

# Survey of optimal motion planning

eISSN 2631-6315 Received on 18th December 2018 Revised 31st March 2019 Accepted on 2nd April 2019 E-First on 28th May 2019 doi: 10.1049/iet-csr.2018.0003 www.ietdl.org

Yajue Yang¹, Jia Pan² ⋈, Weiwei Wan³

<sup>1</sup>Department of Biomedical Engineering, City University of Hong Kong, Hong Kong SAR, People's Republic of China

**Abstract:** Optimal motion planning becomes more and more important these days and is critical for motion planning algorithms developed in the academia to be applicable to real-world applications. The objective of this article is to survey the current state-of-the-art on optimal motion planning algorithms in terms of their three main components: the decision variables and problem formulation, constraints, and objectives. A selection of proposed techniques is reviewed along with a discussion of their effectiveness and limitations. The side-by-side comparison presented in this survey helps to gain insight into the strengths and limitations of the reviewed approaches and assists with system-level design choices.

#### 1 Introduction

Autonomous robots are becoming increasingly important in both industry and everyday life. In industry, rising labour costs are motivating manufacturers to consider using more robots in factories. For example, the average minimum wage in China increased by >20% in 2012, while the supply of manufacturing robots has also increased by 51% in China [1]. Europe and USA exhibit similar trends: autonomous robots are being designed to make workers more productive and make manufacturers more competitive in terms of price and quality.

The tremendous improvement in the design and availability of intelligent robots over the last decade is based on progress in many related areas, including computer vision, artificial intelligence, machine learning, control, sensor systems, and mechanical systems. Compared to traditional industrial robots, one important feature of the modern intelligent robot system is the motion planning, which can be informally described as follows. Given a robot with a description of its dynamics and the environment, and a set of goal states, the motion planning is to find a sequence of control inputs so as to drive the robot from its initial state to one of the goal states while obeying the rules of the environment, e.g. not colliding with the surrounding obstacles. The capability of efficient and safe motion planning is essential for robots to work reliably in dynamic environments along with humans.

The concept of motion planning was first presented in [2], and a large variety of algorithms have been proposed in the past decades. For the detailed history and summarisation of the motion planning, please refer to classical textbooks such as [3–5].

The initial focus of the motion planning research is concentrated on finding a *complete* planning algorithm, where an algorithm is said to be complete if it terminates in finite time, returning a valid solution if one exists, and failure otherwise. Many complete algorithms have been proposed for low-dimensional robots (with state space dimensions  $\leq 3$ ) in the past decades, and the representatives are geometric methods such as bug algorithm [6], visibility roadmaps [7], and their variants. The main idea of these methods is to build a graph structure over the 2D or 3D environment, and then use graph-search algorithms such as  $A^*$  to find the shortest path over the graph either greedily or exhaustively. Unfortunately, the problem of complete planning is known to be very hard from the perspective of computational complexity. For instance, a basic version of the motion planning problem called the piano movers problem is proven to be PSAPCE-hard [8].

Practical planners came around with the development of cell-decomposition methods [9] and potential fields [10, 11]. These approaches relaxed the completeness requirement to the *resolution completeness*, i.e. the ability to return a valid solution if one exists and if the resolution parameter of the algorithm is sufficiently small. These planners demonstrated good performance in complex environments in terms of the success rate and the computational time. However, their practical applications were mostly limited to robots with state spaces up to five dimensions, because the decomposition-based methods suffered from an exponential number of cells, and potential field methods suffered from local optima.

To plan the trajectory for robots with high degrees-of-freedom (DOF), such as the industrial robots (usually six to seven DOFs) and humanoid robots (usually 30 or more DOFs), one main revolution to the motion planning is the development of samplingbased algorithms [12, 13], and representatives are probabilistic roadmaps (PRM) and rapidly exploring random trees (RRT). These methods demonstrated to be very effective for motion planning in high-dimensional spaces and continuously attracted attention over the last decade, including some very recent variants and improvements [14-16]. Instead of using an explicit representation of the environment, sampling-based algorithms rely on a collision checking module to provide information about the feasibility of candidate trajectories and to connect a set of points sampled from the obstacle-free space in order to build a graphic roadmap of feasible trajectories. The roadmap is then used to construct the solution to the original motion planning problem. Informally speaking, sampling-based methods bypass the curse of dimensionality problem by avoiding explicit construction of obstacles in the state space. It provides a relaxed guarantee called the probabilistic completeness in the sense that the probability that the planner fails to return a solution if one exists decays to zero as the number of samples approaches infinity. The rate of decay of the probability of failure is exponential under the assumption that the environment has good 'visibility' properties [12, 17].

The early work on motion planning focused on finding a trajectory that satisfies the constraints imposed by the environment of the application. However, a feasible solution is not sufficient in most applications where the quality of the solution returned by the motion planning algorithm is also important. For instance, in many industrial applications, the users are interested in solution paths with the minimum time to execute, so as to achieve the highest productivity. This requirement results in the problem of optimal

<sup>&</sup>lt;sup>2</sup>Department of Computer Science, The University of Hong Kong, Hong Kong SAR, People's Republic of China

<sup>&</sup>lt;sup>3</sup>Graduate School of Engineering Science, Osaka University, Japan

<sup>⋈</sup> E-mail: jpan@cs.hku.hk

motion planning, i.e. computing motion plans that are of the minimum cost with respect to a given cost functional, such as the length of a path or energy consumed. This problem has been proven to be very challenging even in basic cases [18].

There are some other limitations of the early motion planning work discussed above. First, it is difficult for these approaches to take into account the complex constraints imposed by the robot dynamics, though some early work was able to deal with simple kinodynamic constraints (e.g. the velocity or acceleration bounds) [19, 20]. The lack of *consistency* is another major limitation of the early sampling-based methods, especially for those multi-query algorithms like RRT and its variants [13]. Here, the consistency requires that given two queries with similar initial/goal states, the resulting trajectories should be similar. It is well-known that sampling-based planning is not consistent, which significantly limits their application in industrial application where the predictability of the trajectory is very important. Both these limitations can be handled by the optimal motion planning methods.

Informally speaking, optimal motion planning methods formulate the motion planning task as an optimisation problem, which includes three components:

- a suitable representation of the problem in terms of the choice of decision variables;
- ii. a suitable representation of the constraints;
- iii. a suitable representation of the objectives.

Given the optimisation problem, we can then solve it using either out-of-box optimisation solvers or using specially designed optimisation approaches. Here, we will discuss all the optimal motion planning approaches according to their different choices in terms of the three optimisation components: variables, constraints, and objectives, as summarised in Table 1.

Here, we will discuss three mainstream methods for optimal motion planning, including

- i. grid-based methods with motion primitives and state cells;
- ii. sampling-based methods that are asymptotic optimal;
- iii. trajectory optimisation techniques.

In particular, we will focus on analysing the advantages and disadvantages of these approaches, and whenever possible, provide potential solutions to the limitations of previous optimal motion planning algorithms.

The rest of the paper is organised in the following manner. We discuss the decision variable representation of the optimal motion planning problem in Section 2, investigate different ways to formulate planning constraints in Section 3, and then summarise different ways to describe the objectives in Section 4.

# 2 Decision variables and representation for optimal motion planning

The grid-based optimal motion planning algorithms [21–26] divide the configuration space into finite cells based on a regular structure (often a grid) of some size or resolution and thus are frequently referred to as search-based planning methods. Each cell represents the set of all configuration space states that fall within it. A representative state such as the cell centre is chosen to determine whether a cell is free or not. A graph is constructed by assigning a

**Table 1** Taxonomy of optimal motion planning algorithms in terms of the decision variables, optimisation objectives, and optimisation constraints

Variables	Objectives	Constraints
1. joint positions	1. minimum length	1. collision free
2. joint positions + velocities	2. smoothness	2. pose constraint
3. trajectory segments in terms	3	3
of splines		
4. state cells	_	

vertex to each cell, and by adding edges between vertices that are close in the configuration space and can be connected by a feasible motion. Vertices and edges are only added when they are found to locate in the collision-free region of the configuration space. All edges can have costs associated with them, providing a way to minimise a cost function along the path instead of just finding any path. Using this representation, the optimal motion planning problem is turned into finding a path optimisation problem over the graph. Note that even with a finite number of vertices, the constructed graph can often be too large to fit into the memory. As a result, the graph is often generated in an online and on-demand fashion, where vertices/edges are verified and allocated only when the graph search algorithm research them. In this way, even for problems with an infinite number of states, the search-based method can find the goal by only allocating memory for a small portion of the configuration space. There are many extensions of the grid-based search. For instance, they were extended to run in an anytime fashion [27, 28] to deal with dynamic environments [29], and to handle systems with differential constraints [30]. These have also been successfully demonstrated on various robotic platforms. However, optimality guarantees of these algorithms are only ensured up to the grid resolution. In addition, the number of grid points grows exponentially with the dimensionality of the state space in the worst case, thus the worst-case running time of these algorithms can be slow.

Sampling-based optimal motion planning algorithms can be considered as a variant of the grid-based optimal motion planning, and the main difference is that the vertices are generated by random sampling instead of fixed grid cells. These algorithms avoid explicitly representing configuration space obstacles, which can be computationally expensive. They instead sample vertices and connect them with local paths in the collision-free state space resulting in a graph optimisation problem with the joint configuration as the decision variables. Due to the randomness, the sampling-based approach returns zigzag trajectories which are not acceptable for many applications. Some RRT variants have used different heuristics to improve the path quality but are not provably optimal [31]. One important progress was achieved through the use of random graph theory to rigorously show that with appropriate parameters and an A \* -alike rewiring procedure, roadmap-based approaches can achieve asymptotic optimality [32]. In particular, each new sample must be tested for connection with at least a logarithmic number of neighbours as a function of the total number of nodes using a steering function. The resulting algorithms PRM\* and RRT\* motivated a set of new methods that are asymptotically optimal, including the anytime [33], incremental [34], lazy [35] variants, and some other heuristics [36]. The roadmaps of these asymptotically optimal methods can be slow to construct and quickly grow too large for storage for fast online queries, especially for relatively high-dimensional instances. To overcome this challenge, techniques that guarantee asymptotic nearoptimality using sparse roadmaps have been proposed, including

A summary of the different obstacle representations used in the optimal motion planning algorithms is provided in Table 2.

Trajectory optimisation is an established method in robotics to generate a high-quality path from an initial trajectory that may be in-collision or dynamically infeasible. It has also been used as a post-processing phase to remove redundant or jerky motion in trajectories generated by traditional planning algorithms [5]. Trajectory optimisation works by solving a constrained non-convex optimisation problem over the trajectory space in terms of a sequence of states. The objective usually is the smoothness of the trajectory. Common inequality constraints include obstacle avoidance, joint limits, joint speed limits, inverse singularity etc., and common equality constraints include the end-effector pose (e.g. reaching a target pose at the end of the trajectory) and orientation constraints (e.g. keeping a held object upright). For under-actuated, non-holonomic planning, additional equality constraints are added to ensure consistent kinematics and dynamics. Many approaches use the waypoint-based representation for trajectory with each waypoint being a robot configuration and use cost-gradient information for minimisation. Such methods

**Table 2** Comparison of different optimal motion planning methods in terms of obstacle representations

methods in ten	ns or obstacle	representations	
	Geometric methods	Sampling- based methods	Trajectory optimisation methods
explicit representation of obstacles?	yes;	no;	yes and no;
_	casting geometric information from the workspace to the configuration space	throwing away geometric information	local geometric information is used to generate gradient information
obstacle representation	geometric primitives (polygons, convex shapes,)	no restriction; collision checker is regarded as a black box	geometric primitives (polygons, convex shapes,)
planning result	poor	probabilistic completeness	depending on correct obstacle constraint model
applicable area	small	large, applicable to all kinds of obstacles	small

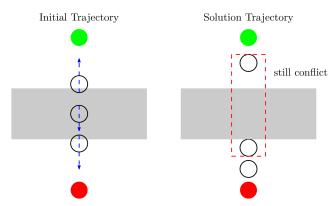


Fig. 1 Scenario where Trajopt's gradient descent cannot resolve the collision

include CHOMP [42], STOMP [43] and Trajopt [44]. It is also possible to use a parameterised spline to represent the trajectory as a sequence of spline segments, and this representation has been used in humanoid robot motion planning [45, 46]. There is extensive, closely related work in optimal control, which focuses less on collisions and more on systems with complex dynamical properties. Some of the most notable work includes differential dynamic programming [47, 48], iterative LQR [49], approximate inference control [50], and optimisation-based control involving contacts [51–54]. Recently, the relationship between the trajectory optimisation and simultaneous localisation and mapping (SLAM) has also been explored [55].

# 3 Constraints for optimal motion planning

In this part, we first analyse three typical types of constraints that are widely used: the collision-free constraints, the pose constraints, and the dynamics constraints. We will discuss how to deal with general constraints. After that, we will briefly discuss how find feasible solutions to these constraints.

#### 3.1 Collision-Free constraints

Both sampling-based and grid-based methods use collision-checking procedure to guarantee all the vertices to be collision-free and use continuous collision checking or other local planning

procedures to guarantee all edges to be collision-free. Thus, the collision-free constraints are implicitly satisfied in the resulting graph. Thus, we here only discuss how the trajectory optimisation formulates the collision-free constraints.

A successful collision-free motion planner for general problems should take into account the global geometric information of obstacles. For instance, the sampling-based planners use the sampling procedure to explore the entire configuration space and obtain the global geometric information step by step. Such global exploration makes the sampling-based planners probabilistic complete. Almost all trajectory optimisation methods only utilise local geometric knowledge about the environment, and thus are not complete, i.e. they cannot guarantee to perform well in all scenarios

For instance, Trajopt [44] determines its collision avoidance strategy based on the gradient information. Such gradient can be misleading in many scenarios (with Fig. 1 as a simple example) since instead of using the entire geometric information about the obstacle, it only leverages the local penetration depth information between the robot and the obstacles.

While geometric methods and CHOMP [42] can succeed in the simple scenario in Fig. 1, they are not general enough to resolve all situations violating the collision-free constraints. CHOMP requires the pre-computation of a potential field in terms of the signed distance to the obstacles, where the potential field can be computed using two different approaches:

- i. calculate a distance function for geometric obstacle primitives (boxes, spheres, and cylinders);
- discretise the entire workspace and calculate distance value for each volume using Euclidean distance transformation (EDT) algorithm.

In this way, CHOMP is able to leverage the global geometric information about the obstacles. However, it can be trapped in the local optima, especially for scenarios with non-convex obstacles. As a result, selecting a good initial guess plays an important role in improving the performance of CHOMP [56].

Note that the reasons why Trajopt and CHOMP need good initial guesses are different. For Trajopt, the distance gradient may not be sufficient to find a feasible solution, and randomness is required to generate enough 'perturbation' such that the initial guess can locate in some attraction region where the gradient can lead the trajectory into a collision-free status. CHOMP fails mainly due to the highly non-linear property of the distance function, which makes the global optimisation challenging.

#### 3.2 Pose constraints

In contrast to the collision-free constraint, trajectory optimisation methods outperform the sampling-based planner when dealing with pose constraints. The main reason is that the pose constraints induces a lower dimensional manifold, i.e. one with zero measure in the configuration space. As a result, the sampling-based planning algorithms, even when using various heuristics, are difficult to generate a sufficient number of samples to reconstruct the connectivity in the constrained manifold [57]. In Table 3, we summarise the different strategies that sampling-based algorithms can use to deal with pose constraints.

Next, in Table 4, we compare the sampling-based methods and trajectory optimisation methods while solving the pose constraints.

#### 3.3 Dynamics constraints

Dynamics constraints are usually the most challenging one for motion planning [19], and there are a wide variety of differential dynamics constraints including non-holonomic constraints and kinodynamic constraints.

Non-holonomic constraints mean the non-integrable speed constraints. This concept was first introduced by Laumond [58] and is mostly about kinematics which can be formulated using action spaces and configuration transition functions. Kinodynamic

**Table 3** Different strategies for sampling-based algorithms to deal with pose constraints

to dear with p	use constraints		
	Rejection	Projection	Direct sampling
strategy	trial and error	heuristic	samples generated according to a suitable
			parameterisation of the constraints
constraint	non-zero volume	zero volume in	1
feature	in the configuration space	the configuration space	
application torque, balance, collision constraints,	binary cases more stringent constraints like pose constraint,	1	_

Note: Both projection and trajectory optimisation are heuristic methods, and thus they are both prone to local minima since the pose constraint is non-linear.

**Table 4** Different strategies for solving the motion planning problems with pose constraints

problems with pose constraints			
	Sampling-based methods: projection	Sampling-based methods: direct sampling	Trajectory optimisation
strategy	r iterative numerical IK: pseudo-inverse- Jacobian method	1. for ≤ 6 DOF: analytical IK algorithm 2. for >6 DOF: pseudo-analytical IK algorithm	SQP to solve non- linear equation
results	prone to local minima	good	prone to local minima

**Table 5** Different constraints and the corresponding ways to find feasible solutions

	Convex constraints	Non-convex constraints	Not well-defined constraints
zero measure	example: end- effector reaching a point	example: pose constraints	example: contact the obstacle surface
_	method: gradient	method: sampling + heuristics	method: heuristics (or projection) + sampling
non-zero measure	example: end- effector reaching a convex region	example: collision constraint (CHOMP)	example: collision- free
_	method: gradient	method: pure sampling or sampling + heuristics	method: sampling

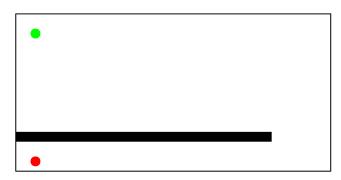


Fig. 2 Scenario where an appropriate sampling distribution is necessary for good performance

constraints were introduced by Donnald *et al.* [59], which added second order or high-order constraints to configuration spaces.

Interesting work about motion planning and optimisation taking into account non-holonomic constraints includes [60], which used sinusoids to steer cars. One solution to deal with the non-holonomic constraints is using discrete time models and reachability graphs.

To deal with kinodynamic constraints, a popular solution is to define motion primitives. For example, Frazzoli [61] used motion primitives to manoeuvre drones; Lamiraux [62] used motion primitives to expand RRT trees, where the node expansion procedure was also optimised. Hsu [63] adapted his EST method to deal with kinodynamic constraints, where the expansion of motion primitives was performed in the configuration-time space and this method is able to deal with moving obstacles.

Besides kinodynamic constraints, there exists work that considered other challenging dynamic constraints such as the friction force [64].

For the drone dynamics, another popular solution besides motion primitives is to use mixed-integer linear programming (MILP), which is first introduced in [65]. In this approach, the environment is segmented into several convex regions [66, 67], and a piecewise polynomial function is used to generate a trajectory that is both free of collision and feasible in dynamics.

The optimal motion planning taking into account the dynamics constraints has many applications in humanoid robots, where balancing is one of the most important factors. There are many classical work about this topic, including [45, 46, 53, 68, 69]. Some recent work on soft robot control also benefited from the progress in this area [70].

#### 3.4 General constraints

The collision-free constraint and pose constraint (for high-DOF robots) share three properties:

- too complex to have an explicit and analytical representation of the constraint manifold in the configuration space;
- ii. the distance to the constraint manifolds is a non-linear function;
- iii. both essentially an inverse kinematics problem.

In Table 5, we further investigate the constraints involved in the motion planning in terms of two factors: the measure of the constraint manifold in the configuration space, and whether the distance to the constraint manifold in the configuration space is well-defined.

# 3.5 Find feasible solutions

Given the constraints, one remaining problem is how to find the feasible set for the optimal motion planning problem, i.e. the set of trajectories that satisfy all the constraints. This is a non-trivial problem due to the non-linearity of many constraints. From the comparison in Table 5, it is obvious that heuristics play an important role in the performance of the motion planning. We here provide a brief comparison between the heuristic sampling-based approach and the trajectory optimisation technique using a concrete example.

Urmson *et al.* [31] proposed a heuristic sampling-based algorithm (hRRT), which biases samples to the state with higher quality by shaping the probability distribution of samples. It emphasised the importance of an appropriate probability distribution in the configuration space for fast convergence. Fig. 2 shows a situation where traditional RRT algorithm may take a long time to find a solution while hRRT can be very fast.

The difference between RRT and hRRT arises because similar to the trajectory optimisation, the sampling-based algorithms also suffer from the lack of global geometric information about the environment when the planning starts, and have to spend a lot of time exploring the configuration space in a local propagation manner. A suitable heuristics can encode the global geometric

**Table 6** Comparison about different approaches to combine the sampling-based methods with optimisation

combine the sampling-based methods with optimisation			
	Offline	Online	Asymptotically
	heuristic	heuristic	optimal methods
	methods	methods	
algorithms	1. short-cutting	heuristically biased tree growth	RRT*
2.	_	_	_
hybridisation			
optimality	no guarantee for optimality	no guarantee for optimality	asymptotically optimal
advantage	better performance within a given time budget [80, 81]	no comparison results available yet	asymptotically optimal
preliminaries	graph	robotics	graph + robotics + math

information implicitly and thus can significantly improve the search performance of the sampling-based methods.

Many heuristics have been proposed in previous work to help the planners satisfy general constraints in an efficient manner. Most of these heuristics are designed for the collision-free constraint, which is considered as the most challenging 'hard' constraint in motion planning. In particular, OBPRM [71], Gaussian sampling [72], retraction-based planners [17, 73], and methods specially designed for narrow passages [12, 74]. Adaptive sampling strategies have also been proposed that evolve while more information about collision-free configuration space can be inferred via sampling. For instance, [75, 76] approximated the free space using a set of size-varying balls around nodes in the RRT representation. Reference [77] approximated the configuration space with a set of prior samples, either collision-free or incollision. Reference [78] extended the adaptive sampling approach in [77] to non-holonomic motion planning by defining the utility of local paths. Reference [79] constructed roadmaps in both the free configuration space and the configuration space obstacles, for generating more samples in narrow passages.

# 4 Objectives for optimal motion planning

Traditional sampling-based algorithms (e.g. RRT, PRM) do not take path cost or quality into account, and most of the time they generate non-optimal value [32]. Luna et al. summarised three ways to improve the quality of sampling-based solution paths [80] as Table 6 lists.

Both [80, 81] demonstrated that given a time budget for planning, offline heuristic techniques for solution optimisation can outperform specialised planners which compute the globally optimal path (RRT\*) in certain scenarios. Optimisation techniques are also combined with the RRT solution in [82].

For grid-based methods, costs are assigned to graph vertices or edges, and then the resolution-based optimality is guaranteed by the underlying graph search algorithm.

For trajectory optimisation, the construction of appropriate objectives is the main way to achieve a path with desirable properties.

# 4.1 Time-optimal trajectory optimisation

An intuitive solution to take into account time is to use a mixed configuration-time space. Most sampling-based methods could be adapted to the configuration-time space by introducing metrics to deal with the asymmetric time dimension [5]. However, when there is bound in speed, the adaption becomes difficult. Researchers studied the problem using motion states and velocity tuning [83], or decoupled coordination [84, 85], but these methods were not complete. Another way is to define action spaces and configuration transition functions. Robot configurations are supposed to change following actions and their correspondent configuration transition functions.

Generally, solving a time-optimal motion planning in an environment with obstacles is not an easy task. However, there is a series work [86-90] on solving the time-optimal trajectory generation in an open space, and the focus is how the complex dynamics determines the time-optimal behaviour of the robot. By leveraging this time-optimal trajectory following procedure, some two-steps are proposed to avoid obstacles in a time-optimal manner [91, 92]. In the first step, a smooth and collision-free trajectory is planned with simplified or zero dynamics taking into account. In the second step, the planner will try to follow the first-step trajectory using the time-optimal following property to compute a trajectory that is both collision-free and time-optimal. Similar techniques have been used for time-optimal RRT replanning in [93].

### 4.2 Other objective functions

There are some other objective functions that are also widely considered in the optimal motion planning, including the trajectory smoothness [42, 44, 94], the energy efficiency [95, 96], which heavily used the tools in Lie groups.

#### 4.3 Summary and conclusions

The past three decades have seen increasingly rapid progress in the optimal motion planning technology. This rapid progress has been enabled by major theoretical advances in the computational complexity, random sampling, and optimisation techniques. Different ways to formalise the optimisation problem have been proposed, including graph/grid formulation and waypoint trajectory optimisation. A large number of different constraints, including challenging collision-free constraints, pose constraints, and nonholonomic constraints, have been tackled. A huge number of objective functions have been proposed to achieve trajectories with different desirable properties. However, there are still many open problems left, such as how to incorporate collision-free constraints in an efficient and effective manner; how to take into account the dynamics constraints in the optimisation framework without twostep sub-optimal solutions. We hope this survey can serve as references for further research in the optimal motion planning area.

# **Acknowledgments**

This work is supported by GRF 17204115 and 21203216.

# References

- [1] Litzenberger, G.: 'Professional service robots: continued increase', (Statistical Department, International Federation of Robotics, 2012. Available at: http://
- www.worldrobotics.org/index.php?id=home&news\_id=262 Lozano Perez, T., Jones, J.L., Mazer, E., et al.: 'HANDEY: a robot task planner' (The MIT Press, Cambridge, 1992) [2]
- Latombe, J.C.: 'Robot motion planning' (Kluwer, Dordrecht, 1991)
- Choset, H., Lynch, K.M., Hutchinson, S., et al.: 'Principles of robot motion: theory, algorithms, and implementations (intelligent robotics and autonomous agents)' (The MIT Press, Cambridge, 2005)
- 'Planning algorithms' (Cambridge University Press, [5] LaVelle, S.M.: Cambridge, 2006)
- [6] Lumelsky, V.J., Stepanov, A.A.: 'Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape', Algorithmica, 1987, 2, (1), pp. 403–430
- [7] Simeon, T., Laumond, J.P., Nissoux, C.: 'Visibility based probabilistic roadmaps for motion planning', *Adv. Robot.*, 2000, **14**, (6), pp. 477–493 Reif, J.H.: 'Complexity of the mover's problem and generalization'. Symp. on
- [8]
- Foundations of Computer Science, San Juan, USA, 1979
  Brooks, R.A., Lozano.Perez, T.: 'A subdivision algorithm in configuration space for findpath with rotation', *IEEE Trans. Syst., Man, Cybern.*, 1985, 2,
- pp. 224-233
  Khatib, O.: 'Real-time obstacle avoidance for manipulators and mobile robot',
  Int. J. Robot. Res., 1988, 5, (1), pp. 90-98 [10]
- Barraquand, J., Latombe, J.C.: 'A monte-carlo algorithm for path planning with many degrees of freedom'. Int. Conf. on Robotics and Automation, Cincinnati, USA, 1990, pp. 1712-1717
- Kavraki, L., Svestka, P., Latombe, J.C., et al.: 'Probabilistic roadmaps for path planning in high-dimensional configuration spaces', IEEE Trans. Robot. Autom., 1996, 12, pp. 566-580

- [13] Kuffner, J.J., LaValle, S.M.: 'RRT-connect: An efficient approach to singlequery path planning'. Int. Conf. on Robotics and Automation, San Francisco, USA, 2000, pp. 995–1001
- Koyuncu, E., Ure, N.K., Inalhan, G.: 'Integration of path/maneuver planning [14] in complex environments for agile maneuvering ucavs', J. Intell. Robot. Syst., 2010, **57**, (1–4), pp. 143–170
- Rickert, M., Sieverling, A., Brock, O.: 'Balancing exploration and [15] exploitation in sampling-based motion planning', IEEE Trans. Robot., 2014, **30**, (6), pp. 1305–1317
- [16] Akbar Agha.mohammadi, A., Chakravorty, S., Amato, N.M.: 'Firm: sampling-based feedback motion-planning under motion uncertainty and imperfect measurements', *Int. J. Robot. Res.*, 2014, **33**, (2), pp. 268–304
- Hsu, D., Kavraki, L.E., Latombe, J.C., et al.: 'On finding narrow passages with probabilistic roadmap planners'. Workshop on the Algorithmic Foundations of Robotics on Robotics, Houston, USA, 1998, pp. 141–153 Canny, J., Reif, J.: 'New lower bound techniques for robot motion planning
- **[18]** problems'. Annual Symp. on Foundations of Computer Science, Los Angeles, USA, 1987, pp. 49–60
- Canny, J., Reg, A., Reif, J.: 'An exact algorithm for kinodynamic planning in the plane', Discrete Comput. Geom., 1991, 6, pp. 461-484
- [20] LaValle, S.M., James, J., Kuffner, J.: 'Randomized kinodynamic planning',
- Int. J. Robot. Res., 2001, 20, (5), pp. 378–400

  Cohen, B.J., Chitta, S., Likhachev, M.: 'Search-based planning for manipulation with motion primitives'. Int. Conf. on Robotics and Automation, Anchorage, USA, 2010, pp. 2902–2908 [21]
- Cohen, B.J., Subramania, G., Chitta, S., et al.: 'Planning for manipulation [22] with adaptive motion primitives'. Int. Conf. on Robotics and Automation, Shanghai, China, 2011, pp. 5478-5485
- Cohen, B., Chitta, S., Likhachev, M.: 'Single- and dual-arm motion planning with heuristic search', *Int. J. Robot. Res.*, 2014, **33**, (2), pp. 305–320 Phillips, M., Likhachev, M.: 'Speeding up heuristic computation in planning [23]
- [24] with experience graphs'. Int. Conf. on Robotics and Automation, Seattle, USA, 2015, pp. 893–899
- [25] Phillips, M., Cohen, B.J., Chitta, S., et al.: 'E-graphs: bootstrapping planning with experience graphs'. Robotics: Science and Systems, Sydney, Australia,
- July 2012 Phillips, M., Dornbush, A., Chitta, S., et al.: 'Anytime incremental planning [26] with e-graphs', Int. Conf. on Robotics and Automation, Karlsruhe, Germany, 2013, pp. 2444-2451
- [27] Likhachev, M., Gordon, G., Thrun, S.: 'ARA\*: anytime A\* with provable bounds on sub-optimality'. Advances in Neural Information Processing, Vancouver, Canada, December 2003
- Likhachev, M., Ferguson, D., Gordon, G., et al.: 'Anytime search in dynamic graphs', Artif. Intell., 2008, 172, (14), pp. 1613–1643
  Stentz, A.: 'The focussed D\* algorithm for real-time replanning'. Int. Joint [28]
- [29] Conf. on Artificial Intelligence, Montreal, Canada, 1995, pp. 1652–1659
- Likhachev, M., Ferguson, D.: 'Planning long dynamically feasible maneuvers [30] for autonomous vehicles', Int. J. Robot. Res., 2009, 28, (8), pp. 933-945
- [31] Urmson, C., Simmons, R.: 'Approaches for heuristically biasing rrt growth'. Int. Conf. on Intelligent Robots and Systems, Las Vegas, USA, 2003, Vol. 2, pp. 1178–1183
- Karaman, S., Frazzoli, E.: 'Sampling-based algorithms for optimal motion [32] planning, Int. J. Robot. Res., 2011, **30**, (7), pp. 846–894 Karaman, S., Walter, M.R., Perez, A., et al.: 'Anytime motion planning using
- the rrt\*'. Int. Conf. on Robotics and Automation, Shanghai, China, 2011, pp. 1478-1483
- Karaman, S., Frazzoli, E.: 'Incremental sampling-based algorithms for optimal motion planning', *Int. J. Robotics Res.*, 2011, **30**, (7) Alterovitz, R., Patil, S., Derbakova, A.: 'Rapidly-exploring roadmaps: [34]
- [35] weighing exploration vs. Refinement in optimal motion planning'. Int. Conf. on Robotics and Automation, Shanghai, China, 2011, pp. 3706-3712
- [36] Gammell, J.D., Srinivasa, S.S., Barfoot, T.D.: 'Informed RRT\*: optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic'. Int. Conf. on Intelligent Robots and Systems, Chicago, USA, 2014, pp. 2997–3004
- [37] Marble, J.D., Bekris, K.E.: 'Asymptotically near-optimal planning with probabilistic roadmap spanners', IEEE Trans. Robot., 2013, 29, (2), pp. 432-
- Li, Y., Littlefield, Z., Bekris, K.E.: 'Asymptotically optimal sampling-based [38]
- kinodynamic planning', *Int. J. Robot. Res.*, 2016, **35**, pp. 528–564 Dobson, A., Bekris, K.E.: 'Sparse roadmap spanners for asymptotically near-optimal motion planning', *Int. J. Robot. Res.*, 2014, **33**, (1), pp. 18–47 [39]
- Shaharabani, D., Salzman, O., Agarwal, P.K., et al.: 'Sparsification of motion-planning roadmaps by edge contraction'. Int. Conf. on Robotics and [40] Automation, Karlsruhe, Germany, 2013, pp. 4098–4105
- Wang, W., Balkcom, D., Chakrabarti, A.: A fast streaming spanner algorithm for incrementally constructing sparse roadmaps'. Int. Conf. on Intelligent [41] Robots and Systems, Tokyo, Japan, 2013, pp. 1257-1263
- Ratliff, N., Zucker, M., Bagnell, J.A., et al.: 'Chomp: gradient optimization [42] techniques for efficient motion planning'. Int. Conf. on Robotics and Automation, Kobe, Japan, 2009, pp. 489-494
- Kalakrishnan, M., Chitta, S., Theodorou, E., et al.: 'STOMP: stochastic trajectory optimization for motion planning'. Int. Conf. on Robotics and Automation, Shanghai, China, 2011, pp. 4569–4574
  Schulman, J., Duan, Y., Ho, J., et al.: 'Motion planning with sequential
- [44] convex optimization and convex collision checking', Int. J. Robot. Res., 2014, 33, (9), pp. 1251–1270
- [45] Lengagne, S., Vaillant, J., Yoshida, E., et al.: 'Generation of whole-body optimal dynamic multi-contact motions', Int. J. Robot. Res., 2013, 32, (9-10), pp. 1104-1119

- Escande, A., Mansard, N., Wieber, P.B.: 'Hierarchical quadratic programming: fast online humanoid-robot motion generation', Int. J. Robot. Res., 2014, 33, (7), pp. 1006-1028
- [47] Jacobson, D.H., Mayne, D.Q.: 'Differential dynamic programming' (Elsevier, Amsterdam, 1970)
- Atkeson, C.G.: 'Using local trajectory optimizers to speed up global optimization in dynamic programming'. Advances in Neural Information Processing Systems, Denver, USA, 1994, pp. 663-670
- [49] Todorov, E., Li, W.: 'A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems'. American Control Conf., Portland, USA, 2005, Vol. 1, pp. 300–306
- Toussaint, M.: 'Robot trajectory optimization using approximate inference'. Int. Conf. on Machine Learning, Montreal, Canada, 2009, pp. 1049–1056 [50]
- Mordatch, I., Todorov, E., Popović, Z.: 'Discovery of complex behaviors through contact-invariant optimization', ACM Trans. Graph., 2012, 31, (4), pp. 43:1–43:8
- Tassa, Y., Erez, T., Todorov, E.: 'Synthesis and stabilization of complex [52] behaviors through online trajectory optimization'. Int. Conf. on Intelligent Robots and Systems, Vilamoura, Portugal, 2012, pp. 4906-4913
- Erez, T., Todorov, E.: 'Trajectory optimization for domains with contacts using inverse dynamics'. Int. Conf. on Intelligent Robots and Systems, Vilamoura, Portugal, 2012, pp. 4914–4919
- Posa, M., Tedrake, R.: 'Direct trajectory optimization of rigid body dynamical [54] systems through contact'. Algorithmic Foundations of Robotics, 2013, pp.
- [55] Dong, J., Mukadam, M., Dellaert, F., et al.: 'Motion planning as probabilistic inference using Gaussian processes and factor graphs', 'Robotics: Science and Systems' (Ann Arbor, USA, 2016)
- [56] Dragan, A., Gordon, G., Srinivasa, S.: 'Learning from experience in manipulation planning: setting the right goals'. Int. Symp. on Robotics Research, Flagstaff, USA, 2011
- [57] Berenson, D., Srinivasa, S., Kuffner, J.: 'Task space regions: a framework for poseconstrained manipulation planning', Int. J. Robot. Res., 2011, 30, (12), pp. 1435–1460
- [58] Laumond, J.P.: 'Trajectories for mobile robots with kinematic and environment constraints'. Int. Conf. on Intelligent Autonomous Systems, Amsterdam, The Netherlands, 1986, pp. 346–354
- Donald, B.R., Xavier, P.G., Canny, J., et al.: 'Kinodynamic planning', J. [59] ACM, 1993, 40, pp. 1048–1066
- Murray, R.M., Sastry, S.: 'Nonholonomic motion planning: steering using sinusoids', IEEE Trans. Autom. Control, 1993, 38, pp. 700-716
- [61] Frazzoli, E.: 'Robust hybrid control of autonomous vehicle motion planning',
- PhD Thesis, Massachusetts Institute of Technology, 2001 Lamiraux, F., Ferre, E., Vallee, E.: 'Kinodynamic motion planning: [62] connecting exploration trees using trajectory optimization methods'. Int.
- Conf. on Robotics and Automation, New Orleans, USA, 2004, pp. 3987–3992 Hsu, D., Kindel, R., Latombe, J.C., et al.: 'Randomized kinodynamic motion planning with moving obstacles', Int. J. Robot. Res., 2002, 21, (3), pp. 233-
- Morinaga, A., Svinin, M., Yamamoto, M.: 'Motion planning of drifting [64] vehicle with friction model considering nonholonomic constraint'. Int. Conf. on Robotics and Automation, Seattle, USA, 2015, pp. 650-655
- Richards, A., How, J.P.: 'Aircraft trajectory planning with collision avoidance [65] using mixed integer linear programming'. American Control Conf., Anchorage, USA, 2002, vol. 3, pp. 1936–1941 Landry, B., Deits, R., Florence, P.R., et al.: 'Aggressive quadrotor flight
- [66] through cluttered environments using mixed integer programming'. Int. Conf. on Robotics and Automation, Stockholm, Sweden, 2016, pp. 1469–1475
- Barry, A.J., Majumdar, A., Tedrake, R.: 'Safety verification of reactive [67] controllers for uav flight in cluttered environments using barrier certificates'. Int. Conf. on Robotics and Automation, Saint Paul, USA, 2012, pp. 484-490
- [68] Feng, S., Whitman, E., Xinjilefu, X., et al.: 'Optimization-based full body control for the darpa robotics challenge', J. Field Robot., 2015, 32, (2), pp.
- Dai, H., Tedrake, R.: 'Planning robust walking motion on uneven terrain via [69] convex optimization'. Int. Conf. on Humanoid Robots, Cancun, Mexico, 2016, pp. 579-586
- [70] Marchese, A.D., Tedrake, R., Rus, D.: 'Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator', Int. J. Robot. Res., 2016, 35, (8), pp. 1000-1019
- Amato, N.M., Bayazit, O.B., Dale, L.K., et al.: 'OBPRM: an obstacle-based [71] prm for 3D workspaces'. Workshop on the Algorithmic Foundations of Robotics on Robotics, 1998, pp. 155–168
- Boor, V., Overmars, M.H., van der Stappen, A.F.: 'The Gaussian sampling [72] strategy for probabilistic roadmap planners'. Int. Conf. on Robotics and
- Automation, Detroit, USA, 1999, pp. 1018–1023
  Rodriguez, S., Tang, X., Lien, J.M., et al.: 'An obstacle-based rapidly-exploring random tree'. Int. Conf. on Robotics and Automation, Orlando, [73] USA, 2006, pp. 895–900
- Sun, Z., Hsu, D., Jiang, T., et al.: 'Narrow passage sampling for probabilistic
- roadmap planners', *IEEE Trans. Robot.*, 2005, **21**, (6), pp. 1105–1115 Jaillet, L., Yershova, A., La Valle, S.M., *et al.*: 'Adaptive tuning of the sampling domain for dynamic-domain RRTs'. Int. Conf. on Intelligent Robots and Systems, Edmonton, Canada, 2005, pp. 2851–2856 Yershova, A., Jaillet, L., Simeon, T., et al.: 'Dynamic-domain RRTs: efficient
- [76] exploration by controlling the sampling domain'. Int. Conf. on Robotics and Automation, 2005, pp. 3856–3861
- Burns, B., Brock, O.: 'Toward optimal configuration space sampling',
- 'Robotics: Science and Systems' (Cambridge, USA, 2005) Knepper, R.A., Mason, M.T.: 'Real-time informed path sampling for motion planning search', Int. J. Robot. Res., 2012, 31, (11), pp. 1231–1250 [78]

- Denny, J., Amato, N.M.: 'Toggle PRM: simultaneous mapping of c-free and c-obstacle a study in 2D'. Int. Conf. on Intelligent Robots and Systems, San [79]
- Francisco, USA, 2011, pp. 2632–2639 Luna, R., Şucan, I.A., Moll, M., et al.: 'Anytime solution optimization for sampling-based motion planning'. Int. Conf. on Robotics and Automation, [80] Karlsruhe, Germany, 2013, pp. 5068–5074 Luo, J., Hauser, K.: 'An empirical study of optimal motion planning'. Int.
- [81] Conf. on Intelligent Robots and Systems, Chicago, USA, 2014, pp. 1761-
- Jetchev, N., Toussaint, M.: 'Trajectory prediction in cluttered voxel environments'. Int. Conf. on Robotics and Automation, Anchorage, USA, [82] 2010, pp. 2523-2528
- [83] Kant, K., Zucker, S.W.: 'Toward efficient trajectory planning: the path velocity decomposition', Int. J. Robot. Res., 1986, 5, pp. 72-89
- LaValle, S.M., Hutchinson, S.A.: 'Optimal motion planning for multiple robots having independent goals', *IEEE Trans. Robot. Autom.*, 1998, **14**, pp. [84] 912-925
- van den Berg, J., Overmars, M.: 'Prioritized motion planning for multiple [85] robots'. Int. Conf. on Intelligent Robots and Systems, Edmonton, Canada, 2005, pp. 2217–2222
- Bobrow, J.E., Dubowsky, S., Gibson, J.S.: 'Time-optimal control of robotic manipulators along specified paths', *Int. J. Robot. Res.*, 1985, **4**, (3), pp. 3–17 [86]
- path parameterization algorithm', *IEEE Trans. Robot.*, 2014, **30**, (6), pp. 1533–1540 [87]
- Constantinescu, D., Croft, E.A.: 'Smooth and time-optimal trajectory planning for industrial manipulators along specified paths', J. Robot. Syst., 2000, **17**, (5), pp. 233–249

- Verscheure, D., Demeulenaere, B., Swevers, J., et al.: 'Time-optimal path [89] tracking for robots: A convex optimization approach', IEEE Trans. Autom. Control, 2009, **54**, (10), pp. 2318–2327 Homsi, S.A., Sherikov, A., Dimitrov, D., et al.: 'A hierarchical approach to
- minimum-time control of industrial robots'. Int. Conf. on Robotics and Automation, Stockholm, Sweden, 2016, pp. 2368-2374
- [91] Shiller, Z., Dubowsky, S.: 'On computing the global time-optimal motions of
- Yamamoto, M., Iwamura, M., Mohri, A.: 'Quasi-time-optimal motion planning of mobile platforms in the presence of obstacles', *IEEE Trans. Robot. Autom.*, 1991, 7, (6), pp. 785–797

  Yamamoto, M., Iwamura, M., Mohri, A.: 'Quasi-time-optimal motion planning of mobile platforms in the presence of obstacles'. Int. Conf. on Robotics and Automation, Detroit, USA, 1999, vol. 4, pp. 2958–2963 [92]
- Otte, M., Frazzoli, E.: 'RRTX: asymptotically optimal single-query samplingbased motion planning with quick replanning', Int. J. Robot. Res., 2016, 35, (7), pp. 797–822
- Macfarlane, S., Croft, E.A.: 'Jerk-bounded manipulator trajectory planning: [94] design for real-time applications', IEEE Trans. Robot. Autom., 2003, 19, (1),
- Bobrow, J.E., Martin, B., Sohl, G., et al.: 'Optimal robot motions for physical criteria', J. Robot. Syst., 2001, 18, (12), pp. 785-795
- Lee, S.H., Kim, J., Park, F.C., et al.: 'Newton-type algorithms for dynamics-based robot movement optimization', *IEEE Trans. Robot.*, 2005, **21**, (4), pp. 657-667