# Robot motion planning
# a survey.

Colin G. Johnson.

Department of Mathematics
Napier University, Edinburgh.

May 3rd, 1995.

## Abstract

This report surveys work on geometry of motion planning for robotic manipulators, and gives a non-technical introduction to some of my recent work in this field. This is an active field of research in which there are a number of outstanding open problems and areas worthy of exploration. The report closes by describing work which I am hoping to carry out in the near future.

# 1 Review of existing work.

The field of robot motion planning has been an active area of research over the last twenty years. The theoretical study of motion planning, rather than just *ad hoc* solutions to individual problems can be traced back to three major sources—the "Configuration Space" concept of Lozano-Pérez [LPW79], the "Piano Movers" papers by Schwartz and Sharir (collected in [HSS87]) and the book *Analysis of Mechanisms and Robot Manipulators* by Joseph Duffy [Duf80]. These works have provided the foundation for more recent theoretical studies. Work since these seminal papers is summarized below, and more details can be found in the survey of Hwang and Ahuja [HA92].

The basic problem could be said to be this: a *robot arm* is set a *task* in an *environment*. By a robot arm is meant a set of jointed rigid bodies able to take different configurations, and to move between these configurations with prescribed limits on velocity and acceleration. Industrial robot arms differ by the size of the rigid bodies, the type of each joint, the order in which the joints are connected and the range of motion permitted at each joint. The individual rigid bodies are called *links* and a typical industrial robot consists of 6 links.

In reality very few arms operate without some additional constraints on their motion in the form of obstacles in the workspace. For a planar robot we can consider several different types of obstacles:

1. Points.

2. Infinite straight lines.

3. Finite straight lines and polygons.

4. Curves, which could be represented by B-splines.

These shapes generalize easily to three dimensions.

## 1.1 Robot tasks.

A robot is required to perform a variety of tasks within its workspace, constrained by its physical form and by the environment in which it moves. A summary of possible tasks is given later. Let the *physical space* be the geometrical representation of the physical space in which the robot is moving, and the *Configuration Space* be a space for which each point corresponds to positioning of every joint in the arm. This space has in prior work often been parameterized by the joint angles and lengths, but there may be other representations which are more effective from either a computational or geometric point of view. Let the *end effector* be the point at the free end of the final link in the kinematic chain.

An important goal in robotics research is the production of *task-level planning systems*. On most current industrial robot systems work must be programmed into the robot at the *joint-level*, i.e. the robot must be programmed to move each individual joint, or at the *robot level*, where the robot as a whole is guided through a sequence of motions which it can then play back. Ideally though a robot should be able to work at a higher level—the *task level*, where the robot is given a sequence of points to visit and tasks to do at those points, and then works through those tasks in the required order, using its awareness of the surroundings to avoid obstacles, avoid self-collisions and singularities and calibrate itself. It is at this level that most current research is based.

There is a higher level still—*objective-level planning*, in which the robot has a number of skills in its repertory and is able to apply these skills to a problem in any situation, rather than being expected to work in a specially prepared environment. Typical objective-level commands would be "Wash the car" for a cleaning robot, or "shear all of the sheep in the pen" for a sheep shearing robot. The theory behind objective-level planning is still in its infancy, and the only existing systems work within very restricted environments, such as the table-tennis robot [And88] and Winograd's "blocks world" SHRDLU [Win72].

Task-level planning is what is being considered here, and the basic tasks which form the foundation for a task-level planning system could be said to be these:

1. Given a point in the workspace, is it possible to find a configuration of the arm such that the end-effector can reach that point?

2. Given two points in the workspace, is it possible to find a motion between them?

3. Given an initial configuration, is it possible to move the robot to a target point in the workspace?

4. Given two configurations of the arm in a space with obstacles, is it possible to move between them?

5. Given a space-curve, is is possible for the robot to follow the curve (approximately or exactly) with its end-effector?

6. For any of the problems above, how many ways are there of doing the particular problem? How can the optimal solution be defined? What algorithms are needed to search for these optimal solutions?

7. If a sequence of motions of the above kind is to be carried out in an arbitrary order, what is the optimal order in which to carry them out?

A number of further areas have been considered by various authors. Some of these are concerned with objective-level planning, while others are problems motivated by a more immediate engineering need.

1. Coordinated motion of more than one robot arm.

2. Motion of a robot arm amidst moving obstacles.

3. Motion of the robot arm when the obstacles are changing shape, either predictably or randomly.

4. The compliant[1] motion problem—motion of the robot arm so that the end-effector *follows exactly* a given path in the workspace (for example a robot which needs to cut a pattern out of cloth).
5. The adaptive motion problem—a robot which moves objects around must subsequently avoid the objects in their new positions.
6. The exploratory problem—a robot adapting to new information about its surroundings as time progresses.
7. Finding smooth paths—i.e. putting restrictions on the curvature of paths.
8. The problem of allocating tasks dynamically. The robot is carrying out some kind of work, and occasionally new tasks are introduced. The robot must schedule these tasks in an efficient way.

## 1.2 Previous approaches to the problem.

Numerous approaches to the solution of these problems have been attempted, and work continues on finding new ideas. Previous ideas may be summarized as follows:

1. Geometrical methods:
   (a) Precalculating the configuration space.
   (b) Working directly with the workspace.
   (c) Voronoï diagrams, skeleton graphs, et cetera. (finite methods)
2. Methods based on classical mechanics.
3. Cell decomposition methods, including quadtrees/octrees.
4. Approximations, for example by convex hulls.
5. Potential field approaches.
6. Heuristic methods for coping with special situations.
7. Learning methods, for example neural nets and genetic algorithms.
8. Stochastic and "fuzzy logic" methods.

Further details may be found in the survey papers [KT92] and [HA92].

One of the important choices to be made when investigating a robot motion problem is how to represent the environment in which the robot is moving. Several different representations have been used in previous work.

1. Approximating to a grid.

[1] Compliant in this sense means that the robot complies with the given path.

2. Using quadtrees.
3. Polygonal approximation.
4. Constructing objects as sums and differences of standard objects (rectangles, circles, and so on).
5. Giving equations and inequalities for the boundaries.

All of these methods have their advantages and disadvantages. Some are more amenable to heavyweight computational methods (for example the various grid representations) while others, having a more geometric bias, are more amenable to the production of analytic or algebraic solutions.

### 1.3 Open problems.

Many of the problems above remain to be explored. The theory related to robot motion amidst moving objects is still very much in its infancy, and the exploratory and adaptive motion problems lack a fundamental theoretical base. [SS88, KLK93, RS94, SK91, BH92]

Similarly the problem of coordinating the motion of two or more robot arms has had only a little in-depth study [Che91, CX92, Jin94, XMS93]. These papers are concerned with only very simple cases, such as the motion of a two-link arm with a single-link arm, or the motion of two arms constantly gripping an object (this reduces to the problem of moving a 6-bar linked chain) and none of them approach the problem from a more general perspective.

There is also the issue of planning sequences of robot tasks, which combine the above mechanical and geometrical problems with combinatorial optimization. For example a robot may have to execute a sequence of tasks, some of which must be completed before others. What is the optimal motion for the robot between these tasks? Work on this problem requires some ideas from combinatorics and graph theory as well as geometrical notions. This problem becomes much more interesting (and harder) when there are multiple robot arms.

There are also other ways in which the problems can be extended. For example, much of the work up to now has used discrete approximations to obstacle shape, based around grid and quadtree/octree representations. Is it possible to find other representations which preserve more of the geometry of objects? This is an important question as discrete approximations are always going to be inaccurate and intensive computationally, regardless of how delicately they are refined. In particular it would be interesting to find a representation of obstacles which is compact and yet which allows the accuracy to be refined to an arbitrary level. The B-spline representation discussed below satisfies this somewhat.

There is also room for development of more elaborate computational methods. Two areas in which development could proceed are in the use of parallel processing and the creation of dynamic algorithms for the problems of exploratory and adaptive motion planning. Some parallel processing work has already been done [FB92, ZM91, BL91, SG92, KAH92], but this has concentrated more on precalculating the dynamics of the problem rather than actual path-planning.

## 2 Research strategy.

I shall now give an outline of some of the work which I have been doing in recent months, and how I propose to take this work further. The general aim of this work is to develop a number of new methods for the investigation of motion planning problems, avoiding some problems which have dogged previous work and bearing in mind the following principles.

1. To exploit the inherent geometry of the problem to the full.

2. To ensure that the natural symmetries of the problem are not lost in the creation of the mathematical models.

3. To create a hierarchical model in which more complicated robots are modelled by combining a number of simple models.

4. To eliminate the need for human intervention. Some current approaches, such as the potential field methods, need to be supervised in order to avoid certain problems.

5. To avoid any unnecessary calculations which are not relevant to the problem at hand. Most previous approaches rely on the precalculation of "configuration spaces" which outline where the robot is free to move. If the robot has many degrees of freedom, or the obstacles are complicated, this can take a very long time to calculate, and some of these calculations will not be needed for the individual problems at hand.

6. To reduce the difference between "local" and "global" planning. Traditionally there has been a difference between "local" methods, plan the path piece-by-piece, and global methods, which search the whole space. Neither of these are ideal—local methods can miss efficient global solutions, and global methods are often computationally intensive.

This is the general research plan:

1. To model obstacles using rational Bézier and B-spline representations of curves and surfaces. These representations allow a wider range of shapes to be modelled and possess a number of geometric properties which enable the development of efficient algorithms. They also have the advantage that they are the standard representation in CAD systems, so that plans of an environment can be lifted directly from the CAD system to use in the planner.

2. To represent the configuration spaces of robots by multivariate rational B-splines. This will produce a representation which is both efficient computationally and geometrically pricise. The importance of understanding

the inherent geometry of the problem is emphasized in Sharir's paper *Algorithmic Motion Planning in Robotics* [Sha89]:

> Many of the heuristic approaches have been implemented for several years now and have working prototypical systems capable of planning motions in a variety of situations. However, because the problem has such a rich geometric and combinatorial structure, it is imperative that we understand this structure from a mathematically rigorous point of view and explore it for potential algorithmic improvements before beginning to negociate approximating or heuristic shortcuts. Such shortcuts, if taken naively, can completely miss the structure, often resulting in considerable loss of efficiency at best, and, at worst, in a complete misunderstanding of the issues at the heart of a problem.

3. To map the Bézier/B-spline representation of the obstacles into a new kind configuration space, and to create a B-spline representation of the robot's motion. This may be done using similar techniques to recent work in the modelling of mechanisms [GM88, GM89, GM90, GMX]. Through this approach relationship between the configuration space and the workspace of the robot is more intuitive than is present methods. Then subdivision techniques [BBB87, MB90b, MB90a] can then be applied to determine subsets of this configuration space corresponding to configurations in which the robot avoids collisions with obstacles (and itself). Similar methods can be used to compute the free-space available for path planning to any desired level of accuracy. Subdivision techniques are known to be robust and the sub-space can be computed to within any desired tolerance. Furthermore, these techniques have a structure ideal for implementing on a parallel computer.

4. To create a computer simulation incorporating the above algorithm, and to implement this simulation on a parallel computer.

5. To implement the planning algorithms using a real robot.

## 2.1 Details of current work.

Consider a simplified model of a robot arm as a number of thin jointed rods linked by hinges or sliding joints. Call the number of joints in the robot the number of *degrees of freedom*, which we shall denote by $n$ below. As the robot moves around, each of these rods sweeps out a ruled surface in space. For the $m$th rod ($1 \leq m \leq n$) parameterize this space by $m + 1$ parameters, the first $m$ describing the position of the rod in space and the last one describing the distance along the rod. Call this the *interference manifold* of the rod. As the robot moves through space, it creates a ruled surface in each of these interference manifolds, which we call the *motion manifold*.

Using this model it is possible to decide whether the robot will hit itself, by asking whether any two of the motion manifolds intersect. There is a one–one correspondence between points of intersection of these surfaces and two of the rods hitting each other. This can be extended to problems in which two or more robots are working together, to decide whether or not they collide with each other during the execution of a given task..

To consider collisions with the obstacles, we similarly introduce an *obstacle manifold* corresponding to each obstacle, and intersect these with the motion manifolds. Again this easily extends to problems with multiple robots.

To facilitate computation of these intersections B-spline curves and surfaces are used as a universal way of representing all objects within the system, as it is possible to calculate the intersections efficiently using subdivision methods [MB90b, MB90a].

To go beyond representing the robot with thin rods, it is possible to model each joint of the robot by a B-spline surface, and to define the motion of each surface with respect to the joints. Now the motions no longer sweep out ruled surfaces but instead use volumes of revolution and extrusions [Pie90]. Thus broadly similar methods can be used to above, though the computation becomes more expensive (it is still more efficient than other methods used for this problem).

Now that a geometrical model has been set up, it is possible to study more complicated problems. To find the free workspace of the robot it is possible to cut away at the interference manifold at points where the robot and obstacles collide, and where there is self-collision. This provides a rapid means of deciding whether a particular job is feasible within a particular environment. Work continues on using these results to provide efficient algorithms for path planning.

## 2.2 Proposed future work.

The research outlined above outlines a substantial area of work, much of which is still ongoing. It could be developed further by considering the following problems, each of which would both build on work suggested above and break new ground in the subject:

1. To extend the techniques of B-spline and Bézier modelling to the problem of robot motion in an environment of moving obstacles. This would make use of the skills developed above, while adding new levels of mathematical complexity and making a contribution towards a major open problem in robotics.

2. To create models for a robot environment in which obstacles are changing shape. For example a robot working in an agricultural environment will need to be able to work around animals which are constantly moving.

3. To consider ways in which a B-spline model of an environment can be built up interactively, as the robot explores more and more of its surroundings.

This could have connections with work in visual pattern recognition or tactile sensing, both active areas of robotics research.

## 3 Conclusions.

This project is attempting to put the subject of robot motion planning on a firm geometrical basis. Until the mathematical foundations of the subject are fully understood, it will be impossible to be certain that path planning algorithms are efficient, but when the geometry of the subject is explained it should be possible to classify the difficulty of problems in a rigorous way, and develop effective algorithms for the solution of these problems.

## References

[And88] Russell L. Andersson. *A robot ping-pong player: experiment in real-time intelligent control.* MIT Press, 1988.

[BBB87] Richard H. Bartels, John C. Beatty, and Brian A. Barsky. *An introduction to splines for use in computer graphics and geometric modeling.* Morgan Kaufman, 1987.

[BH92] A Bagchi and H Hatwal. Fuzzy logic based techniques for motion planning of a robot manipulator amongst unknown moving obstacles. *Robotica*, 10(6):563–573, 1992.

[BL91] J. Barraquand and J.C. Latombe. Robot motion planning : a distributed representation approach. *International Journal of Robotics Research*, 10(6):628–649, 1991.

[Che91] Y.C. Chen. Solving robot trajectory planning problems with uniform cubic b-splines. *Optimal Control Applications and Methods*, 12(4):247–262, 1991.

[CX92] Y.P. Chien and Q. Xue. Path planning for 2 planar robots moving in unknown environment. *IEEE transaction on systems, man and cybernetics*, 22(2):307–317, 1992.

[Duf80] Joseph Duffy. *Analysis of mechanisms and robot manipulators.* Edward Arnold, 1980.

[FB92] A. Fijany and A Bejczy. *Parallel Computation Systems for Robotics.* World Scientific, 1992.

[GM88] C.G. Gibson and D. Marsh. Concerning cranks and rockers. *Mechanism and Machine Theory*, 23(5):355–360, 1988.

[GM89] C.G. Gibson and D. Marsh. On the linkage varieties of the Watt 6-bar mechanisms. *Mechanism and Machine Theory*, 24(2):106–126, 1989.

[GM90] C.G. Gibson and D. Marsh. On the geometry of geared 5-bar motion. *Journal of Mechanical Design*, 112(4):620–627, 1990.

[GMX] C.G. Gibson, D. Marsh, and Y. Xiang. An algorithm to generate the transition curve of the planar four-bar mechanism. Submitted to *Mechanism and Machine Theory.*

[HA92] Y.K Hwang and N. Ahuja. Gross motion planning—a survey. *ACM Computing Surveys*, 24(3):219–291, 1992.

[HSS87] J. Hopcroft, J.E. Schwartz, and M. Sharir, editors. *Planning, Geometry and Complexity of Robot Motion.* Ablex, Norwood, N.J., 1987.

[Jin94] H.W. Jing. Two arms are faster than one. *International Journal of Robotics Research*, 13(4), 1994.

[KAH92] M. Kameyama, T. Amada, and T. Higuchi. Highly parallel collision detection processor for intelligent robots. *IEEE journal of solid-state circuits*, 27(4):500–506, 1992.

[KLK93] N.Y Ko, B.H. Lee, and M.S. Ko. An approach to robot motion planning for time-varying obstacle avoidance in the presence of obstacles. *Robotica*, 10(4):315–327, 1993.

[KT92] Manja Kirćanski and Olga Timčenko. A geometric approach to manipulator path planning in 3D in the presence of obstacles. *Robotica*, 10(4):321–328, 1992.

[LPW79] T. Lozano-Pérez and M.A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22:560–570, October 1979.

[MB90a] D. Marsh and S.A. Barley. A surface/surface intersection algorithm. Technical Report GMG90/4, University of Birmingham, 1990.

[MB90b] D. Marsh and S.A. Barley. A survey of surface/surface intersection techniques in computer-aided design. Technical Report GMG90/1, University of Birmingham, 1990.

[Pie90] Les Piegl. A toolbox for NURBS, May 1990. Lecture Notes, University of South Florida.

[RS94] J. Reif and M. Sharir. Motion planning in the presence of moving obstacles. *Journal of the ACM*, 41(4):764–790, 1994.

[SG92] J. Solano Gonzalez. *Parallel Computation of robot motion planning algorithms.* PhD thesis, University of Wales (Cardiff), 1992. Number: 42-4875.

[Sha89] M. Sharir. Algorithmic motion planning in robotics. *IEEE Computer*, 22:9–20, March 1989.

[SK91] R. Srikant and K. Krithivasan. Fastest path across constrained moving rectilinear obstacles. *Information Processing Letters*, 37(6):349–353, 1991.

[SS88] J.T. Schwartz and M. Sharir. A survey of motion planning and related geometric algorithms. *Artificial Intelligence*, 37:157–169, 1988.

[Win72] Terry Winograd. *Understanding natural language.* Edinburgh University Press, 1972.

[XMS93] Q. Xue, A.A. Maciejewski, and P.C.Y. Sheu. Determining the collision-free joint space graph for 2 cooperating robot manipulators. *IEEE transactions on systems man and cybernetics*, 23(1):285–294, 1993.

[ZM91] A.M.S. Zalzala and A.S. Morris. Distributed robot control on a transputer network. *IEE proceedings series E—computers and digital techniques*, 138(4):169–176, 1991.