dots],$$

$$\begin{aligned} b_{\text{odd}_1} &= -\sigma_2 g(q_2) + 3q_1 + 4q_3; \\ b_{\text{odd}_2} &= -\sigma_4 g(q_4) + 4q_3 + 4q_5; \\ &\dots \quad \dots \quad \dots \\ b_{\text{odd}_k} &= -\sigma_{2k} g(q_{2k}) + 4q_{2k-1} + 4q_{2k+1}; \\ &\dots \quad \dots \quad \dots \end{aligned}$$

If p is even, then the last entry of B_{odd} is

$$b_{\text{odd}_l} = -\sigma_{p-1} g(q_{p-1}) + 3q_p + 4q_{p-2},$$

where $l = \frac{p-2}{2}$ — *cf* eq. (9), else it is

$$b_{\text{odd}_l} = -\sigma_{p-2} g(q_{p-2}) + 4q_{p-1} + 4q_{p-3} - q_p,$$

where $l = \frac{p-3}{2}$ — *cf* eq. (10).

Note, that the structures of M_{odd} and M_{even} are identical, only the dimensions of these matrices may differ by 1. They are constant positive

definite three-diagonal matrices. Thus solving of equations (8) and (11) with respect to the subpaths Q_{even} and Q_{odd} is even simpler in comparison with solving of eq. (7) with respect to the path Q. As mentioned above, problems arising with computation of the B_{even} and B_{odd} matrices imply utilization of iterative procedures. The optimum path is obtained according to the following algorithm:

1. An initial path $Q^{(0)}$ is given using some path-find technique. Alternatively, a configuration-space straight line can be used initially in the case of relatively simple robot environments;

2. $k = 0$;

3. Form the subpaths $Q^{(i)}_{\text{even}}$ and $Q^{(i)}_{\text{odd}}$ using data from $Q^{(i)}$;

4. $i := i + 1$; if $i > i_{\max}$ then jump to step 15; else continue with next step;

5. Calculate the $B^{(i-1)}_{\text{even}}$ matrix on the base of $Q^{(i-1)}_{\text{even}}$ and $Q^{(i-1)}_{\text{odd}}$;

6. Solve the system $M_{\text{even}}Q^{(i)}_{\text{even}} = B^{(i-1)}_{\text{even}}$ with respect to the subpath $Q^{(i)}_{\text{even}}$;

7. Calculate the $B^{(i-1)}_{\text{odd}}$ matrix on the base of $Q^{(i-1)}_{\text{even}}$ and $Q^{(i-1)}_{\text{odd}}$;

8. Solve the system $M_{\text{odd}}Q^{(i)}_{\text{odd}} = B^{(i-1)}_{\text{odd}}$ with respect to the subpath Q_{odd};

9. Obtain the path $Q^{(i)}$ incorporating the subpaths $Q^{(i)}_{\text{even}}$ and $Q^{(i)}_{\text{odd}}$;

10. Evaluate the deviation $\|Q^{(i)} - Q^{(i-1)}\|$;

11. If the above deviation is not sufficiently small, then go back to step 4; else go on;

12. Evaluate $h(q_j), 1 < j < p$.

13. If $\forall j \Longrightarrow h(q_j) < 1$, then jump to step 16;

14. Else for all $h(q_j) \geq 1$ increase the corresponding weights σ_j; go back to step 4;

15. Unsuccessful end of the procedure (the initial path is not appropriate; a safe path does not exist; or $i_{\max}$ is not sufficiently large);

16. Successful end of the procedure; a smooth safe path is found.

It should be noted, that assigning of various weights σ_j for the different indices j, $j = 2,3,\ldots,p$, as adopted in step 14 above, correlates with utilization of penalty functions. Instead of this, the amount of path points may be increased at path segments, where local values of the cost (6) are greater, or where local path ruggedness estimates (2) are unacceptably large. A proper interpolation technique can be used to the purpose.

6. Conclusion

The approach proposed concerns planning of smooth collision-free paths for manipulation robots. It is based on a method for optimum path finding, that ensures obtaining of a local minimum. Planning is performed in the configuration space, although a task space formulation is also possible. The generated path is similar to an n-dimensional spline with variable flexibility and elasticity, that is fixed at both ends only. The elasticity and the flexibility of this spline increase in the neighbourhood of obstacles.

REFERENCES

[1] GUO, Z., T. HSIA. Joint trajectory generation for redundant robots in an environment with obstacles. *Proc. IEEE Int'l Conf. on Robotics and Automation*, **1** (1990), 157–163.

[2] SHIH, C., T. LEE, W. GRUVER. Motion planning with time-varyings polyhedral obstacles based on graph search and mathematical programming. *Proc. IEEE Int. Conf. on Robotics and Automation*, **1** (1990), 331–336.

[3] WARREN, C. Multiple robot path coordination using artificial potential fields. *Proc. IEEE Int. Conf. on Robotics and Automation*, **3** (1990), 500–505.

[4] KHATIB, O. Real-time obstacle avoidance for manipulators and mobile robots. *Int'l J. Robotics Research*, **5** (Spring 1986), *3*, 90–98.

[5] BALAVESSOV, V. Adaptive control of robotic manipulators using local optimization of the motions. *Proc. 6th National Congr. Theor. Appl. Mechanics*, **5** (1989), 396–399.

[6] BRANICKY, M. S., W. S. NEWMAN. Rapid computation of configuration space obstacles. *Proc. IEEE Int'l Conf. on Robotics and Automation*, **2** (1990), 304–310.

[7] FUJIMURA, K., H. SAMET. Motion planning in a dynamic domain. *Proc. IEEE Int'l Conf. on Robotics and Automation*, **2** (1990), 324–329.

[8] LOZANO-PEREZ, T. Spatial planning: Configuration space approach. *IEEE Trans. on Computers*, **32** (1983), 108–120.

[9] HEINZINGER, G., P. JACOBS, P. CANNY, B. PADEN. Time-optimal trajectories for a robot manipulator: a provably good approximation algorithm. *Proc. IEEE Int'l Conf. on Robotics and Automation*, **1** (1990), 150–156.

[10] J. SUNGTAEG, S. KANG. Shortest path planning with dominance relation. *IEEE Int'l Conf. on Robotics and Automation*, **1** (1990), 138–143.

[11] BOBROW, J. E, S. DUBOWSKY, J. S. GIBSON. Time-optimal Control of robotic manipulators. *Int'l J. of Robotics Research*, **4** (1985), *3*, 93–101.

Presented at the 7th Congress on Theoretical and Applied Mechanics, Sofia, August 30 – September 1, 1993
section: Applied Mechanics