Kuipers, B. and Byun. Y-T., 1990, *A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representation*, AAAI'90 Workshop on Qualitative Vision, pp. 1 - 28.

Levitt, T.S. and Lawton D.T., 1990, *Qualitative Navigation for Mobile Robots*, Artificial Intelligence, vol. 44, pp. 305 - 360.

Levitt, T.S., 1987, *Qualitative Navigation*, Proc. DARPA Image Understanding Workshop, Los Angeles, Morgan Kaufmann.

Tsuji, S. and Zhang, J.Y., 1990, *The Qualitative Representation of Scenes along a Route*, AAAI'90 Workshop on Qualitative Vision, pp. 67-71.

Venkatesh, S. and Kieronska, D., 1992, *Landmark Based Representation for Maps and Navigation Plans*, Working Document in Preparation, Curtin University of Technology.

Zadeh,L.A., 1981, *PRUF- a meaning representation language for natural languages*, in Fuzzy Reasoning and its applications, Eds. Mamdani, E.H. and Gaines, B.R., pp. 1-66.

A60 32

TRAJECTORY PLANNING OF INDUSTRIAL ROBOTS IN A STRUCTURED ENVIRONMENT

Aldo Balestrino, Antonio Bicchi, Alberto Landi
Dipartimento di Sistemi Elettrici e Automazione (DSEA),
Università di Pisa,
Pisa, 56126, Italia, Via Diotisalvi, 2.
Tel. 0039-50-553521 - Fax: 0039-50-555636

Abstract

Trajectory planning in a given environment is a difficult task, notably if a real time solution is required. In this paper the trajectories are described by means of B-splines so that the search of an eventually optimal solution is reduced to a parametric one in contrast to functional solutions.

In order to reduce the time required for a successful computation we assume that a number of pre computed or learned trajectories (e.g. by guiding and teach pendant programming) are available. Therefore the main problems are the choice among the available trajectories and the completing of the path by subsections to be computed on line. The proposed solutions are based on the convex hull properties of the characteristic polygons describing the B-splines; simple visibility rules allow the construction of paths free from collisions. Some examples are worked out by using a CAD simulation for a typical industrial robot in a flexible manufacturing system.

Keywords: robotics, trajectory planning, B-splines, collision avoidance, rule-based decision

1.Introduction.

In the literature on robotics motion planning is a long standing, fundamental problem. Special sessions are devoted to this problem in many conferences; chapters and whole books have appeared in this field [1], [2], [3], [4], [5], [6]. In a special research program on robotics promoted by CNR, the National Council for Research in Italy, known as P.F.R. (Progetto Finalizzato Robotica) 1989-92, a research line deals with motion planning. The results given in this paper are an outgrowth of the line research the University of Pisa - DSEA. Planning refers generally both to tasks and to motions and for generic environments and manipulators the problem is very hard, typically classified in the area of artificial intelligence (AI). By delimiting the environment to Flexible Machining Systems (FMS) and manipulators to industrial robots the problem is more specific and more easily solvable. In this paper we suppose that task planning has been already solved, for example as a sequence of assembling instructions; therefore we must face only the problem of motion planning. The problem requires

(i) a description of the environment, e.g. FMS layout, numerical control machines, conveyors, tool-room and so on
(ii) a description of the manipulator, e.g. his structure, the kinematics and dynamics, the constraints on joint velocities, accelerations and torques
(iii) a description of the sub task (e.g., move the end effector from point A_0 to point A_N

through the via points A_i, i=1,2,...,N-1).

Usually the objects in the workspace of the robot are described by means of elementary solid objects, for instance polyhedra; of course the approximation used in the description of the objects heavily limits the solution techniques and the time required for executing the algorithms. The solution of motion planning would give an automatic procedure for generating motions with collision avoidance, e.g. the part-program in the specific language for the robot, alleviating the problem of software production; if the solution time is reasonably short it would be possible to add some 'intelligence' to the robot increasing its flexibility.

2.Planning Rules.

Motion planning of industrial robots in a structured environment, e.g. a FMS, depends on the available hardware and software resources. Conditions being equal we evaluate the performances by comparing the executed times of the solution algorithms. Note that if a motion must be executed periodically for a very large number of times usually the motion instructions, determined by any algorithm, e.g. by using a teach pendant, are stored in the computer memory and used in play-back. On the other hand if the manoeuvre is needed just a few times, it may happen that the programming time affects heavily the economic cost. However, if the environment includes moving parts besides the robot, e.g. conveyors or other robots sharing part of their workspaces, then play-back can prove no more applicable. In FMS among the resources we have computer hardware and CAD software. Therefore it is reasonable to assume that the environment, including the robot, its end effector and workpieces, is described in the CAD/CAM. Moreover by planning the motion with the aid of CAD/CAM we can simulate the motion and verify its correctness in a simple way. As a rule CAD is available for mechanical design but is seldom suitable for animation; however the trend is to widen existing software and to make available low price hardware without the need of expensive workstations. Being motion planning a hard problem if we try to get a practical solution we need some heuristics, i.e., we must choose rules that seem reasonable leaving the proof of their validity to an experimental verification of the results more than to analytic demonstrations. We assume that motion planning:

Rule 1
must be developed inside a CAD environment

Rule 2
can be split in trajectory planning (i.e. planning limited only to the geometric characteristics of the motion) and subsequent computation of the time history referred to curvilinear abscissa.

In this way the trajectory planner takes into account the non moving obstacles while the travelling speed is determined accordingly to speed, acceleration and torque constraints by minimizing the execution time.

Motion planning can be posed as a constrained optimization problem to be solved analytically, but this is not a practical approach, i.e. the time required for the solution may be comparable with the execution time. Mainly this occurs because the best solution may be a curve admitting a very difficult description. A simplification can be realized by a suitable

restriction of the curve family. Of course we must take into account Rule 1 and choose a family of curves leading to an easy approach to obstacle avoidance.
We assume that:

Rule 3
the trajectories are given as non uniform rational B-splines (NURBS).

In order to make the exposition very plain in the sequel we limit ourselves to B-splines [7], [8], [9], [10]. Among the various properties of B-splines we remind that

3.1 a B-spline is completely determined by a finite number of points, the vertices of its descriptor polygon
3.2 there exist algorithms for approximating B-splines with any desired degree of precision, e.g. the algorithm of De Casteljau [11]
3.3 convex hull property guarantees that B-splines lie completely into the interior of the space region defined by the convex closure of the vertices of the descriptor polygon

Property 3.1 assures that curves can be memorized recording only a finite number of points, the vertices of its descriptor polygon, while the implementation of a special processor based on property 3.2 allows the fine reconstruction of the trajectories as needed by the controllers of the joint actuators.
Property 3.3 is a fundamental aid for collision avoidance; e.g. for cubic splines if we assure that any triplet of successive vertices is visible from each other then the B-splines do not touch or intersect the obstacles.
Generally a trajectory is given by concatenating B-splines, then we can specify the motion by choosing the time history. If we need a continuous motion, including continuous speed and acceleration, and with zero speed at the end point, for instance we can choose the motion as represented by curvilinear abscissa S as:

$$S(\tau) = 6\tau^5 - 15\tau^4 + 10\tau^3 \qquad (1)$$

$$S'(\tau) = 30\tau^2(\tau^2 - 2\tau + 1) \qquad (2)$$

$$S''(\tau) = 60\tau(2\tau^2 - 3\tau + 1) \qquad (3)$$

$$\dot{S}_{max} = 15\frac{L}{8t_f} \qquad (4)$$

$$\ddot{S}_{max} = 10\sqrt{3}\frac{L}{3t_f^2} \qquad (5)$$

where :

$$\tau = \frac{t}{t_f} \qquad (6)$$

In this way the only free parameter is the race time t_f which is limited by the constraints on

maximum velocity and acceleration in the Cartesian space and taking into account the dynamics and inverse kinematics of the robot by the constraint velocity and acceleration at the joints and on maximum torques developed by the actuators. Usually we are interested in minimum race time; using C^2-splines it is possible to set up efficient algorithms for optimal motion planning [12].

Obstacle avoidance in the simplest case can be realized with a batch technique. For given CAD software and robots a human operator can interactively select triplets of points, vertices of the descriptor polygon, which are visible each other so that by rule 3.3 a trajectory free of collision can be set up. An example is shown in Fig.1 for a cylindrical robot in a FMS. Often selection of visible points is a simple task; for a planar case see Fig.2.

However note that a trajectory without collisions for the end effector not always assures that all links and joints avoid the obstacles; the structure of robot is of importance notably if it has many degrees of freedom.

Other rules must be added in order to solve efficiently the problem by taking into account the robot kinematics and geometry [13].

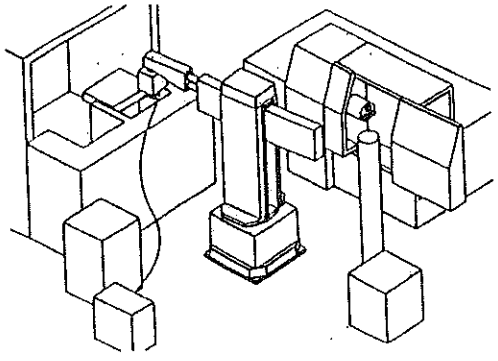

Fig.1. JOB'OT6 robot in a flexible cell

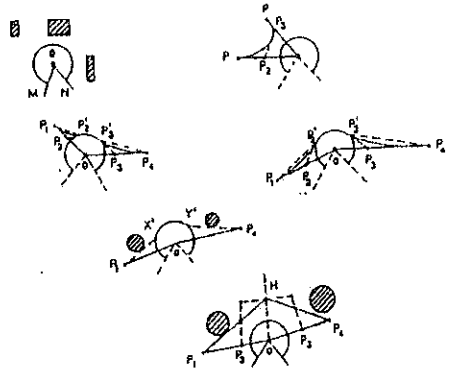

Fig.2. Path planning

3. Real-time Planning.

Also if we restrict our attention to the special case of industrial robots in FMS, planning is difficult and time consuming.

Real-time planning, e.g. automatic planning of robots in FMS while work is in progress, or also planning in a time interval comparable or less than the execution time of an elementary task by the robot, is almost impossible if every time we need to execute algorithms such as described previously.

In order to realize planning in real time we have to reduce computations to simple decisional processes that can be efficiently solved on a computer in a very short time. If we reconsider with attention the problem at hand we recognize that motion planning has two distinct features: fine or guarded motion and gross motion. In a FMS fine motion is related to operations such as assembling, pick and place or generally to operations including interactions between the end effector of the robot and other (fixed) parts of the FMS.

Gross motion is related to large transfer of the end effector from one position to another inside the workspace of the robot.

The operations requiring fine motions, such as tool insertion, grasping of objects and so on, are typical operations, depending on the FMS layout and robot structure and can be programmed out of line and made available in a suitable data base. Also many transfer operations requiring gross motion planning are highly repetitive; therefore it is convenient to set up the part programs by using play-back or algorithms as described in the previous section.

Therefore for real time planning we can select the following rule:

Rule 4

for real time planning make a batch computation of specialized fine motions and of highly repetitive gross motions.

This rule is in accordance with industrial practice and reduces the planning problem to the simplest one of choosing the right motion from the data base.

The real-time planning problem is significant if we note that a high number of trajectories may lead to unmanageable data bases and that, mainly if we have other moving parts besides the robot, we may need a motion planning in a new situation not yet considered.

What is needed is a compromise between data base requirements and computation time of new motions.

We can add the following rule:

Rule 5

realize a covering of the robot workspace with a (minimum) finite number of reference trajectories.

With 'covering' we mean that every point in the robot workspace can be transferred to at least one point of a reference trajectory and that on every pair of trajectory there are at least two points visible from each other.

It is now apparent that adopting these rules leads to a motion planning procedure that can be decomposed in two parts:

I - the choice of the segments of reference trajectories

2 - the computation of smoothed connections among these segments and with the end points of the desired trajectory.

The final trajectory results from patching segments and smoothed connections. The smoothed connections may be computed as described in the previous section using B-splines. The switching from a segment of a reference trajectory to another one or the connections of the end points can be determined by computing and comparing the mutual distances. Of course if the reference trajectories give a good covering of the robot workspace, we are almost always sure that connections do not give rise to collisions; however if a collision occurs for a robot link or joint, it is a simple matter to set up a correct transfer by a local search.

This procedure has been implemented in Pascal for a robot PUMA 560 using a world modeller (WM) [14] .

The choice of the WM is due to the need of complete transparency of data base organization and the non availability of source codes for commercial WM.

We have implemented various subroutines for computing inverse kinematics, distance between a point and a curve, for selecting segments of trajectories and smoothing connections. The computations are kept at a minimum often being only decisional processes. A simple planar exemplification is given in Fig. 3 with two obstacles and two reference trajectories.

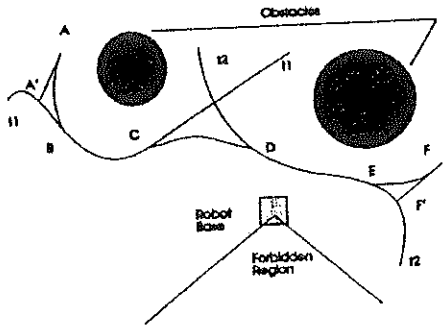

Fig.3 Example 1

Determined the nearest curves to the starting point A, i.e. trajectory t_1, and to end point F, i.e. trajectory t_2, the procedure selects the segments B-C and D-E and computes the smoothed connections (B-splines) A-B, C-D, and E-F.

Note that curves t_1 and t_2 share a point, but it is not used for switching.

Examples of solved motion planning referred to a PUMA 560 are given in the following figures. In Fig.4 the robot is considered in a rest position. Two pre-computed trajectories t_1 and t_2 are shown. The movement from Ps to P_g is to be planned.

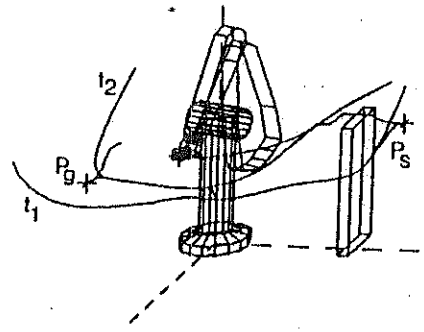

Fig.4

Fig. 5 shows a rotation around the first joint in order to align the end-effector with the starting point.

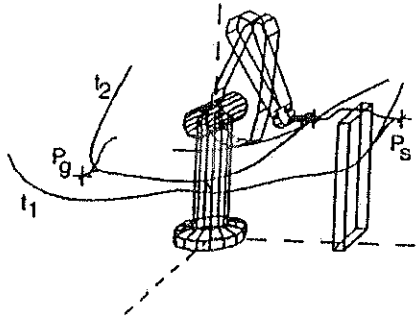

Fig.5

In fig.6 the end effector reaches the starting point P_s.

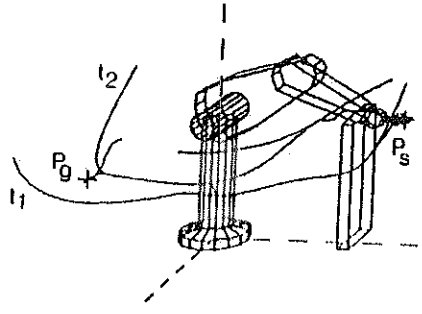

Fig.6

In Fig.7 the end effector is connected to the pre-computed trajectory t_1.

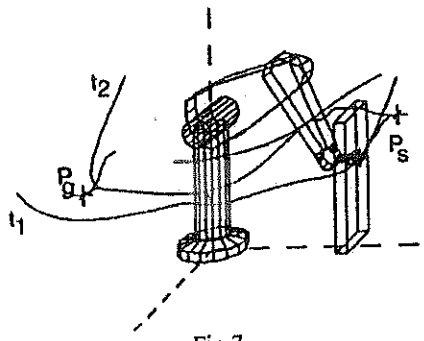

Fig.7

In Fig.8 the switching between t_1 and t_2 is shown.

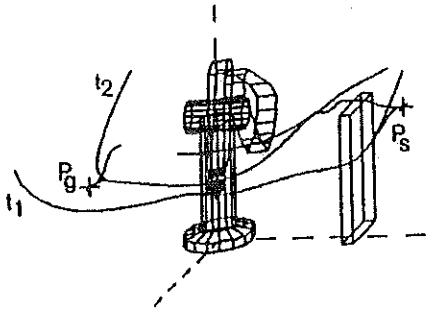

Fig.8

Fig.9 shows the robot leaving the pre-computed trajectory t_2 to reach the ending point P_g.

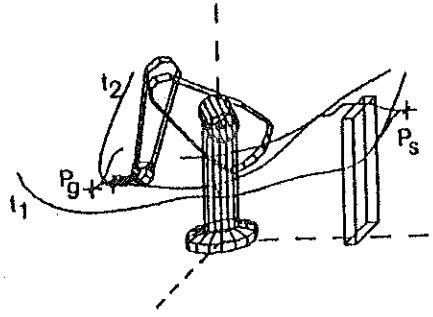

Fig.9

Fig. 10 shows the robot in the ending position P_g after a deviation from the pre-computed trajectory t_2.

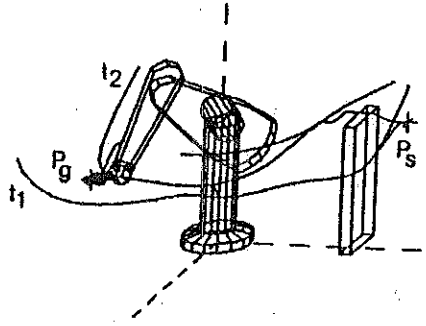

Fig.10

In Fig. 11 the robot reaches again the resting position.

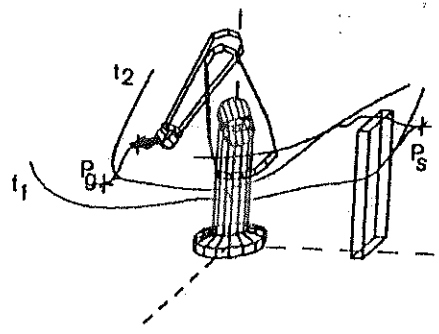

Fig.11

Figures were made possible by a dedicated software developed in Pascal both on a MacII and PC-IBM platforms.
Another example is shown in Fig.12.

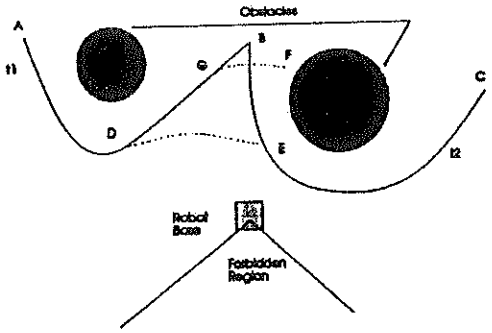

Fig.12 Example 2

In a first case, the transfer from A to C is realized by joining two segments of the reference trajectories t_1 and t_2 and a connection arc D-E. If there is no time for computation an obvious solution would be the execution of t_1 and t_2 using the via point B; however the race time in this case is certainly greater than the previous one.
Therefore the problem to be faced is an optimization problem: minimize the race time or the length of the trajectory by using a minimum number of connection segments.
With reference to Fig. 4 and to transfer from A to C it is obvious that this optimization problem reduces only to the correct choice of via points D and E.
If the transfer is from A to F the optimal solution depends only on the choice of via point G. It is apparent that in more general cases the optimization problem can lead to lengthy procedures not usable at all in real time planning.
To overcome these difficulties it is possible to demand to the lowest level of control, the actuator level, the smoothed transfer of the end effector from initial pose to reference trajectories, between reference trajectories and the end point. The selection of the active reference trajectory in dependence of the robot pose and end point to reach has to be solved at the upper level.

4. Conclusions.

A heuristic approach to motion planning of industrial robots in a structured environment has been presented. The robot structures considered are the cylindrical one and PUMA like; research activity is currently devoted to extension of the procedures to other robot structures.
Problem of optimization and hierarchical control will be faced in the next future.

5. Acknowledgment.

The financial support of CNR-Progetto Finalizzato Robotica is gratefully acknowledged.

6. References

[1] Brady M.,Hollerbach J.M.,Johnson T.L.,Lozano-Perez T.,Mason M.T. (Eds.) "Robot Motion: Planning and Control" *The MIT Press, Cambridge*, 1982.
[2] Brady M.,Paul R. (Eds.) "Robotics Research-The First Int. Symp." *The MIT Press, Cambridge*, 1984.
[3] Hanafusa H.,Inoue H. (Eds.) "Robotics Research-The Second Int. Symp." *The MIT Press, Cambridge*, 1985.
[4] J.C.Latombe: "Robot Motion Planning", *Kluwer Academic Publishers*, 1991.
[5] A. Balestrino: "Costruzione di Traiettorie in Tempo Reale", Atti del Convegno *'Robotica tra Scienze e Tecnologia'*, SIRI Milano, Marzo 1988.
[6] A. Balestrino: "Pianificazione Automatica delle Traiettorie senza Collisioni di Robot Manipolatori" Seminario su *"La Fabbrica Automatica alle Soglie del 2000 - Risultati e Prospettive della Ricerca"*, pp. 125-140, Istituto di Tecnologia Meccanica - Pisa 1989.
[7] De Boor C."A Practical Guide to Splines" *Springer-Verlag*, New York, 1978.
[8] Schumaker L. "Spline Functions: Basic Theory" *J. Wiley, New York*, 1981.
[9] Newman W.M.,Sproull R.F. "Principles of Interactive Computer Graphics" , *McGraw Hill, New York*, 1981.
[10] Rogers D.F.,Adams A.J. "Mathematical Elements for Computer Graphics" *Mc Graw Hill, New York*, 1976.
[11] P.deF.de Casteljau, "Formes a Poles" in *Mathematiques et CAO*, Hermes Publishing, 1985.
[12] C.S.Lin, P.R.Chang and J.Y.S.Luh, "Formulation and Optimization of Cubic Polynomial Joint Trajectories for Industrial Robots" *IEEE Trans. Aut. Control*,AC-28, pp. 1067-1073,1983.
[13] A. De Luca, C. Oriolo: "The Reduced Gradient Method for Solving Redundancy in Robot Arms", *XI IFAC World Congress*, Tallinn 1990.
[14] C.Mirolo and E.Pagello, "A Solid Modelling System for Robot Action Planning" *IEEE Computer Graphics and Applications*,pp.55-69,1989.