Math.Comput.Sci. (2012) 6:269–279
DOI 10.1007/s11786-012-0123-8

Mathematics in Computer Science

Trajectory Planning in Robotics

**Alessandro Gasparetto · Paolo Boscariol ·
Albano Lanzutti · Renato Vidoni**

Received: 20 February 2012 / Revised: 12 June 2012 / Accepted: 25 June 2012 / Published online: 30 August 2012

Abstract Trajectory planning is a fundamental issue for robotic applications and automation in general. The ability to generate trajectories with given features is a key point to ensure significant results in terms of quality and ease of performing the required motion, especially at the high operating speeds necessary in many applications. The general problem of trajectory planning in Robotics is addressed in the paper, with an overview of the most significant methods, that have been proposed in the robotic literature to generate collision-free paths. The problem of finding an optimal trajectory for a given path is then discussed and some significant solutions are described.

Keywords Trajectory · Planning · Robot · Optimal · Path

Mathematics Subject Classification (2000) 93C85 · 68T40

1 Introduction

Nowadays, human activity in many fields is supported or replaced by robots, ranging from simple robots used for industrial applications to complex autonomous robots for space exploration. A reason for this diffusion is the excellent versatility and flexibility of robots, which makes them suitable to perform different tasks. With reference to industrial robots, a manipulator is defined [1] as a mechanical structure consisting of a set of rigid bodies (arms) interconnected with each other by means of joints. It is the possible to identify a structure that provides mobility, a wrist which gives dexterity, and an end-effector that performs the task for which the robot is used.

A. Gasparetto (✉) · P. Boscariol · A. Lanzutti
Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica,
Universita' di Udine, Via delle Scienze 208, 33100 Udine, Italy
e-mail: gasparetto@uniud.it

P. Boscariol
e-mail: paolo.boscariol@uniud.it

A. Lanzutti
e-mail: albano.lanzutti@gmail.com

R. Vidoni
Faculty of Science and Technology, Free University of Bolzano-Bozen,
Piazza Universita' 5, 39100 Bolzano, Italy
e-mail: renato.vidoni@unibz.it

Regardless of the particular mechanical structure, in all applications the achievement of a generic task by a robot is based on the execution of a specific movement imposed to the end-effector. The movement may be free, if the end-effector does not have a physical interaction with the environment, or bound, if interaction forces are exchanged between the end-effector and the environment.

The laws of motion, generated by a motion planning module, are the reference inputs for the control system of the robot. The motion planning module may operate off-line, by using a priori knowledge of the robot and the environment, or on-line, by employing suitable sensors to adjust the movements in real time.

The problem of robot control may be defined thus: determine the forces and torques that the actuators must develop at the manipulator joints, in order to ensure the implementation of the reference trajectories. This problem is particularly complex, since a manipulator is an articulated structure, and the motion of an arm affects the other arms. Indeed, the motion equations of a manipulator, except in the case of Cartesian structures, contain terms due to dynamic coupling effects between joints. Usually, robot controllers are based on closed loops, based on the error between the reference and the actual position, so as to achieve the accuracy required to the robot in executing the planned trajectory. If a manipulation task requires contact between the end-effector and the environment, the control problem is further complicated by the fact that, in addition to the motion, the forces exchanged in the interaction must also be controlled.

In this paper, we will focus on *trajectory planning*, defined as: generating the reference inputs for the robot control system that ensures the implementation of the desired motion. In the classical scheme, trajectory planning is preceded by *path planning*, which will be defined in the next section.

2 Path Planning

Path planning is a purely geometric matter, since it implies the generation of a geometric path without a specified time law, while the trajectory planning assigns a time law to the geometric path. It should be highlighted that the two phases are not necessarily distinct; for example, in the case of point-to-point trajectories, where only the initial and final positions are specified, the solution of the two problems is usually contextual.

Depending on the application area and on the specific task, different situations may arise. For example, in the case of manipulators used in industrial processes, the standard path is defined by the geometry of the task, so that trajectory planning is focused on finding a time law that meets appropriate specifications. In more advanced applications, or for robots operating in complex environments, some more features, such as automatic obstacles avoidance, may be required.

Path planning is certainly one of the most interesting topics of advanced robotics, which is characterized by a high degree of autonomy of robots for applications in hostile environments (space, underwater, nuclear, military, etc.).

The problem of path planning for a mobile system is defined as: find a collision-free motion between an initial and final configuration (goal). The simplest situation is found when the path is planned in a static and known environment, but the general problem concerns the generation of the motion of at least one robotic system subject to kinematic and dynamic constraints in a dynamic and not known a priori environment.

Much research in Robotics has focused on path planning, starting from more than 30 years ago. Basics of path planning are presented in [2]: the configuration space (*C-space*), the space of free configurations (*C-free*) and the obstacles representation in the C-space are defined, together with the three most important approaches to path planning: road-map, cell decomposition and potential field.

The *road-map* approach is based on the capture of the free space connectivity on a system of 1-dimensional curves (road-map) in the C-free space or in its closure. The so-built road-map R is used as a set of standardized paths. The path planning is then reduced to linking the initial and final configurations to R, with the aim to find a path in R between the two configurations.

The *cell decomposition* methods divide the free space of the robot in regions, called cells, so that it is possible to easily create a path between any two configurations of the same cell. Afterwards a connectivity graph is constructed, which represents the adjacency relations between cells. The nodes of the graph are the cells extracted from the free space. Two nodes are connected by a link if and only if the corresponding cells are adjacent. Finally a research is performed with the aim to identify a sequence of cells (channel) to calculate a continuous free path between initial and final configurations.

A different approach consists of the discretization of the C-space in a dense and regular grid of configurations, finding a free path. As the grid is generally very wide, this approach requires very powerful heuristics used in the path research. The *potential field* method is a possible solution. The idea is to represent the robot in the configuration space as a point moving under the influence of the potential that is produced by the objective configuration and the obstacles: typically, the target configuration generates an attractive potential, while the obstacles are characterized by a repulsive potential. The total potential anti-gradient is seen as an artificial force applied to the robot; for every configuration the direction of the artificial force is the most promising direction of motion in terms of free path. Potential field techniques have found many applications (see [3–5]), in different fields, because they move the obstacle avoidance problem from a high and slow level to a low and faster level, typical of controllers. The properties of the potential field method (i.e. reactivity to environment changes), associated to sensor systems, allow to quickly manage unexpected workspace changes. The crucial problem using artificial potential methods is the presence of local minima, that can stall the robot. A possible solution is to employ potential functions without local minima (navigation functions), see for example [6–9]. For other recent applications of potential field techniques see [10] and [11]. Another solution may be found in [12], where a type of planners called RPP (random path planners) is presented, that combines the concepts of potential field with random search techniques to avoid local minima. Although with some limitations, RPP techniques have proved their ability in solving planning problems for robots with many degrees of freedom with computation times ranging from several seconds to several minutes. Other examples of RPP are found in [13,14] and [15].

An alternative approach that has achieved remarkable results in extremely complex problems is the *probabilistic roadmap planners* (PRMs). It is a methodology that uses probabilistic techniques (random sampling) to construct the roadmap. The great advantage of PRM is that their complexity depends more on the path complexity rather than the complexity of the environment and the size of the configurations space. The general idea is to consider a collection of random configurations in the free space, making the nodes of a graph. First, the number of nodes is chosen, then a local planner tries to connect these configurations with a path. When a solution is found, a new vertex is generated in the graph. Once the graph reflects the connectivity of the C-free, it can be used for specific instances (motion planning queries). To find a motion between two configurations, these are added to the graph using the local planner, then a path that corresponds to the motion required is found. Usually a post-processing correction is necessary to improve the quality of the path. The PRM has been successfully applied to path planning for robotic manipulators with up to 16 degrees of freedom. References [16–19] provide an overview of developments and applications of these techniques.

Some examples of motion planners that consider, in addition to the geometric problem of obstacles avoidance, kinematics and dynamics constraints (kinodynamic motion planning) are available in [17,19,20] and [21]. The problem of mobile obstacles is discussed in [17,21,22] and [23].

In [24] is presented an overview of the problem, and in [25] some results are collected, including PRM and RPP, obtained in the field of motion planning. In [25] it is pointed out that all the techniques that have proven of practical use for motion planning are based on a discretization of the space of configurations. The key to an efficient implementation is given by both the efficiency of collision detection algorithms (to this aim, new obstacle representations have been proposed, such as spherical representation instead of polyhedral), and the efficiency of graph searching techniques.

3 Trajectory Planning

The target of the trajectory planning is to generate the reference inputs for the manipulator control system that ensures the implementation of the desired movement. It can be assumed that a trajectory planning algorithm takes as inputs the geometric path, the kinematic and dynamic constraints of the manipulator; the output is the trajectory of the joints, or of the end-effector, expressed as a sequence of values of position, velocity and acceleration [1].

The geometric path is usually specified in the operative space, since both the task to perform and the obstacles to avoid can be more naturally described in this space. The trajectory planning in the operative space consists to generate a time sequence of values (within the constraints imposed) that specify the position and orientation of the end-effector. This solution is adopted when the movement is on a path with geometric characteristics defined in the operative space; the path can be specified exactly by primitive path or in an approximated way by the allocation of path points that are usually connected with polynomial sequences. Anyway, since the control action on the manipulator is made on the joints, a kinematic inversion is necessary in order to derive the evolution of variables in the joint space.

A trajectory planned in the joint space consists in the acquisition of the values, for each joint of the manipulator, corresponding to the via-points set by the user. The trajectories defined in the joint space are generated by means of interpolation functions which respect the limits imposed. This planning solution can also easily avoid the problems involved in moving near singular configurations and the possible presence of redundant degrees of mobility. The main drawback is related to the fact that the execution of a movement planned in the joint space is not so easy to predict in the operative space, due to the nonlinear effects introduced by the direct kinematics. Regardless of the particular strategy adopted, it is essential that the laws of motion generated in the planning phase are such as induce forces and torques at the joints compatible with the given constraints, hence reducing the possibility to excite mechanical resonance modes, that are most often not modelled. Starting from this fundamental consideration, it is necessary that the planning algorithms output smooth trajectories, i.e. trajectories with a high order of continuity. In particular, it is preferable to ensure the continuity of the accelerations at the joints, in order to obtain trajectories with a limited jerk. Limiting the jerk is crucial to reducing the wear of the manipulator and to decrease the excitation of resonant frequencies of the manipulator. The vibrations induced by non-smooth trajectories can cause damage to the actuators, as well as degrade the tracking performance of the trajectory. Furthermore in many applications the execution of sudden movements can compromise the quality of workmanship or even pose a risk. Moreover, low-jerk trajectories can be performed more accurately and quickly.

There are, in general, different ways to accomplish a given task using a robotic manipulator; a possible solution is to run an optimal motion with respect to some relevant criterion. The optimal trajectory planning for robotic manipulators is a very interesting topic which focuses on generating off-line movements to perform tasks known a priori and in a defined environment. In the scientific literature on the trajectory planning problem it is possible to find different optimality criteria; the most significant are:

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

Besides the aforementioned approaches, some hybrid optimality criteria have also been proposed (e.g. time-energy optimal trajectory planning). By considering the second criterion, a quick clarification is necessary. The term energy, in most situations where it is used in the optimization of path, does not correspond to a physical quantity measured in joules, but it is the integral of squared torques and measures the effort of the actuators. However, in the scientific literature it is possible to find also planning algorithms where the index of optimality is just an energy in the strict sense. Anyway, it should be noted that in the electric motors that are used on the robots, generally the torque is proportional to the current, so even in the first case there is a correlation between the minimization function and the energy required to the system.

4 Minimum Execution Time Algorithms

The trajectories planned using the minimum execution time criterion occupy a prominent place on the economic implications related to increasing productivity in industry. For this reason, it is possible to find a vast scientific literature devoted to the presentation of trajectory planning algorithms that minimize the performance index given by the execution time. The algorithms described in [26] and [27] are based on the position-velocity phase plane. The basic idea is to use the curvilinear abscissa θ of the path as a parameter to express the dynamic equation of manipulator in a parametric form. The curvilinear abscissa θ (path parameter) and its derivative $\dot{\theta}$ (pseudo-velocity) are the state of the system, while $\ddot{\theta}$, i.e. the second derivative of θ (pseudo-acceleration) is taken as the control variable. In this way the constraints given by the nonlinear dynamics of the robot and the constraints on the actuators are transformed into constraints on the control that depend from the state. For each point of the path, the constraints determine the maximum admissible value for the pseudo-velocity of the end-effector, and consequently it is possible to build in the plane $(\theta, \dot{\theta})$, called position-velocity phase plane, a velocity curve limit (VLC). The optimal trajectory is calculated by finding the admissible control that produces at each point of the path the maximum velocity that does not violate the curve limit, and the solution is provided in the form of a curve (switching curve) in the phase plane.

An alternative approach is to use dynamic programming techniques [28,29]. The idea is to discretize the state space into a grid of points (state-points). Based on the velocity, acceleration and jerk limits, to each point is associated the set of the subsequent admissible state-points, and the cost of each possible solution is given by the movement time. The cost is calculated assuming at every step a constant value of acceleration. The algorithm based on dynamic programming finally determines the minimum time trajectory. Compared with the phase plane method, the dynamic programming method does not require a parameterized path and can incorporate others performance indices than the minimum time, therefore represents a general methodology of trajectory optimization. However, the phase plane approach is very efficient if the computational load is considered and can be used also for on line trajectory planning algorithms [30,31].

The aforementioned techniques generate trajectories with discontinuous values of accelerations and joint torques, because the dynamic models used for trajectory computation assume the robot members as perfectly rigid and neglect the actuator dynamics [32]. This leads to two undesired effects: first, the real actuators of the robot cannot generate discontinuous torques, which causes the joint motion to be always delayed with respect to the reference trajectory. This reduces the accuracy in trajectory following and actives the tracking controller. Indeed, after each switching the actuators have the so-called chatter phenomenon, i.e. high frequency oscillations. The chatter of the actuators produces mechanical vibrations in the structure of the manipulator robot that cause wear and reduce the accuracy in trajectory following. The result is that the tracking controller is activated more frequently and the actuators are further stressed. Second, the time-optimal control requires saturation of at least one robot actuator at any time instant, so the controller cannot correct the tracking errors arising from disturbances or modelling errors.

A possible solution to these problems can be found in [32] and [33], where the phase plane method is used together with an imposed limitation on the torque variations (actuator jerks).The algorithm considers the third derivative of the curvilinear abscissa (pseudo-jerk) as limited control and requires a dynamic equation of the third order. The experimental outcomes documented in [32] show that the imposition of a limit to the jerk is a prerequisite for the practical realization of time-optimal trajectories using conventional PID by highlighting the correlation between accuracy in trajectory following and low values of jerk. An alternative way to limit the torque variations is to include in the objective function, with the execution time, an energy contribution that in [34] is the integral of squared torques along the path. The experimental results reported in [34] show that the increase in terms of motion time is largely compensated by a greater precision in the trajectory following made by conventional PD controllers, resulting in a reduction of actuator stresses, with obvious advantages in the duration of the electromechanical components. Another example of time-energy optimal trajectory planning is presented in [29].

Another solution is to use smooth functions to characterize the trajectories in the joint space, because the torque generated by the actuators are calculated as continuous functions of the trajectory parameters, thus a smooth trajectory results in smooth control signals. In the common situation in which the path is specified using a limited number of via-points, the solution is given by the spline interpolation. In the scientific literature there are several algorithms for computing time-optimal trajectories for robot manipulators based on optimization of cubic splines. The main differences are:

- the considered type of constraints, which may be kinematic or dynamic;
- the used iterative algorithm;
- the possibility to extend the optimization problem in more general criteria than the minimum time.

The first distinction is certainly the most important, and it is valid for any type of planning algorithm (this allows to distinguish between kinematic trajectory planning and dynamic trajectory planning). The kinematic method for the trajectory planning provides the inclusion of velocity, acceleration and jerk limits, which are usually taken constant. The alternative is to consider the dynamic model of the manipulator and formulate an optimization problem with dynamic constraints, such as the limits of torque and torque variation (actuator jerk), possibly together with kinematic constraints (typically the velocity). The benefits of the two approaches are the computational simplicity of the kinematic algorithms and the better capacity utilization of the manipulator for the dynamic algorithms. Indeed, the kinematic methods consider a simplified computational model that gives rise to an under-capacity utilization of the manipulator but still produces good solutions. Dynamic methods are instead more defined and therefore produce better solutions, but to the detriment of the computational model, since they must deal with non-trivial issues, such as identification of the dynamic model and the implementation of efficient algorithms for calculating the robot dynamics.

In [35], the trajectories are expressed by means of cubic splines, in order to ensure the continuity of acceleration. The resulting nonlinear optimization problem concerns the calculation of the value of the time intervals between the nodes that minimizes the execution time of the trajectory subject to kinematic constraints. The unconstrained optimization method FPS (Flexible Polyhedron Search) is used in combination with a FSC (Feasible Solution Converter) conversion process of the not-physically feasible solutions (i.e. solutions that are not compatible with the kinematic constraints) into feasible solutions. This process is based on the time-scaling of the trajectory. The same optimization algorithm presented in [35], can be used by substituting the cubic splines with cubic B-splines [36].

These types of algorithms produce a local optimal solution, while other optimization methods output a global optimal solution for the minimum time trajectory planning. In [37], extending the results already presented in [38,39], interval analysis (see also [40]) is used to calculate a minimum-time trajectory subject to kinematic constraints at the joints on the maximum value of velocity, acceleration and jerk. The simulation outcomes in [37] reported an improvement of 18 % of the execution time in comparison to a local optimization algorithm. In [41] and [42] a method of global optimization, which combines a stochastic technique (genetic algorithm) with a deterministic procedure based on the analysis interval, is presented. The algorithm is proposed for solving general problems of global optimization with semi-infinite constraints. In [41] this hybrid technique is applied to the problem of minimum-time trajectory planning with dynamic and kinematic constraints. In particular the trajectories, parameterized by cubic splines, are subject to restrictions on the maximum torque of the actuators and on the Cartesian velocity of the end-effector. The velocity constraint, unlike the more traditional approaches, it is not imposed at the level of joint but as a limit on the linear and angular speed of the end-effector. A further example of minimum-time trajectory planning for robot manipulators is presented in [43]. Unlike the optimization algorithms based on cubic splines mentioned above, in this case the objective function is composed of two terms. The first term is quadratic in the optimization variables, which are the time intervals between the via-points; the second term is the sum of the squared accelerations calculated at the interpolation points. The introduction of the second term has the effect of increasing the smoothness of the trajectory with respect to a minimum-time approach. The optimization is done by using the DFP algorithm (Davidon–Fletcher–Powell), that does not take into account the kinematic limits, and then performs an unconstrained minimization. After that, a procedure of time-scaling scales the solution find with the DFP method, until the more restrictive kinematic limit is saturated. The resulting trajectory is therefore time sub-optimal and respects the limits on velocity, acceleration and jerk. In [44] a method for determining time optimal path-constrained trajectories subject to velocity, acceleration and jerk constraints, acting on both the manipulator actuators and on the task to be executed, is presented. The optimization problem is solved using a hybrid optimization strategy, starting from the path description, the kinematic relations of the manipulator and the defined constraints. The resulting trajectories are optimal with respect to time, not with respect to smoothness. Other minimum time algorithms under kinematic constraints (such as the maximum allowed values of velocity, acceleration and jerk) can be found in [45–48].

5 Minimum Energy Algorithms

As noted above, the minimum time trajectory planning is an issue very much discussed in the scientific literature, mainly for industrial interest to reduce the production cycles. However, the criterion of minimum time certainly does not exhaust the possible applications or respond to all needs.

The trajectory planning based on energy criteria has many interesting aspects. On the one hand it produces smooth trajectories that are easier to track, and reduce the stresses to the actuators and the manipulator structure. On the other hand, this method allows energy saving, which is not just a mere economic implication, since this characteristic may be required by specific applications in which the energy source is limited by technical factors, such as robotic applications in outer space, for underwater exploration or for military tasks.

Time-energy optimal trajectories planning methods are described in [29] and [34]: the function to optimize is composed by two terms, the first related to the execution time, the second related to the energy and it is intended to reduce the stresses of the actuators and to facilitate the trajectory tracking. The trade-off between the two criteria can be controlled by two weights. In [34], in the objective function the integral of the squared torques along the trajectory is taken, in [29] the energy is considered.

Other examples of time-energy optimized trajectories are presented in [49–53]. In [49] one may find a trajectory parameterized by cubic splines, subject to kinematic constraints on maximum value of velocity, acceleration and jerk, and with the dynamic constraint given by the maximum torque to the joints. In [50] the trajectory is parameterized by cubic B-splines; 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 security zones. In [51], two strategies for the off-line 3-dimensional optimal trajectory planning of the industrial manipulators in the presence of fixed obstacles are presented. In [52], the time-energy optimal trajectory planning problem is transformed into a convex control problem based on only one state variable, through a nonlinear change of variables. In [53], a technique based on the minimization of an objective function that takes into account both the execution time and total energy along the whole trajectory is presented; the via-points of the path are interpolated by means of cubic splines. The kinematic and dynamic constraints, expressed by means of upper bounds on velocity, acceleration, jerk and input force/torque are also taken into account. It is worth noting that while in approaches such as the one in [34] the energy term is added to produce trajectories a bit slower but smoother than the minimum planning time, in the case of methods such as [50] the objective function is primarily designed to minimize the energy and to plan trajectories where the execution time is not imposed.

In [54] a trajectory is optimized with constraints on the motion of the end-effector. The objective function is the integral of squared torques and the velocity to the first and the last via-point is imposed to zero; the geometric limits of the joints are also taken into account. The trajectories are expressed by cubic B-splines and the property of the convex hull allows to transform the joint limits into the optimization parameter limits, which are the control points of the B-splines. The resulting motion minimizes the effort of the actuators.

6 Minimum Jerk Algorithms

The importance of generating trajectories that do not require discontinuous values of joint torques has already been pointed out; in [32] and [33] such a result is obtained by imposing upper limits to the torques rate of change. This type of solutions requires the computation of the third order dynamics of the manipulator.

An indirect method to obtain profiles of acceptable torque values is based on the idea of limiting the jerk, defined as the time derivative of the acceleration. This is due to the fact that the derivative of the torque vector (i.e. the torque variations) depends from the dominant term of the matrix of inertia multiplied by the vector of joint jerk. Some of the methods mentioned above take into account the limit on the jerk, in addition to achieving a jerk limited trajectories; furthermore can be interesting optimize a performance index as a measure of jerk. The minimization of the jerk produces positive results, such as: reducing the error during the trajectory tracking phase, reducing the excitation of resonance frequencies, limiting the stresses to the manipulator structure and to the actuators, resulting in natural and coordinated movements.

The last result is reported because there are some studies that suggest that the movements of the human arm satisfy an optimization criterion that can be a measure of jerk, or the rate of torque variations [55]. The minimum-jerk trajectories for robotic manipulators are an example of optimization based on physical criteria that mimic the human ability to produce natural movements [56].

In [57] the analytical solutions of a trajectory planning problem for a point-to-point path, based on minimum jerk optimization with the values of velocity and acceleration imposed to zero, are obtained. The optimization, performed by applying Pontryagin's principle, concerns two objective functions:

- the maximum absolute value of jerk (*minimax* approach);
- the time integral of the squared jerk.

If the execution time of the trajectory is not imposed, it can be chosen so that the kinematic limits on velocity and acceleration are observed. Nevertheless, in the event of an assigned movement on a sequence of via-points, most of the minimum-jerk algorithms found in the scientific literature consider an execution time imposed a priori and do not consider the kinematic limits.

In [55], the integral of squared jerk is minimized on the trajectory execution time along the set of via-points. With the aim to ensure a trajectory with a smooth start and stop, the values of the velocity, acceleration and jerk is set to zero in the first and last via-points. This algorithm is focused on a stochastic optimization technique based on neural networks. The algorithm does not guarantee the exact interpolation of intermediate nodes, but implements an interpolation with a tolerance, settable by appropriate weights. This is not a problem in situations where exact interpolation is not required, whereas it is required the passage in the neighbourhood of tolerance centres specified by the via-points. The main limitation of this technique is that the planning of the trajectories are a numerical time functions, and not analytical.

An interesting approach is described in [58], where the interpolation of the via-points is guaranteed by using a trigonometric spline (hence the continuity of the jerk is ensured). The technique assumes that the time interval between the via-points is known and constant, and allows to specify the values for the velocity, the acceleration and the jerk, typically all set to zero, in the first and the last via-points. If trigonometric splines are used to interpolate the trajectory via-points, some advantages can be obtained, such as the property of locality: if a node is changed, it is necessary to recalculate only the two sections of the spline that are connected to the node. Based on this property, it is possible to implement procedures of obstacle avoidance in real time. The most significant aspect in terms of trajectory optimization is that the parameterization has some degrees of freedom, given by the value of the first three derivatives (velocity, acceleration and jerk) at the intermediate nodes, which can be used to minimize a cost function, e.g. the time integral of the squared jerk. The optimization is not bounded, as no kinematic limits are imposed, and presents a closed form solution; therefore iterative minimization procedures are not required.

In [59] and [60] an algorithm based on interval analysis is presented. This method globally minimizes the maximum absolute value of the jerk along a trajectory whose execution time is imposed a priori. Hence, it is a minimax approach bounded on the trajectory execution time. The trajectories are expressed by means of cubic splines and the intervals between the via-points are calculated to produce the lowest maximum absolute jerk value. The paper [60] presents a simulation comparison with the method based on trigonometric spline [58], reporting the highest values of the jerk, of the torques and of the torque variations. The simulation, which calculates the dynamics using the Matlab Robotics Toolbox developed by Corke [61], highlights the better outcomes obtained with the minimax approach.

7 Hybrid Optimization Approaches

Starting from the fundamental optimization techniques above described, hybrid optimality approaches are implemented. With the aim to reach the advantages of the jerk reduction in fast trajectories, hybrid time-jerk optimal techniques are proposed (see for example [58,62–66]). These algorithms use on different primitives to interpolate the path (e.g. trigonometric splines in [58], polynomials of fourth and fifth order in [63]), and different optimization procedures (e.g. genetic algorithms are used in [62], SQP algorithm in [64–66]).

In particular, in [64–66] a minimum time-jerk trajectory planning technique is presented. Two algorithms based upon a minimization of an objective function that takes into account the velocity and the smoothness of the trajectory are described. More in detail, the objective function is composed of a term that is proportional to the total execution time and of a term that is proportional to the integral of the squared jerk along the path, both weighted by two parameters. Furthermore, this algorithm enables one to define constraints on the robot motion before execution. The constraints are expressed as upper bounds on the absolute values of velocity, acceleration and jerk for all robot joints, so that any physical limitation of the real manipulator can be taken into account when planning its trajectory. Moreover, unlike most jerk-minimization methods, this technique does not require an a priori setting of the total execution time. A method based on the objective function defined in techniques above described ([64–66]) and extended by considering also the power consumption of the actuating motors and the joints physical limits (so that the technique is a time-jerk-energy planning algorithm) is presented in [67] and [68]. The analysis of the scientific literature has highlighted the importance of generating trajectories with a high degree of continuity (i.e. smooth), and where the forces produced by the actuators have to comply with the amplitude constraints and not exciting mechanical resonance modes, that often are not modelled.

Moreover, it is better to plan trajectories that require limited rate of torque variations. This can be achieved directly integrating the dynamic model of the manipulator in the trajectory planning algorithm and limiting the rate of torque variations. An indirect way that can be used to obtain comparable results, but with lower computational loads, is to limit the jerk, defined as the derivative of the acceleration at the joints. Whichever approach is chosen, the limitation of the jerk is reported in the scientific literature as a way that can be used in order to reduce tracking errors, the actuator stresses (with advantages of both reliability and the duration of electromechanical components) and the excitation of resonant frequencies.

8 Conclusions

In this paper, the general problem of trajectory planning in Robotics has been addressed, with an overview of the most significant methods, that have been proposed in the robotic literature to generate collision-free paths. The problem of finding an optimal trajectory for a given path has then been discussed and the most significant solutions have been described.

References

1. Sciavicco, L., Siciliano, B., Villani, L., Oriolo, G.: Robotics. Modelling, Planning and Control. Springer, London (2009)
2. Latombe, J.C.: Robot Motion Planning. Kluwer, Norwell (1991)
3. Khatib,O.: Real-time obstacle avoidance for manipulators and mobile robots. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 500–505 (1985)
4. Volpe, R.A, Khosla, P.K: Manipulator control with superquadric artificial potential functions: theory and experiments. IEEE Trans. Syst. Man Cybern. **20**(6), 1423–1436 (1990)
5. Volpe, R.A.: Real and Artificial Forces in the Control of Manipulators: Theory and Experiments. Carnegie Mellon University, The Robotics Institute, Pittsburgh (1990)
6. Koditschek, D.E.: Exact robot navigation using artificial potential functions. IEEE Trans. Robot. Autom. **8**(5), 501–518 (1992)
7. Kim, J.O., Khosla, P.K.: Real-time obstacle avoidance using harmonic potential functions. IEEE Trans. Robot. Autom. **8**(3), 338–349 (1992)
8. Connoly, C.I., Burns, J.B.: Path planning using Laplace's equation. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2102–2106 (1990)
9. Connoly, C.I., Grupen, RA.: On the application of harmonic functions to robotics. In: Proceedings of the IEEE International Symposium on Intelligent Control, pp. 498–502 (1992)
10. Guldner, J., Utkin, V.I.: Sliding mode control for gradient tracking and robot navigation using artificial potential fields. IEEE Trans. Robot. Autom. **11**(2), 247–254 (1995)

11. Ge, S.S., Cui, Y.J.: New potential functions for mobile robot path planning. IEEE Trans. Robot. Autom. **16**(5), 616–620 (2000)
12. Barraquand, J., Latombe, J.C.: Robot motion planning: a distributed representation approach. Int. J. Robot. Res. **10**(6), 628–649 (1991)
13. Caselli, S., Reggiani, M., Sbravati, R.: Parallel path planning with multiple evasion strategies. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 260–266 (2002)
14. Caselli, S., Reggiani M.: ERPP an experience-based randomized path planner. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1002–1008 (2000)
15. Caselli, S., Reggiani, M., Rocchi R.: Heuristic methods for randomized path planning in potential fields. In: Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp. 426–431 (2001)
16. Amato, N.M., Wu, Y.: A randomized roadmap method for path and manipulation planning. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 113–120 (1996)
17. Hsu, D., Kindel, R., Latombe, J.C., Rock, S.: Randomized kinodynamic motion planning with moving obstacles. Int. J. Robot. Res. **21**(3), 233–255 (2002)
18. Nissoux, C., Simon, T., Latombe, J.C.: Visibility based probabilistic roadmaps. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, pp. 1316–1321 (1999)
19. Clark, C.M., Rock S.: Randomized motion planning for groups of nonholomic robots. In: Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space, pp. 1316–1321 (1999)
20. Donald, B.R., Xavier, P.G.: Provably good approximation algorithms for optimal kinodynamic planning for Cartesian robots and open chain manipulators. In: Proceedings of the 6th Annual Symposium on Computational Geometry, pp. 290–300 (1990)
21. Fraichard, T., Laugier, C.: Dynamic trajectory planning, path-velocity decomposition and adjacent paths. In: Proceedings of the 2nd International Joint Conference on Artificial Intelligence, pp. 1592–1597 (1993)
22. Fiorini, P., Shiller, Z.: Time optimal trajectory planning in dynamic environments. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1553–1558 (1996)
23. Fraichard, T.: Trajectory planning in a dynamic workspace: a state-time space approach. Adv. Robot. **13**(1), 74–94 (1999)
24. Kumar, V., Efran, M., Ostrowski, J.: Motion planning and control of robots. In: Handbook of Industrial Robotics. 2nd edn, Shimon, Y. Nof (ed) (1999)
25. Gupta, K., del Pobil, A.P.: Practical Motion Planning in Robotics: Current Approaches and Future Directions. Wiley, West Sussex (1998)
26. Bobrow, J.E, Dubowsky, S., Gibson, J.S.: Time-optimal control of robotic manipulators along specified paths. Int. J. Robot. Res. **4**(3), 554–561 (1985)
27. Shin, K.G., McKay, N.D.: Minimum-time control of robotic manipulators with geometric path constraints. IEEE Trans. Autom. Control **30**(6), 531–541 (1985)
28. Shin, K.G., McKay, N.D.: A dynamic programming approach to trajectory planning of robotic manipulators. IEEE Trans. Autom. Control **31**(6), 491–500 (1986)
29. Balkan, T.: A dynamic programming approach to optimal control of robotic manipulators. Mech. Res. Commun. **25**(2), 225–230 (1998)
30. Croft, E.A., Benhabib, B., Fenton, R.G.: Near time-optimal robot motion planning for on-line applications. J. Robot. Syst. **12**(8), 553–567 (1995)
31. Pardo-Castellote, G., Cannon, R.H. Jr.: Proximate time-optimal algorithm for on-line path parameterization and modification. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1539–1546 (1996)
32. Costantinescu, D.: Smooth Time Optimal Trajectory Planning for Industrial Manipulators. PhD Thesis, The University of British Columbia (1998)
33. Constantinescu, D., Croft, E.A.: Smooth and time-optimal trajectory planning for industrial manipulators along specified paths. J. Robot. Syst. **17**(5), 233–249 (2000)
34. Shiller, Z.: Time-energy optimal control of articulated systems with geometric path constraints. J. Dyn. Syst. Meas. Control **11**(8), 139–143 (1996)
35. Lin, C.S., Chang, P.R., Luh, J.Y.S: Formulation and optimization of cubic polynomial joint trajectories for industrial robots. IEEE Trans. Autom. Control **28**(12), 1066–1073 (1983)
36. Wang, C.H., Horng, J.G.: 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 (1990)
37. Piazzi, A., Visioli, A.: Global minimum-time trajectory planning of mechanical manipulators using interval analysis. Int. J. Control **71**(4), 631–652 (1988)
38. Piazzi, A., Visioli, A.: A global optimization approach to trajectory planning for industrial robots. In: Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems, pp. 1553–1559 (1997)
39. Piazzi, A.,Visioli, A.: 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 (1997)
40. Horst, R., Pardalos, P.M. (eds.): Handbook of Global Optimization. Kluwer, Dordrecht (1995)
41. Guarino Lo Bianco, C., Piazzi, A.: A semi-infinite optimization approach to optimal spline trajectory planning of mechanical manipulators. In: Goberna e, M.A., Lopez, M.A. (eds.) Semi-infinite Programming: Recent Advances (2001)
42. GuarinoLo Bianco, C., Piazzi, A.: A hybrid algorithm for infinitely constrained optimization. Int. J. Syst. Sci. **32**(1), 271–297 (2001)

43. Cao, B., Dodds, G.I.: Time-optimal and smooth constrained path planning for robot manipulators. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1853–1858 (1994)
44. Dong, J., Ferreira, P.M., Stori, J.A.: Feed-rate optimization with jerk constraints for generating minimum-time trajectories. Int. J. Mach. Tools Manuf. **47**(12), 1941–1955 (2007)
45. van Dijk, N.J.M., van de Wouw, N., Nijmeijer, H., Pancras, W.C.M.: Path-constrained motion planning for robotics based on kinematic constraints. In: Proceedings of the ASME IDETC/CIE Conference, pp. 1–10 (2007)
46. Dongmei, X., Daokui, Q., Fang, X.: Path constrained time-optimal robot control. In: Proceedings of the International Conference on Robotics and Biomimetics, pp. 1095–1100 (2006)
47. Tangpattanakul, P., Artrit, P.: Minimum-Time Trajectory of Robot Manipulator Using Harmony Search Algorithm. In: Proceedings of the IEEE 6th International Conference on ECTI-CON, pp. 354–357 (2009)
48. Joonyoung, K., Sung-Rak, K., Soo-Jong, K., Dong-Hyeok, K.: A practical approach for minimum-time trajectory planning for industrial robots. Ind. Robots Int. J. **37**(1), 51–61 (2010)
49. Saramago, S.F.P, Steffen, V. Jr.: Optimization of the trajectory planning of robot manipulators tacking into account the dynamics of the system. Mech. Mach. Theory **33**(7), 883–894 (1998)
50. Saramago, S.F.P, Steffen, V. Jr.: Optimal trajectory planning of robot manipulators in the presence of moving obstacles. Mech. Mach. Theory **35**(7), 1079–1094 (2000)
51. Saravan, R., Ramabalan, S., Balamurugan, C.: Evolutionary multi-criteria trajectory modeling of industrial robots in the presence of obstacles. Eng. Appl. Artif. Intell. **22**, 329–342 (2009)
52. Verscheure, D., Demeulenaere, B., Swevers, J., De Schutter, J., Diehl, M.: 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 (2008)
53. Xu, H., Zhuang, J., Wang, S., Zhu, Z.: Global Time-Energy Optimal Planning of Robot Trajectories. In: Proceedings of the International Conference on Mechatronics and Automation, pp. 4034–4039 (2009)
54. Martin, B.J., Bobrow, J.E.: Minimum effort motions for open chain manipulators with task-dependent end-effector constraints. Int. J. Robot. Res. **18**(2), 213–224 (1999)
55. Simon, D.: The application of neural networks to optimal robot trajectory planning. Robot. Autonom. Syst. **11**, 23–34 (1993)
56. Bobrow, J.E., Martin, B., Sohl, G., Wang, E.C., Park, F.C., Kim, J.: Optimal robot motion for physical criteria. J. Robot. Syst. **18**(12), 785–795 (2001)
57. Kyriakopoulos, K.J., Saridis, G.N.: Minimum jerk path generation. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 364–369 (1998)
58. Simon, D., Isik, C.: A trigonometric trajectory generator for robotic arms. Int. J. Control **57**(3), 505–517 (1993)
59. Piazzi, A., Visioli, A.: An interval algorithm for minimum-jerk trajectory planning of robot manipulators. In: Proceedings of the 36th Conference on Decision and Control, pp. 1924–1927 (1997)
60. Piazzi, A., Visioli, A.: Global minimum-jerk trajectory planning of robot manipulators. IEEE Trans. Ind. Electron. **47**(1), 140–149 (2000)
61. Corke, P.I.: Robotics toolbox for Matlab. Available on http://petercorke.com/RoboticsToolbox.html, 2010
62. Huang, P., Xu, Y., Liang, B.: Global minimum-jerk trajectory planning of space manipulator. Int. J. Control Autom. Syst. **4**(4), 405–413 (2006)
63. Petrinec, K., Kovacic, Z.: Trajectory planning algorithm based on the continuity of jerk. In: Proceedings of the Mediterranean Conference on Control and Automation, pp. 1–5 (2007)
64. Gasparetto, A., Zanotto, V.: A new method for smooth trajectory planning of robot manipulators. Mech. Mach. Theory **42**(4), 455–471 (2007)
65. Gasparetto, A., Zanotto, V.: A technique for time-jerk optimal planning of robot trajectories. Robot. Comput. Integr. Manuf. **24**(3), 415–426 (2008)
66. Gasparetto, A., Lanzutti, A., Vidoni, R., Zanotto, V.: Trajectory planning for manufacturing robots: algorithm definition and experimental results. In: Proceedings of the ASME 10th Biennial Conference on Engineering Systems Design and Analysis, pp. 609–618 (2010)
67. Lombai, F., Szederkenyi, G.: Trajectory tracking control of a 6-degree-of-freedom robot arm using nonlinear optimization. In: Proceedings of the IEEE International Workshop on Advanced Motion Control, pp. 655–660 (2008)
68. Lombai, F., Szederkenyi, G.: Throwing motion generation using nonlinear optimization on a 6-degree-of-freedom robot manipulator. In: Proceedings of the IEEE IInternational Conference on Mechatronics, pp. 1–6 (2009)