Path Planning and Trajectory Planning Algorithms: A General Overview

Alessandro Gasparetto, Paolo Boscariol, Albano Lanzutti and Renato Vidoni

Abstract Path planning and trajectory planning are crucial issues in the field of Robotics and, more generally, in the field of Automation. Indeed, the trend for robots and automatic machines is to operate at increasingly high speed, in order to achieve shorter production times. The high operating speed may hinder the accuracy and repeatability of the robot motion, since extreme performances are required from the actuators and the control system. Therefore, particular care should be put in generating a trajectory that could be executed at high speed, but at the same time harmless for the robot, in terms of avoiding excessive accelerations of the actuators and vibrations of the mechanical structure. Such a trajectory is defined as *smooth*. For such reasons, path planning and trajectory planning algorithms assume an increasing significance in robotics. Path planning algorithms generate a geometric path, from an initial to a final point, passing through pre-defined via-points, either in the joint space or in the operating space of the robot, while trajectory planning algorithms take a given geometric path and endow it with the time information. Trajectory planning algorithms are crucial in Robotics, because defining the times of passage at the via-points influences not only the kinematic properties of the motion, but also the dynamic ones. Namely, the inertial forces (and torques), to which the robot is subjected, depend on the accelerations along the trajectory, while the vibrations of its mechanical structure are basically determined by the values of the jerk (i.e. the derivative of the acceleration). Path planning algorithms are usually divided according to the methodologies used to generate the geometric path, namely:

A. Gasparetto (✉) · P. Boscariol
DIEGM – Dipartimento di Ingegneria Elettrica Gestionale E Meccanica,
University of Udine, Via Delle Scienze, 206, 33100 Udine, UD, Italy
e-mail: alessandro.gasparetto@uniud.it

A. Lanzutti
MBP, Via Toscanini, 48/B, 46043 Castiglione Delle Stiviere, MN, Italy

R. Vidoni
Faculty of Science and Technology,
Free University of Bozen-Bolzano Piazza Università,
39100 Bolzano, Italy

G. Carbone and F. Gomez-Bravo (eds.), *Motion and Operation Planning of Robotic Systems*, Mechanisms and Machine Science 29,
DOI 10.1007/978-3-319-14705-5_1

- *roadmap* techniques
- *cell decomposition* algorithms
- *artificial potential* methods.

The algorithms for trajectory planning are usually named by the function that is optimized, namely:

- *minimum time*
- *minimum energy*
- *minimum jerk.*

Examples of hybrid algorithms, which optimize more than a single function, are also found in the scientific literature. In this chapter, the general problem of path planning and trajectory planning will be addressed, and an extended overview of the algorithms belonging to the categories mentioned above will be carried out, with references to the numerous contributions to this field.

Keywords Path planning · Trajectory planning · Roadmap · Cell decomposition · Artificial potential · Minimum time · Minimum energy · Minimum jerk

1 Introduction

Human activity in many sectors is nowadays supported or substituted by robots, which range from standard robots for industrial applications to autonomous robots for complex tasks, such as space exploration. Indeed, the great versatility and flexibility of robots allows them to be employed in different sectors, to perform even very diverse tasks. Referring to the industrial environment, a robot can be defined [78] as a mechanical structure made of several rigid bodies (links) connected one to another by means of joints. Within the robot, it is possible to identify a structure that implements mobility, a wrist which provides dexterity, and an end-effector which performs the task given to the robot.

Regardless of the specific mechanical structure, in all types of applications a generic task is achieved by a robot by imposing a specific motion to the end-effector. This motion may be free or bound: the former case applies if the end-effector does not have a physical interaction with the environment, while the latter case applies if the end effector interacts with the environment by exchanging forces and/or torques.

The input of the control system of the robot is generally given by the law of motion, which is generated by a dedicated module for motion planning. Such motion planning module can operate off-line, by using a knowledge of the robot and the environment which is given *a priori*, or can operate on-line: in this case, suitable sensors must be employed to monitor the robot motion and enable the control system to adjust the movements in real time.

Ultimately, controlling the robot means determining the forces and torques that the actuators must develop at the joints, so as to ensure that the reference trajectories are properly followed. However, this problem turns out to be very complex, because a robot is an articulated structure, so the motion of a single link arm affects the other links. Mathematically, this is expressed by the fact that the dynamic equations of a robot (with the exception of Cartesian structures), contain some terms due to the coupling effects between different links.

In most cases, robot controllers are based on closed loops, driven by the error between the reference and the actual position, which allows to achieve the accuracy required to the robot in executing the planned trajectory. In the case that, during a manipulation task, there is contact between the end-effector and the environment, the control problem is further complicated because not only the motion, but also the forces exchanged in the interaction should be monitored and controlled.

In this chapter, we will focus on the path planning and trajectory planning problems, which constitute the two main parts of the general motion planning problem. The interest for such topics is dramatically increasing, because operations at high speed are required to robots in the modern automatic systems; hence, smooth motions should be planned (where *smooth* means that such motions must avoid excessive values of accelerations of the actuators, as well as vibrations of the mechanical structure).

Many algorithms have been proposed, both for path planning and for trajectory planning, in the scientific literature of the robotic domain. The aim of this chapter is to provide a general overview of such algorithms, which have been subdivided into suitable categories.

2 Path Planning

Path planning is a merely geometric matter, because it is defined as the generation of a geometric path, with no mention of any specified time law. On the other hand, trajectory planning consists in assigning a time law to the geometric path. In most cases, path planning precedes trajectory planning; however, these two phases are not necessarily distinct; for instance, if point-to-point trajectories are considered (i.e. only the initial and final positions are specified), the two problems may be solved at the same time.

In this section the analysis of available works in literature deals with the case of systems without non-holonomic constraints.

Different types of paths are possible, depending on the specific case. For instance, for industrial manipulators, the standard path is usually defined by the geometry of the task, which is defined in a static way. In more advanced applications, or for robots operating in dynamic environments, some extra features, such as the need for automatic obstacle avoidance, may be added.

In applications of advanced robotics, the problem of path planning is definitely very challenging, especially for robots characterized by a large degree of autonomy or for robots that must operate in hostile environments (space, underwater, nuclear, military, etc.).

The definition of the path planning problem is very straightforward: "find a collision-free motion between an initial (start) and a final configuration (goal) within a specified environment". The simplest situation is when the path is to be planned in a static and known environment; however, more generally, the path planning problem can be formulated for any robotic system subject to kinematic constraints, in a dynamic and unknown environment.

Much work can be found the robotic literature, dealing with path planning. The first definitions and algorithms date back to the 1970s. In [57] a complete overview of the path planning techniques can be found. An overview of many techniques cited in this work can be found also in the classic book [23] or in the recent book [48]. Other useful reviews of path planning techniques are [49, 55].

Some basic definitions are needed to introduce the path planning problem, namely: the configuration space (*C-space*), the space of free configurations (*C-free*) and the obstacles' representation in the C-space (*C-obs*).

The configuration space is the space of all possible robot configurations, where a configuration q is the specification of position and orientation of the robot A with respect to a fixed reference frame F_W. Referring to Fig. 1, the C-space of the robot A is $\boldsymbol{R}^3$, since the configuration of A is specified by the origin of F_A with respect to F_W, and by its orientation.

For an articulated robot (Fig. 2), the C-space is given by its joint space (in this case, $\boldsymbol{R}^2$). The C-obs is given by the image of the obstacles in the C-space, and the C-free is defined as {C-space—C-obs}.

Path planning algorithms are usually divided in three categories, according to the methodologies used to generate the geometric path, namely:

- *roadmap* techniques
- *cell decomposition* algorithms
- *artificial potential* methods.

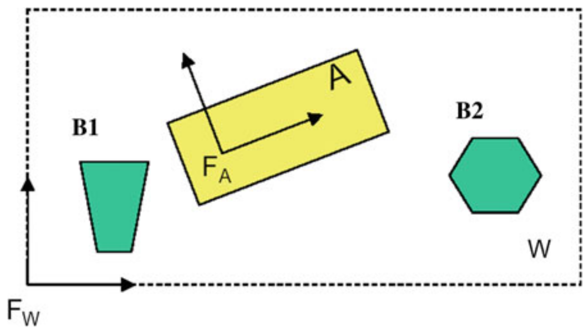

Fig. 1 Mobile robot in a 2-dimensional space with obstacles

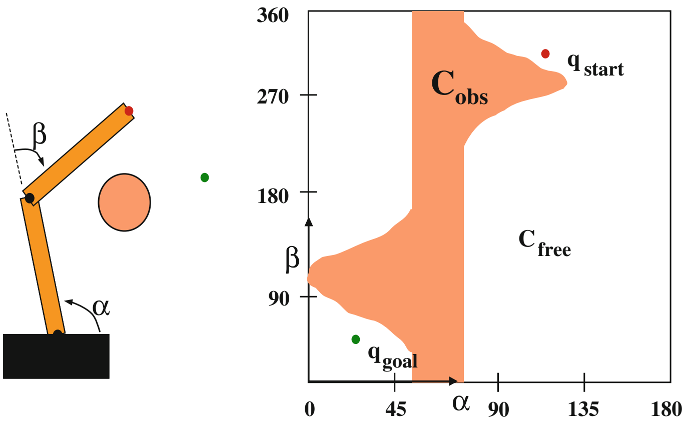

Fig. 2 C-space, C-free and C-obs for an articulated robot with two joints

2.1 Roadmap Techniques

The roadmap techniques are based upon the reduction of the N-dimensional configuration space to a set of one-dimensional paths to search, possibly on a graph.

In other words, this approach maps the free space connectivity into a system of one-dimensional curves (the roadmap) in the C-free space or in its closure. The roadmap R thus obtained contains a set of paths: hence, the path planning consists in linking the initial and final configurations to R. In this way a feasible path between the two configurations is found.

It is very natural to associate a graph to the roadmap and to define some optimality index (e.g. the Euclidean length): the graph can then be searched in order to get the optimal solution to the path planning problem (in most cases, this is represented by the shortest path).

Figure 3 represents the so-called *visibility graph*, i.e. the graph whose nodes are the vertices of all the obstacles in the configuration space. Searching the graph would lead to get the shortest Euclidean path in the C-space. The nodes of the graphs indicate point locations, while edges represents visible connections between the nodes. Grey areas indicate obstacles to be avoided. The concept of visibility graph, which represents a milestone in the literature related to path planning, was introduced by Lozano-Pérez [63, 64].

Another kind of roadmap algorithms are those based on *Voronoi diagrams*, which are defined as a way to divide the space into regions having the following characteristic: given a set of points $\{p_1, \ldots p_n\}$, each point belonging to the ith region is closer to p_i than to any other $p_j \neq \mathrm{p}_i$. This approach is dual to that based on the visibility graph, because the Voronoi diagrams enable one to obtain a path lying at the maximum distance from the obstacles, whereas the visibility graph generates a path that passes as close as possible to the obstacle vertices.

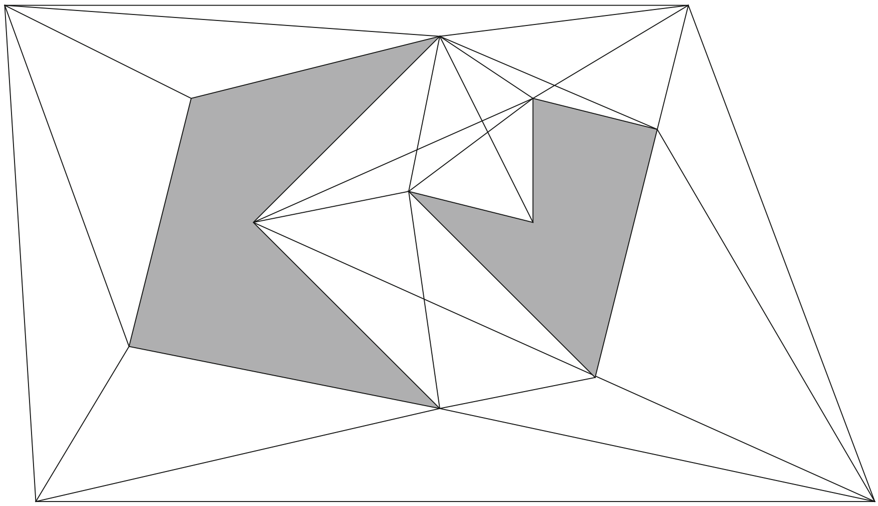

Fig. 3 Visibility graph

Figure 4 shows some path generated by using Voronoi diagrams. The three squares in the diagram represents obstacles, while the blue lines are the set of points equidistant from at least two obstacles. Therefore the paths defined with this technique are designed to be as far away as possible from nearby obstacles. Examples of path planning algorithms may be found in [15, 35, 84].

2.2 Cell Decomposition Methods

According to the cell decomposition methods, the free space of the robot is subdivided into several regions, called cells, in such a way that a path between any two configurations lying in the same cell is straightforward to generate. It is then natural to define a so-called *connectivity graph*, which represents the adjacency relations between cells. Namely, the nodes of the graph represent the cells extracted from the free space, and there is an arch between two nodes are connected if and only if the corresponding cells are adjacent. The path planning problem is, again, turned into a graph searching problem, and can therefore be solved using graph-searching techniques.

Figure 5 illustrates the procedure described above, which is named *exact cell decomposition* technique, because the union of the cell represents exactly the free space. In some cases, an exact computation of the free space is not possible or convenient. *Approximate cell decomposition* methods must therefore be employed. Figure 6 shows how these techniques work:

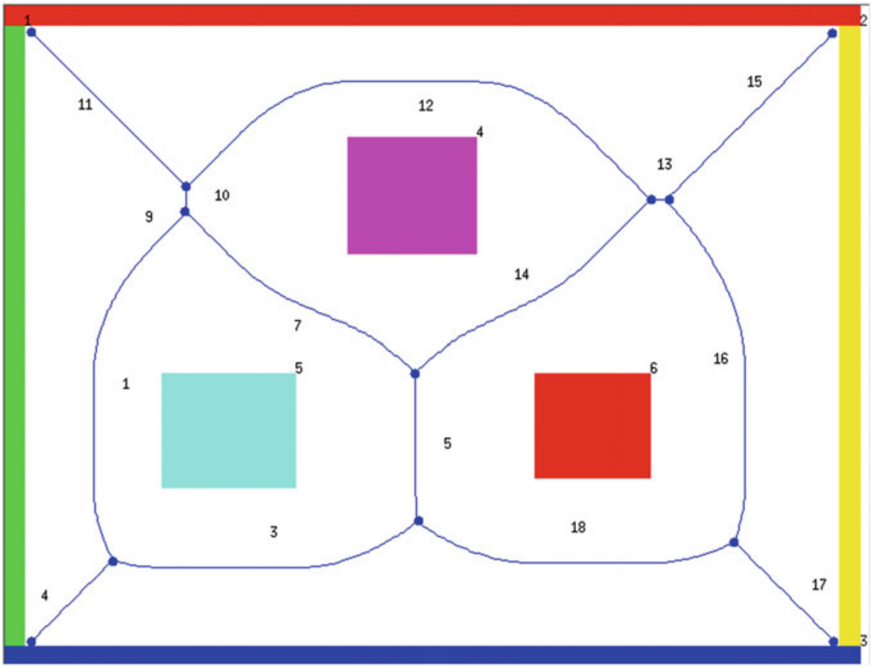

Fig. 4 Paths resulting from Voronoi diagrams

- the whole C-space (assumed 2-dimensional) is divided into four cells;
- the algorithm checks if each cell is completely empty, completely full or mixed (such words obviously refer to the occupancy by the obstacles);
- each mixed cell is in turn divided into four subcells, and the algorithm is recursively applied to check the status of every subcell and recursively divide each mixed subcell into four sub-subcells.

The graph that may be naturally associated to the approximate cell decomposition is a tree, named *quadtree* for 2-dimensional spaces (Fig. 7), *octree* for 3-dimensional spaces (Fig. 8), 16-*tree* for 4-dimensional spaces, and so forth.

2.3 Artificial Potential Methods

The artificial potential methodologies are a different approach to the path planning problem. The basic idea is to consider the robot in the configuration space as a moving point subject to a potential field generated by the goal configuration and the obstacles in the C-space: namely, the target configuration produces an attractive potential, while the obstacles generate a repulsive potential. The sum of these two contribution is the total potential, which can be seen as an artificial force applied

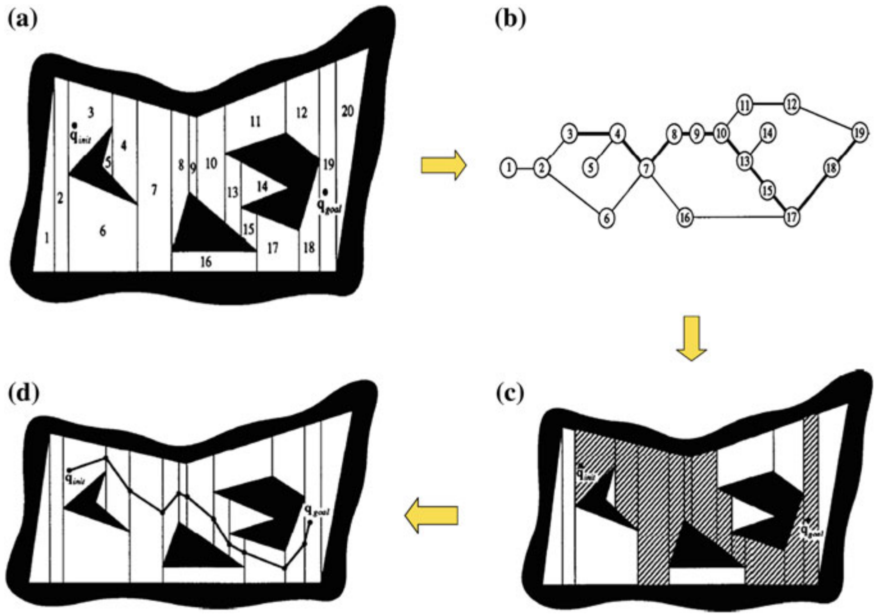

Fig. 5 Exact cell decomposition: **a** subdivision of space into numbered polygons, **b** connectivity graph, **c** regions to be crossed, **d** path

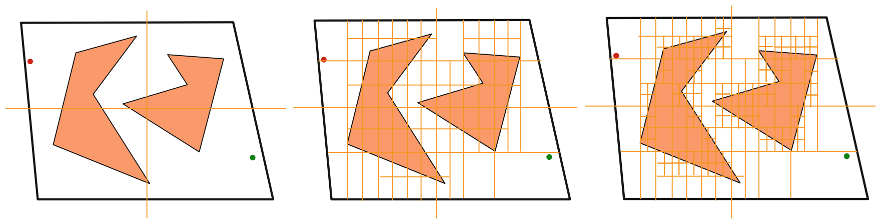

Fig. 6 Approximate cell decomposition

to the robot, aimed at approaching the goal and avoiding the obstacles. Thus, given any configuration during the robot motion, the next configuration can be determined by the direction of the artificial force to which the robot is subjected. This normally represents the most promising direction of motion in terms of free path. An example of the application of the artificial potential method is shown in Fig. 9.

The artificial potential method was originally conceived by Khatib [50] and further developed by Volpe [91, 92]. Such a technique can find applications in many fields, because it can be successfully implemented online, thus moving the obstacle avoidance problem from the higher (and slower) level of path planners to the lower (and faster) level of online motion controllers. This implies that the good features of the artificial potential methods, especially the reactivity to environment changes,

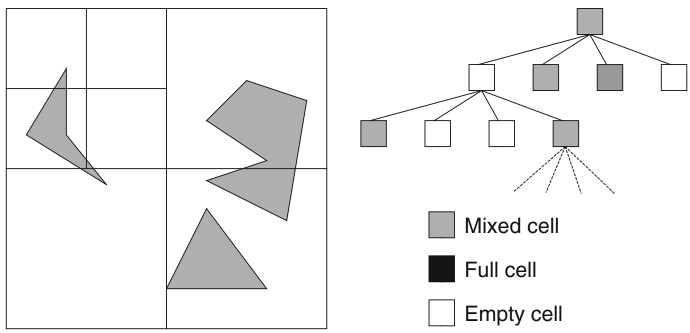

Fig. 7 Quadtree

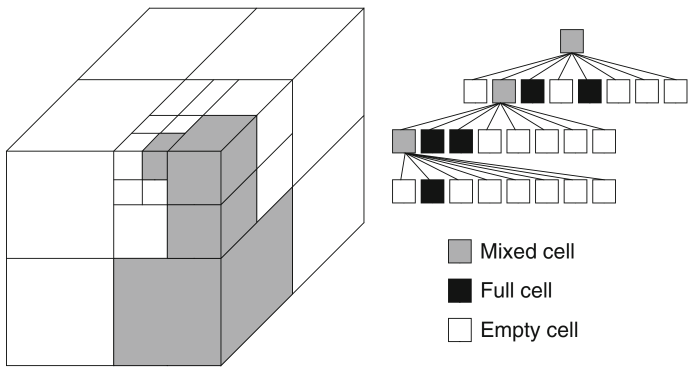

Fig. 8 Octree

duly detected by the robot sensors, enable the robot controller to manage unexpected workspace changes in a fast way.

However, the artificial potential methods are intrinsically affected by a major problem, namely the presence of local minima, where the robot may find itself trapped. In order to overcome this problem, several solutions have been proposed: for instance, using potential functions which do not have local minima [25, 26, 51, 53]. Such functions are called *navigation functions*.

In [39, 42] alternative applications of the artificial potential method are presented.

Another approach to solve the path planning problem is found in [5], where a special kind of planners, named RPP (Random Path Planners), is proposed: local minima are avoided by combining the concepts of artificial potential field with random search techniques. Albeit with some limitations, RPP proved to be able to solve path planning problems for robots with a high number of degrees of freedom, with reasonable computation times.

Other examples of RPP can be found in [18–21].

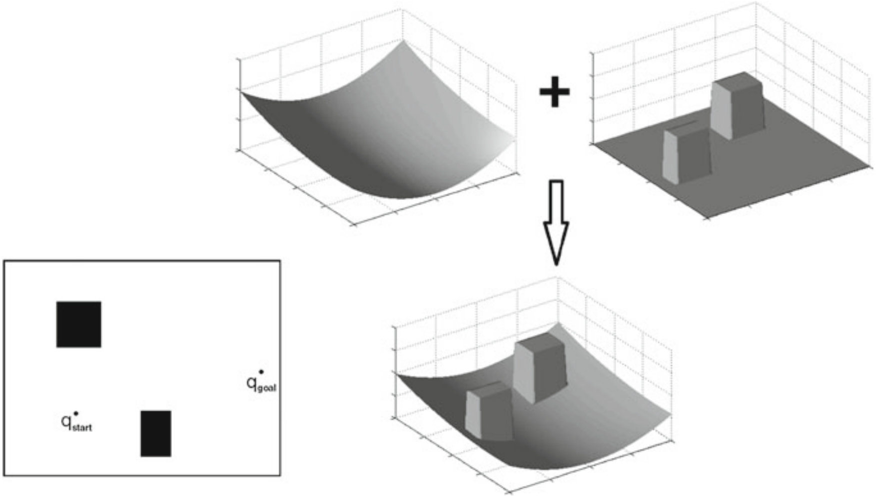

Fig. 9 The artificial potential method

2.4 Alternative Approaches to Path Planning

A possible alternative approach, which had remarkable results in very complex path planning problems, is given by the Probabilistic Roadmap Planners (PRM). It is a technique which employs probabilistic algorithms, such as random sampling, to build the roadmap. The most important advantage of PRM is that their complexity do not strictly depend on the complexity of the environment and on the dimension of the configurations space. The basic idea is to consider a graph where the nodes are given by a set of random configurations in the C-free. A local planner can then try to connect these configurations by means of a path: if a path is found, a new node is added to the graph. In this way the graph reflects the connectivity of the C-free. In order to find a path between two configurations, these configurations are added to the graph, then a graph search is performed in order to find a feasible path. Given the probabilistic nature of the algorithm, post-processing is often necessary to improve the quality of the path. PRM algorithms have been successfully applied to robotic manipulators with up to 16 degrees of freedom. Examples of PRM can be found in [1, 24, 45, 66].

There are some examples [29, 34] of path planners that take into account kinematic and dynamic constraints of the robot, in addition to the pure geometric problem of obstacles avoidance. This problem is referred to as kinodynamic motion planning. Kinodynamic and nonholonomic motion planning can be handled by the Rapidly-exploring Random Tree (RRT) method [58]. This method allows to search non-convex high-dimensional spaces by randomly building a space-filling tree.

Another important version of the general problem is given by path planning in presence of mobile obstacles. As it can be easily understood, this kind of problem results very complex with respect to the basic version. This approach is used, for instance, in [32, 33].

A general overview of the path planning problem can be found in [54] and in [43], where the most important results achieved in the field of path planning, including PRM and RPP techniques, are reported. In [43] it is claimed that all the methodologies that have proven to be practically usable for path planning are based on a discretization of the configuration space. There are two crucial requirements in order to ensure an efficient implementation of path planning methodologies, namely: the efficiency of collision detection algorithms and the efficiency of graph searching techniques.

3 Trajectory Planning

Solving the trajectory planning problem means generating the reference inputs for the control system of the robot, so as to ensure that the desired motion is performed. Usually, the algorithm employed for trajectory planning takes as inputs the path generated by the path planner, as well as the kinematic and dynamic constraints of the robot. The output of the trajectory planning module is given by the trajectory of the joints, or of the end-effector, in form of a sequence of values of position, velocity and acceleration.

The geometric path is normally defined in the operating space of the robot, because the task to be performed, as well as the obstacles to avoid, are described in the operating space more naturally than in the joint space. Thus, planning the trajectory in the operative space means generating a sequence of values that specify the position and orientation that the end-effector of the robot must assume at every time interval. Planning the trajectory in the operating space is usually done when the motion follows a path with specific geometric characteristics defined in the operating space; in this case, the path can be specified in an exact form (i.e. taking the original path), or in an approximate form, by allocating some path points and connecting them by means of polynomial sequences. However, in most cases the trajectory is planned in the joint space of the robot because, since the control action on the manipulator is made on the joints, planning in the operating space requires a kinematic inversion to transform the end-effector position and orientation values into the joint values.

In order to plan a trajectory in the joint space, first a sequence of via-points should be extracted from the desired end-effector path, then a kinematic inversion is to be performed to get the corresponding values of the robot joints. The trajectory is then generated in the joint space by means of interpolation functions, taking into account the kinematic and dynamic limits imposed to the robot joints (in terms of position, velocity, acceleration and jerk). Normally, this way of planning the trajectory can also avoid the problems involved in moving near singular configurations, and can efficiently deal with the possible presence of redundant degrees of mobility. The main drawback of planning a trajectory in the joint space is given by the fact that the execution of a motion planned in the joint space is not so straightforward to predict in the operative space, due to the nonlinearities introduced by the direct kinematics. However, no matter if the trajectory is planned in the operating space or in the joint space, it is crucial that the laws of motion resulting from the planning do not generate forces and torques at the joints that are not compatible with the given constraints: in this way the possibility of exciting mechanical resonance modes can be greatly reduced. For this reason, the planning algorithms must output smooth trajectories, i.e. trajectories represented by a curve whose derivatives are continuous up to a certain order. In particular, it is highly desirable to ensure the continuity of the accelerations of the joints, in order to get trajectories with a limited jerk, because limiting the jerk is crucial in order to reduce the vibrations induced to the robot (which may lead to considerable wear of the mechanical structure), as well as to avoid the excitation of the resonance frequencies of the robot. The vibrations caused by non-smooth trajectories may seriously damage the actuators and degrade the tracking performance of the trajectory. Furthermore, low-jerk trajectories can be executed faster and with a higher accuracy as demonstrated in [6]. In addition, there are some applications where abrupt motions can jeopardize the quality of the work or constitute a risk to the human operators working near the robot.

In order to classify the different trajectory planning methodologies into categories, it is useful to consider that a trajectory is usually planned after some optimality criterion has been set. The most significant optimality criteria that can be found in the literature are:

- minimum execution time;
- minimum energy (or actuator effort);
- minimum jerk.

In addition to the above, hybrid optimality criteria have been proposed, such as, for instance, time-energy optimal trajectory planning. With respect to the minimum energy criterion, a short clarification is necessary. In most of the cases related with trajectory planning, the term "energy" does not correspond to a physical quantity measured in Joules, but it is defined as the integral of squared torques: in other words, it measures the effort of the robot actuators. However, in the robotic literature it is possible to find also trajectory planning algorithms where the optimality index is "energy" in its strict meaning. Actually, this is not really a problem, because in the electric motors used on the robots, the torque can be assumed proportional to the current, so there is a correlation between the actuators' effort and the energy required to the system.

3.1 Minimum Execution Time Algorithms

The optimality criterion based on minimum execution time was the first to be considered in trajectory planning, because short execution times are strictly related to high productivity in automatized production plants in industrial environments. Thus, no wonder that many papers can be found, in the robotic literature, proposing trajectory planning algorithms aimed at minimizing the performance index given by the execution time.

The algorithms described in [7, 80] are defined in the position-velocity phase plane. The basic idea of these algorithms is to write the dynamic equation of manipulator in a parametric form using the curvilinear abscissa s of the path as the independent parameter. The curvilinear abscissa s (path parameter) and its derivative s' pseudo-velocity) constitute the state of the system, while the second derivative of s (i.e. the pseudo-acceleration s'') is chosen as the control variable. In this way, it is possible to transform the constraints given by the nonlinear robot dynamics, as well as the constraints on the actuators, into constraints on the control variable depending from the state of the system. For every point on the path, the maximum admissible value for the pseudo-velocity of the end-effector is determined from the constraints; it is then possible to build in the position-velocity phase plane (i.e. in the (s, s') plane), a velocity limit curve (VLC). The optimal trajectory is then computed by finding the admissible control that yields, for each point of the path, the maximum velocity that does not exceed the limit curve. The solution turns out to be in the form of a curve (named switching curve) in the phase plane.

An alternative approach to minimum time trajectory planning consists in using dynamic programming techniques, such as those described in [2, 81]. The basic idea is to take the state space and discretize it by building a grid of points (called state points). On the basis of the limits set on velocity, acceleration and jerk, it is possible to associate to each point the set of the subsequent admissible state points, and to define the cost of each possible solution by considering the time needed for the motion. This cost is defined by assuming a constant value of acceleration for each step. Finally, an algorithm based on dynamic programming generates the minimum time trajectory. Compared with the phase plane methods, the dynamic programming methods do not require the parameterization of the path and enables to choose an arbitrary performance index. Therefore, such algorithms may be used as a general technique for trajectory optimization. On the other hand, the phase plane approach turns out to be very efficient in terms of computational load; moreover, it may also be used for on-line trajectory planning, as in [28, 67].

A model-based approach is used to maximize the speed of industrial robots by obtaining the minimum-time trajectories that satisfy various constraints commonly given in the application of industrial robots in [52]. Conventional trajectory patterns, such as trapezoidal velocity profiles and cubic polynomial functions.

The algorithms described above produce trajectories with discontinuous accelerations and joint torques, because the dynamic models used consider the robot members as perfectly rigid and do not take into account the actuator dynamics. Neglecting the link flexibility and the actuator dynamics normally leads to some undesired effects. First, in reality the robot actuators cannot generate discontinuous torques: this causes the joint motion to be delayed with respect to the reference trajectory. This accuracy in trajectory following is thus greatly reduced, and the tracking controller has to be often activated during the execution of the trajectory. Moreover, each switching of the actuators may cause the so-called chatter phenomenon, i.e. high frequency oscillations inducing vibrations of the mechanical structure of the robot. This obviously results in wearing of the mechanical components and in a decrease of the accuracy in trajectory following. Again, the tracking controller is activated more frequently and the actuators are further stressed. Another undesired effect resulting from an inaccurate model is that, since the time-optimal control requires saturation of at least one robot actuator at any time instant, it is impossible for the controller to correct the tracking errors arising from disturbances or modelling errors.

In [26, 27] a possible solution to these kind of problems is proposed: in these works, the phase plane method is used, together with a limitation set on the torque variations (actuator jerks). The proposed algorithm takes the pseudo-jerk, defined as the third derivative of the curvilinear abscissa, as the control variable: a dynamic equation of the third order is thus obtained. The experimental results presented in [26] show that, if some upper bound is set on the pseudo-jerk, time-optimal trajectories can be practically obtained by simply employing a conventional PID controller. This proves the correlation between accuracy in trajectory following and low values of jerk.

A different way to limit the torque variations is to consider in the objective function not only the execution time, but also an energy contribution: for instance in [79] the integral of squared torques along the whole trajectory is taken into account. The experimental results presented in [79] show that the increase of the overall motion time is compensated by a greater accuracy in trajectory following, even if conventional PD controllers are used. This results in a reduction of actuator stresses, with obvious advantages in the total lifetime of the electro-mechanical components of the robot.

It is possible to approach the problem of minimum-time trajectory planning by defining *a priori* the primitives of the motion, i.e. the curves that define the trajectory in the joint space. Such curves must be smooth functions, so that the control signals and, consequently, the torque signals at the actuators, result also smooth functions. The most common situation is that in which the path is specified using a limited number of via-points: the solution is then given by spline interpolation. In the literature, several methodologies are proposed to compute time-optimal trajectories for robot manipulators based on optimization of splines, whose order may be three (cubic splines) or higher. The main differences among these techniques are:

- the type of constraints considered (either kinematic or dynamic);
- the algorithm used to compute the optimal trajectory;
- the possibility to extend the optimization problem, by taking into account other optimization criteria, in addition to the minimum time.

The distinction based on the type of constraints can be considered the most important. It can be extended to any type of trajectory planning algorithm, so that the two categories of kinematic trajectory planning and dynamic trajectory planning can be defined. The kinematic trajectory planning algorithms take as their input upper (sometimes also lower) bounds on velocity, acceleration and jerk. In most cases such bounds are considered constant. The dynamic trajectory planning algorithms consider the dynamic model of the robot and define an optimization problem taking into account dynamic constraints, such as bounds on the actuator torques, or on the actuator jerks, defined as the variation of the torques. In some cases kinematic constraint (typically the velocity) are also considered. Both approaches have pros and cons: the kinematic trajectory planning has its main advantage in the simplicity and in the lower computational load; on the other hand, the dynamic trajectory planning features a better capacity to use the robot actuators. In other words, the kinematic methods are based on a simplified computational model that yields a non-optimal use of the robot actuators, although in most cases reasonably good trajectories are planned. Dynamic methods are based on a more accurate model and therefore produce better solutions, but at the cost of a heavier computational load, since they have to deal with non-trivial issues, such as identification of the dynamic parameters of the robot, or the efficiency in implementing efficient algorithms to solve the robot dynamic equations.

An interesting example of an algorithm based on the inverse dynamic of a parallel robot is given by [17]. In this work, a multi-objective optimisation problem is formulated and a dedicated genetic algorithm is employed to find an optimal trajectory based upon spline functions.

Splines function are therefore used as trajectory primitives in order to ensure the continuity of the acceleration. Another example can be found in [59], where a nonlinear optimization problem is set, namely the computation of the value of the time intervals between the via-points, so as to minimize the total execution time of the trajectory subject to kinematic constraints. The technique is based upon an unconstrained optimization algorithm named FPS (Flexible Polyhedron Search), in combination with an algorithm called FSC (Feasible Solution Converter), which converts the solutions that are not physically feasible (i.e. that are not compatible with the kinematic constraints) into feasible ones, by implementing a suitable time scaling of the trajectory generated by the FPS algorithm. In [93], the same optimization algorithm presented in [59] is used, but instead of cubic splines, cubic B-splines are taken as primitives of motion.

The algorithms described above produce a local optimal solution, while other minimum-time trajectory planning methods output a global optimal solution. Piazzi and Visioli use interval analysis to calculate a minimum-time trajectory subject to kinematic constraints at the joints. Such kinematic constraints are on the maximum value of velocity, acceleration and jerk. In [71] they extend the results already presented in [71, 72]. The simulations presented in [71] showed an improvement of 18 % of the total execution time with respect to the results yielded by a local optimization algorithm.

In [40, 41] a global optimization method is presented, which combines a stochastic technique, such as a genetic algorithm, with a deterministic procedure based on interval analysis. The proposed technique can be applied to solve general global optimization problems where semi-infinite constraints are defined. In [40] this algorithm is applied to the problem of minimum-time trajectory planning with specific kinematic and dynamic constraints: namely, the trajectories, represented by cubic splines, are subject to restrictions on the maximum actuator torques, as well as on the linear and angular velocities of the end-effector in the operating space. It is remarkable that, differently from usual, in [40] the velocity constraint is not imposed in the joint space, but in the operating space of the end-effector.

A composition of polynomial functions of different orders are used in [11, 12] to obtain jerk continuity along a trajectory planned from a set of pre-defined via-points, obtaining a global minimum time solution.

Another example of minimum-time trajectory planning for robotic manipulators can be found in [16]. In this case the objective function is made of two terms: the first term takes the squared values of the optimization variables (i.e. of the time intervals between the via-points), while the second term is the sum of the squared accelerations computed at the via- points. The introduction of this second term has the effect of increasing the trajectory smoothness with respect to a pure minimum-time approach. The optimization is performed by using the DFP (Davidon-Fletcher-Powell) algorithm, which does not consider the kinematic bounds, therefore performing an unconstrained minimization. The solution obtained by means of the DFP algorithm is then subjected to a procedure of time-scaling, until the more restrictive kinematic bound has been saturated. The resulting trajectory, although respecting the limits on velocity, acceleration and jerk, is sub-optimal with respect to time.

In [30] a technique for determining time-optimal path-constrained trajectories subject to velocity, acceleration and jerk constraints, acting on both the robot actuators and on the task to be executed, is presented. The solution of the optimization problem is based upon a hybrid optimization strategy, which takes into account the path description, the kinematic model of the robot and constraints defined by the user. The resulting trajectories are optimal with respect to time, but not with respect to smoothness.

In the work [60] a combination of spline functions up to the seventh order are used together to achieve minimum time solutions with velocities, acceleration and jerk bounds. Other examples of minimum-time algorithms subject to kinematic constraints may be found in [31, 49, 85, 86, 89]. In [74] the minimum-time trajectory problem is solved under kinematic and dynamic constraints, i.e. teorque, power, jerk and energy, taking into account both the robot dynamics and the obstacle presence.

3.2 Minimum Energy Algorithms

As already remarked, the minimum-time trajectory planning algorithms received a lot of consideration in the robotic literature, mainly because of the strong industrial interest to reduce the length of the production cycles. However, the minimum-time optimization criterion is not the only one that can be considered: other criteria are definitely more suitable for different needs and requirements.

The trajectory planning based on energetic criteria is interesting under many aspects. On one hand, it generates smooth trajectories which are easier to track, and reduce the stresses induced to the actuators and to the mechanical structure of the robot. On the other hand, this optimization criterion enables one to better comply with energy saving requirements, which are driven not only by mere economic considerations, but may be imposed by specific applications in which the energy source is limited by technical factors, such as robotic applications for outer space, for underwater exploration or for military tasks.

A classical example of minimum-energy trajectory planning algorithm is contained in [65], where a trajectory is optimized with respect to energy taking into account constraints on the motion of the end-effector, as well as the physical limits of the joints. The proposed objective function is the integral of squared torques. The trajectories are expressed by cubic B-splines and, by exploiting some property of the convex hull, it is possible to transform the joint limits into some limits set on the optimization parameter, which are the control points of the B-splines. The resulting motion thus minimizes the effort of the actuators.

In [2, 79] some techniques for optimal trajectories planning, with respect to energy and time, are described: the function to optimize is made of two terms, the first related to the execution time, the second related to the energy consumption. Such algorithms are intended to reduce the stresses of the actuators and to facilitate the trajectory tracking. In [79], the integral of the squared torques along the trajectory is considered in the objective function, while in [2] the function of total energy is considered.

Other examples of optimized trajectories, with respect to energy as well as to time, are presented in [75–77, 90, 95]. In [75] the Authors consider a trajectory parameterized by cubic splines, subject to kinematic constraints set on the maximum value of velocity, acceleration and jerk, and to dynamic constraint given by the maximum torque applicable to the joints. In [76] the same Authors consider a trajectory parameterized by cubic B-splines, where the physical limits of the joints are added to the torque and kinematic constraints. The objective function includes also an additional term (penalty function), in order to avoid mobile obstacles expressed as spherical or hyperspherical safety zones. In [77], two strategies for offline 3-dimensional optimal trajectory planning of industrial robots, in presence of fixed obstacles, are presented. In [90], a nonlinear change of variables is employed to convert the time-energy optimal trajectory planning problem into a convex control problem based on only one state variable. In [95], a methodology based on the minimization of an objective function which considers both the total execution time and the total energy spent along the whole trajectory is presented; the via-points of the trajectory are interpolated by means of cubic splines. Kinematic and dynamic constraints, in terms of upper bounds on velocity, acceleration, jerk and input forces and torques are also considered. It is worth noting that in algorithms such as the one presented in [79] the energy term is added in order to produce trajectories which result slower but smoother with respect to those generated by minimum-time trajectory planning algorithms; on the other hand, in approaches such as the one presented in [76] the objective function is primarily designed to minimize the energy and to plan trajectories with no regard to the execution time.

Recently, due to the development and installation of energy recovery and redistribution devices in robotic systems, the minimum-energy topic has gained new interest among the research community, e.g. [44, 68].

3.3 Minimum Jerk Algorithms

The importance of generating trajectories that do not impose discontinuities of the actuator torques at the robot joints has already been remarked; for instance, in [26] and in [27] this result is obtained by imposing upper bounds to the rate of change of the actuator torques. However, this kind of approach requires the computation of the third order dynamics of the robot.

An alternative method to obtain smooth profiles of the actuator torques is based on the idea of limiting the jerk, defined as the time derivative of the acceleration. Indeed, the torque variations depend upon the dominant term of the matrix of inertia multiplied by the vector of the joint jerk. Thus, some trajectory planning methods take the jerk as the variable to be minimized, in order to obtain smooth trajectories. The minimization of the jerk yields positive results, such as: reduction of the error during the trajectory tracking phase, reduction of the excitation of resonance frequencies, reduction of the stresses induced to the mechanical structure of the robot and to the actuators.

This results in a natural and coordinated motion: indeed, some studies suggest that the movements of the human arm satisfy an optimization criterion based upon the minimization of the jerk, or of the torque variations [82]. The minimum-jerk trajectory planning for robotic manipulators are an example of optimization based on physical criteria which mimic the human ability to produce natural movements [8].

In [56] the analytical solution of a trajectory planning problem for a point-to-point path, based on a minimum-jerk optimization criterion, is presented. The optimization, performed by applying Pontryagin's principle, involves two objective functions, namely: the maximum absolute value of jerk (minimax approach) and the time integral of the squared jerk.

In some cases, the total execution time of the trajectory is not imposed, so it can be chosen so as to comply with the kinematic limits on velocity and acceleration. However, most of the minimum-jerk algorithms that can be found in the robotic literature consider an execution time imposed *a priori*.

In [82], the integral of the squared jerk is minimized along the executed trajectory. In order to have a trajectory with a smooth start and stop, the values of velocity, acceleration and jerk are set to zero at the first and at the last via-points. The proposed algorithm is based upon a stochastic optimization technique performed by means of neural networks. The algorithm does not ensure the exact interpolation of intermediate nodes, but allows a tolerance, which can be set by tuning appropriate weights. This does not constitute a problem in cases where the exact interpolation is not needed, but just the passage in the neighbourhood of the via-points is required. The main limitation of this technique is that the resulting trajectories are not analytical functions, but are numerically defined.

Another approach is contained in [83], where the interpolation of the via-points is performed by means a trigonometric spline, thus ensuring the continuity of the jerk. The algorithm assumes that the time interval between the via-points is known and constant, and takes as input the values for the velocity, the acceleration and the jerk, at the first and at the last via-points (such values are typically all set to zero). There are some advantages in using trigonometric splines to interpolate the trajectory via-points, for instance the property of locality: namely, if a via-point is changed, it is not necessary to recalculate the whole trajectory, but only the two splines that are connected to the via-point need to be recomputed. This property allows fast computation, thus making it possible to implement obstacle avoidance procedures in real time. The most significant aspect, in terms of trajectory optimization, is that parameterizing the trajectory allows some degrees of freedom, namely those given by the values of the first three derivatives (velocity, acceleration and jerk) at the intermediate via-points. Such values can be adjusted in order to minimize an objective function, such as the time integral of the squared jerk. The optimization presented in [83] is not bounded, since no kinematic limits are imposed, and yields a closed form solution, thus not requiring iterative minimization procedures.

In [70, 73] an algorithm based on interval analysis is presented. This technique seeks the minimum of the maximum absolute value of the jerk along a trajectory whose execution time is imposed *a priori*. It is therefore a so-called *minimax* approach bounded on the trajectory execution time. The trajectories primitives are cubic splines and the intervals between the via-points are computed, so as to obtain the lowest maximum absolute jerk value. In [70] the Authors present a comparison with the method based on trigonometric splines [83], reporting the highest values of the jerk, of the torques and of the torque variations. The simulation, which calculates the robot dynamics using the MatLab™ Robotics Toolbox, highlights the efficiency of the minimax algorithm with respect to other approaches.

3.4 Hybrid Optimization Approaches

Optimal trajectory planning with respect to time, energy and jerk has been discussed in the foregoing. Hybrid optimization approaches have also been proposed in the robotic literature. For instance, in order to get the advantages of the jerk reduction while executing fast trajectories, hybrid time-jerk optimal techniques are proposed, for instance [9, 11, 36–38, 46, 69]. These algorithms differ from the primitives used to interpolate the path, or from the optimization procedures implemented.

In [9, 11, 36–38] a minimum time-jerk trajectory planning technique is described, based upon two algorithms aimed at the minimization of an objective function, which is designed so as to ensure fastness in execution and smoothness of the trajectory at the same time. Such an objective function is composed of a term which is proportional to the total execution time and of a term which is proportional to the integral of the squared jerk along the path. The proposed algorithm enables one to define constraints on the robot motion before the execution of the trajectory. The constraints are expressed in form of upper bounds on the velocity, acceleration and jerk values of all robot joints. In this way, any physical limitation of the real robot can be taken into account when planning the trajectory. Unlike most jerk-minimization methods, this technique does not ask for an *a priori* setting of the total execution time.

In [61, 62], the methodology is extended by taking into account also the power consumption of the actuators and physical limits of the joints. In this way, the technique becomes a time-jerk-energy planning algorithm.

Several objectives are taken into account in the work [51]: in particular minimum elctrical and kinematic energy, minimum time and maximum maniuplability are obtained with the solution of a single optimization problem.

Minimum effort trajectories planned trough model-based approaches are presented in [10, 14]. The first one includes bounds on jerk, while second one has bounded joint speed. The work [13] introduces the novel topic of robustness in trajectory planning algorithms. Such approach allows to increase the tolerance of the resulting trajectory to the inevitable mismatches between the dynamic model used for the planning and the actual robot dynamics.

The problem of finding minimum time-effort trajectories for motor-driven parallel platform manipulators, subject to the constraints imposed by the kinematics and dynamics of the manipulator structure is the topic of the paper [21]. Computational efficiency is obtained trough a hybrid scheme comprising the particle swarm optimization method and the local conjugate gradient method. Also in [22] a constrained multi-objective genetic algorithm (MOGA) based technique is proposed to address this problem for a general motor-driven parallel kinematic manipulator. The planning process is composed of searching for a motion ensuring the accomplishment of the assigned task, minimizing the traverse time, and expended energy subject to various constraints imposed by the associated kinematics and dynamics of the manipulator.

All the trajectory planning methods introduced above are applicable to rigid link robot, with either serial or parallel kinematic configurations. However, it is worthwhile to mention that also cable-driven robots application are gaining a growing interest in robotics. Among the advantages brought by this class of manipulators, low overall mass and high stiffness make them very advantageous in many applications. On the other hand, the fact that they often require to use actuation redundancy and that they operation must avoid cable interference [94], has led to the development of trajectory planning algorithms specifically designed for them. The work [87] presents a method to compute trajectories for underconstrained parallel robot that ensures positive and bounded cable tension, while in [88] a similar procedure is also experimentally validated. A detailed study of the dynamics of cable-driven parallel robot is reported in [47], as a tool for developing accurate path planning algorithms. The time-optimality of trajectories designed for cable-driven robot is the topic covered in the works [3, 4].

4 Conclusions

In this paper, the fundamental problems of path planning and trajectory planning in Robotics have been addressed. An overview of the most significant methods, that can be found in the robotic literature to generate collision-free paths, has been presented. Then, the problem of finding an optimal trajectory given a planned path has been discussed and the most significant approaches have been described.

References

1. Amato NM, Wu Y (1996) A randomized roadmap method for path and manipulation planning. In: Proceedings of the 1996 IEEE international conference on robotics and automation, pp 113–120
2. Balkan T (1998) A dynamic programming approach to optimal control of robotic manipulators. Mech Res Commun 25(2):225–230
3. Bamdad M (2013) Time-energy optimal trajectory planning of cable-suspended manipulators. Cable-driven parallel robots. Springer, Berlin, pp 41–51
4. Barnett E, Gosselin C (2013) Time-optimal trajectory planning of cable-driven parallel mechanisms for fully-specified paths with g1 discontinuities. In: ASME 2013 international design engineering technical conferences and computers and information in engineering conference. American Society of Mechanical Engineers
5. Barraquand J, Latombe JC (1991) Robot motion planning: a distributed representation approach. Int J Robot Res 10(6):628–649
6. Barre PJ, Bearee R, Borne P, Dumetz E (2005) Influence of a jerk controlled movement law on the vibratory behaviour of high-dynamics systems. J Intell Robot Syst 42(3):275–293
7. Bobrow JE, Dubowsky S, Gibson JS (1985) Time-optimal control of robotic manipulators along specified paths. Int J Robot Res 4(3):554–561
8. Bobrow JE, Martin BJ, Sohl G, Wang EC, Kim J (2001) Optimal robot motion for physical criteria. J Robot Syst 18(12):785–795
9. Boscariol P, Gasparetto A, Lanzutti A, Vidoni R, Zanotto V (2011) Experimental validation of minimum time-jerk algorithms for industrial robots. J Intell Robot Syst 64(2):197–219
10. Boscariol P, Gasparetto A (2013) Model-based trajectory planning for flexible link mechanisms with bounded jerk. Robot Comput Integr Manuf 29(4):90–99
11. Boscariol P, Gasparetto A, Vidoni R (2012) Jerk-continous trajectories for cyclic tasks. In: Proceedings of the ASME 2012 international design engineering technical conferences (IDETC), pp 1–10
12. Boscariol P, Gasparetto A, Vidoni R (2012) Planning continuous-jerk trajectories for industrial manipulators. In: Proceedings of the ESDA 2012 11th biennial conference on engineering system design and analysis, pp 1–10
13. Boscariol P, Gasparetto A, Vidoni R (2013) Robust trajectory planning for flexible robots. In: Proceedings of the 2013 ECCOMAS multibody dynamics conference, pp 293–294
14. Boscariol P, Gasparetto A, Vidoni R, Romano A (2013) A model-based trajectory planning approach for flexible-link mechanisms. In: Proceedings of the ICM 2013—IEEE international conference on mechatronics, pp 1–6
15. Canny J, Donald B (1988) Simplified voronoi diagrams. Discret Comput Geom 3(1):219–236
16. Cao B, Dodds GI (1994) Time-optimal and smooth constrained path planning for robot manipulators. In: Proceedings of the 1994 IEEE international conference on robotics and automation, pp 1853–1858
17. Carbone G, Ceccarelli M, Oliveira PJ, Saramago SF, Carvalho JCM (2008) An optimum path planning for Cassino parallel manipulator by using inverse dynamics. Robotica 26(2):229–239

18. Caselli S, Reggiani M (2000) ERPP: an experience-based randomized path planner. In: Proceedings of the ICRA'00—IEEE international conference on robotics and automation, pp 1002–1008
19. Caselli S, Reggiani M, Rocchi R (2001) Heuristic methods for randomized path planning in potential fields. In: Proceedings of the 2001 IEEE international symposium on computational intelligence in robotics and automation, pp 426–431
20. Caselli S, Reggiani M, Sbravati R (2002) Parallel path planning with multiple evasion strategies. In: Proceedings of the ICRA'02—IEEE international conference on robotics and automation, pp 260–266
21. Chen CT, Liao TT (2011) A hybrid strategy for the time-and energy-efficient trajectory planning of parallel platform manipulators. Robot Comput-Integr Manuf 27(1):72–81
22. Chen CT, Pham HV (2012) Trajectory planning in parallel kinematic manipulators using a constrained multi-objective evolutionary algorithm. Nonlinear Dyn 67(2):1669–1681
23. Choset HM, Lynch KM, Hutchinson S, Kantor GA, Burgard W, Kavraki LE, Thrun S (2005) Principles of robot motion: theory, algorithms, and implementation. MIT Press, Cambridge
24. Clark CM, Rock S (2001) Randomized motion planning for groups of nonholonomic robots. In: Proceedings of the 6th international symposium on artificial intelligence, robotics and automation in space, pp 1–8
25. Connolly CI, Burns JB (1990) Path planning using Laplace's equation. In: Proceedings of the 1985 IEEE international conference on robotics and automation, pp 2102–2106
26. Constantinescu D (1998) Smooth time optimal trajectory planning for industrial manipulators. Ph.D. Thesis, The University of British Columbia, 1998
27. Constantinescu D, Croft EA (2000) Smooth and time-optimal trajectory planning for industrial manipulators along specified paths. J Robot Syst 17(5):233–249
28. Croft EA, Benhabib B, Fenton RG (1995) Near time-optimal robot motion planning for on-line applications. J Robot Syst 12(8):553–567
29. Donald BR, Xavier PG (1990) Provably good approximation algorithms for optimal kinodynamic planning for Cartesian robots and open chain manipulators. In: Proceedings of the sixth annual symposium on computational geometry, pp 290–300
30. Dong J, Ferreira PM, Stori JA (2007) Feed-rate optimization with jerk constraints for generating minimum-time trajectories. Int J Mach Tools Manuf 47(12–13):1941–1955
31. Dongmei X, Daokui Q, Fang X (2006) Path constrained time-optimal robot control. In: Proceedings of the international conference on robotics and biomimetics, pp 1095–1100
32. Fiorini P, Shiller Z (1996) Time optimal trajectory planning in dynamic environments. In: Proceedings of the 1996 IEEE international conference on robotics and automation, pp 1553–1558
33. Fraichard T (1999) Trajectory planning in a dynamic workspace: a state-time space approach. Adv Robot 13(1):74–94
34. Fraichard T, Laugier C (1993) Dynamic trajectory planning, path-velocity decomposition and adjacent paths. In: Proceedings of the 1993 international joint conference on artificial intelligence, pp 1592–1597
35. Garrido S, Moreno L, Lima PU (2011) Robot formation motion planning using fast marching. Robot Auton Syst 59(9):675–683
36. Gasparetto A, Zanotto V (2007) A new method for smooth trajectory planning of robot manipulators. Mech Mach Theor 42(4):455–471
37. Gasparetto A, Zanotto V (2008) A technique for time-jerk optimal planning of robot trajectories. Robot Comput-Integr Manuf 24(3):415–426
38. Gasparetto A, Lanzutti A, Vidoni R, Zanotto V (2012) Experimental validation and comparative analysis of optimal time-jerk algorithms for trajectory planning. Robot Comput-Integr Manuf 28(2):164–181
39. Ge SS, Cui YJ (2000) New potential functions for mobile robot path planning. IEEE Trans Robot Autom 16(5):615–620
40. Guarino Lo Bianco C (2001a) A semi-infinite optimization approach to optimal spline trajectory planning of mechanical manipulators. In: Goberna MA, Lopez MA (eds) Semi-infinite programming: recent advances. Springer, pp 271–297

41. Guarino Lo Bianco C, Piazzi A (2001b) A hybrid algorithm for infinitely constrained optimization. Int J Syst Sci 32(1):91–102
42. Guldner J, Utkin VI (1995) Sliding mode control for gradient tracking and robot navigation using artificial potential fields. IEEE Trans Robot Autom 11(2):247–254
43. Gupta K, Del Pobil AP (1998) Practical motion planning in robotics: current approaches and future directions. Wiley
44. Hansen C, Oltjen J, Meike D, Ortmaier T (2012) Enhanced approach for energy-efficient trajectory generation of industrial robots. In: Proceedings of the 2012 IEEE international conference on automation science and engineering (CASE 2012), pp 1–7
45. Hsu D, Kindel R, Latombe JC, Rock S (2002) Randomized kinodynamic motion planning with moving obstacles. Int J Robot Res 21(3):233–255
46. Huang P, Xu Y, Liang B (2006) Global minimum-jerk trajectory planning of space manipulator. Int J Control, Autom Syst 4(4):405–413
47. Ismail M, Samir L, Romdhane L (2013) Dynamic in path planning of a cable driven robot. Design and modeling of mechanical systems. Springer, Berlin, pp 11–18
48. Jing XJ (2008) Edited by. Motion planning, InTech
49. Kazemi M, Gupta K, Mehrandezh M (2010) Path-planning for visual servoing: a review and issues. Visual servoing via advanced numerical methods. Springer, London, pp 189–207
50. Khatib O (1985) Real-time obstacle avoidance for manipulators and mobile robots. In: Proceedings of the 1985 IEEE international conference on robotics and automation, pp 500–505
51. Kim JO, Khosla PK (1992) Real-time obstacle avoidance using harmonic potential functions. IEEE Trans Robot Autom 8(3):338–349
52. Kim J, Kim SR, Kim SJ, Kim DH (2010) A practical approach for minimum-time trajectory planning for industrial robots. Ind Robot: Int J 37(1):51–61
53. Koditschek DE (1992) Exact robot navigation using artificial potential functions. IEEE Trans Robot Autom 8(5):501–518
54. Kumar V, Zefran M, Ostrowski JP (1999) Motion planning and control of robots. In: Nof Shimon Y (ed) Handbook of industrial robotics, 2nd edn, vol 2. Wiley
55. Kunchev V, Jain L, Ivancevic V, Finn A (2006) Path planning and obstacle avoidance for autonomous mobile robots: a review. Knowledge-based intelligent information and engineering systems. Springer, Berlin, pp 537–544
56. Kyriakopoulos KJ, Saridis GN (1988) Minimum jerk path generation. In: Proceedings of the 1988 IEEE international conference on robotics and automation, pp 364–369
57. Latombe JC (1991) Robot motion planning. Kluwer
58. LaValle SM (2006) Planning algorithms. Cambridge University Press
59. Lin CS, Chang PR, Luh JYS (1983) Formulation and optimization of cubic polynomial joint trajectories for industrial robots. IEEE Trans Autom Control 28(12):1066–1073
60. Liu H, Lai X, Wu W (2013) Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints. Robot Comput-Integr Manuf 29(2):309–317
61. Lombai F, Szederkenyi G (2008) Trajectory tracking control of a 6-degree-of-freedom robot arm using nonlinear optimization. In: Proceedings of the 10th IEEE international workshop on advanced motion control, pp 655–660
62. Lombai F, Szederkenyi G (2009) Throwing motion generation using nonlinear optimization on a 6-degree-of-freedom robot manipulator. In: Proceedings of the 2009 IEEE international conference on mechatronics, pp 1–6
63. Lozano-Pérez T, Wesley MA (1979) An algorithm for planning collision-free paths among polyhedral obstacles. Commun ACM 22(10):560–570
64. Lozano-Perez T (1983) Spatial planning: a configuration space approach. IEEE Trans Comput 100(2):108–120
65. Martin BJ, Bobrow JE (1999) Minimum effort motions for open chain manipulators with task-dependent end-effector constraints. Int J Robot Res 18(2):213–224
66. Nissoux C, Simon T, Latombe JC (1999) Visibility based probabilistic roadmaps. In: Proceedings of the 1999 IEEE international conference on intelligent robots and systems, pp 1316–1321

67. Pardo-Castellote G, Cannon RH (1996) Proximate time-optimal algorithm for on-line path parameterization and modification. In: Proceedings of the 1996 IEEE international conference on robotics and automation, pp 1539–1546
68. Pellicciari M, Berselli G, Leali F, Vergnano A (2013) A method for reducing the energy consumption of pick-and-place industrial robots. Mechatronics 23(3):326–334
69. Petrinec K, Kovacic Z (2007) Trajectory planning algorithm based on the continuity of jerk. In: Proceedings of the 2007 Mediterranean conference on control and automation, pp 1–5
70. Piazzi A, Visioli A (2000) Global minimum-jerk trajectory planning of robot manipulators. IEEE Trans Ind Electron 47(1):140–149
71. Piazzi A, Visioli A (1997b) A cutting-plane algorithm for minimum-time trajectory planning of industrial robots. In: Proceedings of the 36th Conference on decision and control, pp 1216–1218
72. Piazzi A, Visioli A (1997a) A global optimization approach to trajectory planning for industrial robots, In: Proceedings of the 1997 IEEE-RSJ international conference on intelligent robots and systems, pp 1553–1559
73. Piazzi A, Visioli A (1997c) An interval algorithm for minimum-jerk trajectory planning of robot manipulators. In: Proceedings of the 36th Conference on decision and control, pp 1924–1927
74. Rubio F, Valero F, Sunyer J, Cuadrado J (2012) Optimal time trajectories for industrial robots with torque, power, jerk and energy consumed constraints. Ind Robot Int J 39(1):92–100
75. Saramago SFP, Steffen V Jr (1998) Optimization of the trajectory planning of robot manipulators tacking into account the dynamics of the system. Mech Mach Theory 33(7):883–894
76. Saramago SFP, Steffen V Jr (2000) Optimal trajectory planning of robot manipulators in the presence of moving obstacles. Mech Mach Theory 35(8):1079–1094
77. Saravan R, Ramabalan R, Balamurugan C (2009) Evolutionary multi-criteria trajectory modeling of industrial robots in the presence of obstacles. Eng Appl Artif Intell 22(2):329–342
78. Sciavicco L, Siciliano B, Villani L, Oriolo G (2009) Robotics. Modelling, planning and control. Springer, London
79. Shiller Z (1996) Time-energy optimal control of articulated systems with geometric path constraints. J Dyn Syst Meas Control 118:139–143
80. Shin KG, McKay ND (1985) Minimum-time control of robotic manipulators with geometric path constraints. IEEE Trans Autom Control 30(6):531–541
81. Shin KG, McKay ND (1986) A Dynamic programming approach to trajectory planning of robotic manipulators. IEEE Trans Autom Control 31(6):491–500
82. Simon D (1993) The application of neural networks to optimal robot trajectory planning. Robot Auton Syst 11(1):23–34
83. Simon D, Isik C (1993) A trigonometric trajectory generator for robotic arms. Int J Control 57(3):505–517
84. Takahashi O, Schilling RJ (1989) Motion planning in a plane using generalized Voronoi diagrams. IEEE Trans Robot Autom 5(2):143–150
85. Tangpattanakul P, Meesomboon A, Artrit P (2010) Optimal trajectory of robot manipulator using harmony search algorithms. Recent advances in harmony search algorithm. Springer, Berlin, pp 23–36
86. Tangpattanakul P, Artrit P (2009) Minimum-time trajectory of robot manipulator using harmony search algorithm. In: Proceedings of the IEEE 6th international conference on ECTI-CON 2009, pp 354–357
87. Trevisani A (2010) Underconstrained planar cable-direct-driven robots: a trajectory planning method ensuring positive and bounded cable tensions. Mechatronics 20(1):113–127
88. Trevisani A (2013) Experimental validation of a trajectory planning approach avoiding cable slackness and excessive tension in underconstrained translational planar cable-driven robots. Cable-driven parallel robots. Springer, Berlin, pp 23–29
89. Van Dijk NJM, Van de Wouw N, Nijmeijer H, Pancras WCM (2007) Path-constrained motion planning for robotics based on kinematic constraints. In: Proceedings of the ASME 2007 international design engineering technical conference and computers and information in engineering conference, pp 1–10

90. Verscheure D, Demeulenaere B, Swevers J, De Schutter J, Diehl M (2008) Time-energy optimal path tracking for robots: a numerically efficient optimization approach. In: Proceedings of the 10th international workshop on advanced motion control, pp 727–732
91. Volpe RA (1990) Real and artificial forces in the control of manipulators: theory and experiments. The Robotics Institute, Carnegie Mellon University, Pittsburgh, 1990
92. Volpe RA, Khosla PK (1990) Manipulator control with superquadric artificial potential functions: theory and experiments. IEEE Trans Syst, Man, Cybern 20(6):1423–1436
93. Wang CH, Horng JG (1990) Constrained minimum-time path planning for robot manipulators via virtual knots of the cubic B-spline functions. IEEE Trans Autom Control 35(5):573–577
94. Williams RL, Gallina P (2002) Planar cable-direct-driven robots: design for wrench exertion. J Intell Robot Syst 35(2):203–219
95. Xu H, Zhuang J, Wang S, Zhu Z (2009) Global time-energy optimal planning of robot trajectories. In: Proceedings of the international conference on mechatronics and automation, pp 4034–4039