machines

MDPI

Review

Conventional, Heuristic and Learning-Based Robot Motion Planning: Reviewing Frameworks of Current Practical Significance

Fatemeh Noroozi [1,*], Morteza Daneshmand [1,†] and Paolo Fiorini [2]

[1] Norwegian Institute of Bioeconomy Research (NIBIO), 1431 Ås, Norway; morteza.daneshmand@nibio.no
[2] Department of Engineering for Innovation Medicine, University of Verona, 37134 Verona, Italy; paolo.fiorini@univr.it
* Correspondence: fatemeh.noroozi@nibio.no
† M.D.'s affiliation is previously with Institute of Technology, University of Tartu, 50411 Tartu, Estonia.

Abstract: Motion planning algorithms have seen considerable progress and expansion across various domains of science and technology during the last few decades, where rapid advancements in path planning and trajectory optimization approaches have been made possible by the conspicuous enhancements brought, among others, by sampling-based methods and convex optimization strategies. Although they have been investigated from various perspectives in the existing literature, recent developments aimed at integrating robots into social, healthcare, industrial, and educational contexts have attributed greater importance to additional concepts that would allow them to communicate, cooperate, and collaborate with each other, as well as with human beings, in a meaningful and efficient manner. Therefore, in this survey, in addition to a brief overview of some of the essential aspects of motion planning algorithms, a few vital considerations required for assimilating robots into real-world applications, including certain instances of social, urban, and industrial environments, are introduced, followed by a critical discussion of a set of outstanding issues worthy of further investigation and development in future scientific studies.

Keywords: motion planning; mobile robots; social robots; self-driving cars; humanoid robots

Citation: Noroozi, F.; Daneshmand, M.; Fiorini, P. Conventional, Heuristic and Learning-Based Robot Motion Planning: Reviewing Frameworks of Current Practical Significance. *Machines* **2023**, *11*, 722. https://doi.org/10.3390/machines11070722

Academic Editors: Nicola Ivan Giannoccaro, Ramiro Velázquez and Gianfranco Parlangeli

Received: 17 May 2023
Revised: 4 July 2023
Accepted: 4 July 2023
Published: 7 July 2023

Correction Statement: This article has been republished with a minor change. The change does not affect the scientific content of the article and further details are available within the backmatter of the website version of this article.

1. Introduction

Artificial Intelligence (AI)-based utilities and techniques have their most prevalent applications in robotics, where agents possessing computing capabilities gather, analyze, and use information from their environment to make and act upon informed decisions. Despite the usual lack of sufficiently reliable information about their environment, a robotic agent has to make rational decisions or at least be capable of autonomously making a sequence of moves that optimize the expected values of indicators of certain functional criteria to an acceptable extent, while minimizing the costs incurred. The environment may change during the course of executing a motion plan, which motivates the incorporation of the capability for modifying the plan based on the aforementioned variations. Taking into account the fact that it is not feasible to compute reactions to all possible scenarios in advance, it is common to instead perform replanning computations and optimizations based on the specific scenario at hand. This may include various factors, such as model and tracking errors, as well as different types of constraints, according to the sensor feedback and user inputs [1,2].

An autonomous (e.g., robotic) system might involve multiple rigid parts connected using numerous types of links, hinges, and joints. They may be either dependent or independent in terms of their movement [3]. The navigation of an autonomous system requires localization, perception and cognition, and motion planning [4], with the latter constituting the central focus of this review. A schematic representation of the role of

motion planning within the workflow of AI-based systems is shown in Figure 1, together with more specific illustrations of its place in the structure governing the performance of robotic systems and self-driving cars. The figure was inspired by the definitions and representations provided in [1,5].

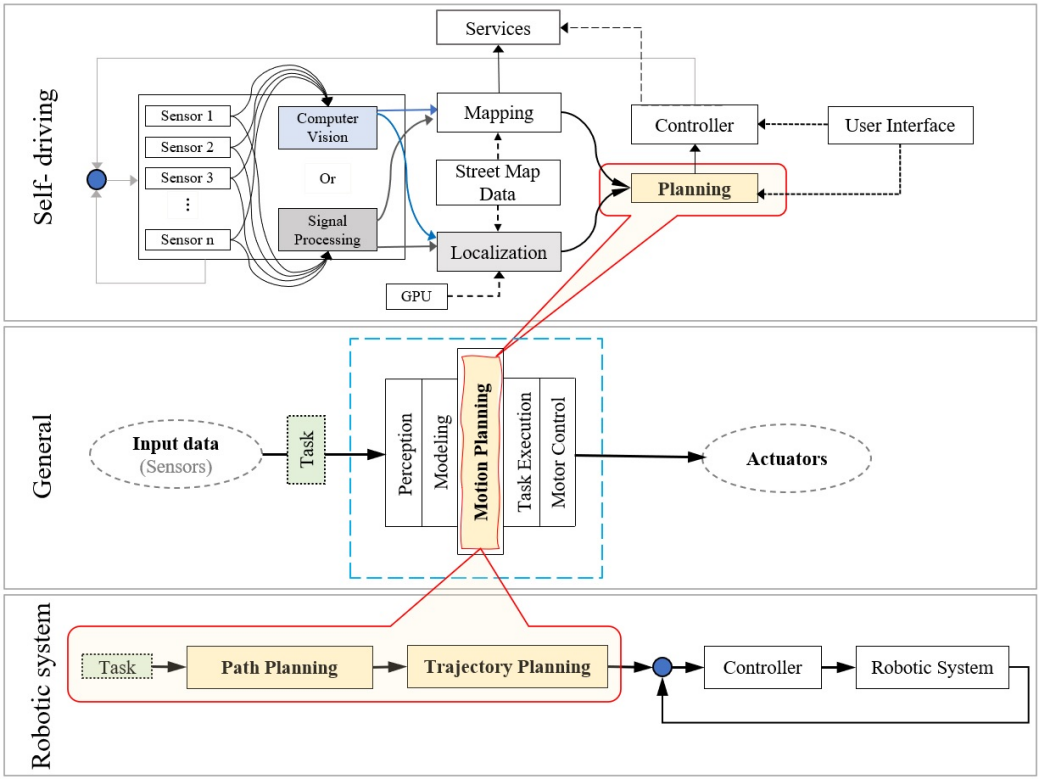

Figure 1. A schematic representation of the place of motion planning within a general AI-based pipeline (**middle**), with more specific illustrations of self-driving cars (**top**) and robotics systems (**bottom**). The figure was inspired by the conceptions and diagrams reported in [1,5].

Motion planning in cluttered environments is important in various applications relying on autonomous systems. A basic motion planning problem for an autonomous system with the shape known from the outset deals with a 2D or 3D environment, which may include obstacles with known shapes and positions, where the aim is to determine a collision-free route for the autonomous system from an initial position and orientation (pose) to the final one, if a route exists [3].The pose needs to be estimated throughout the operation using, e.g., image processing techniques, to rectify the path, and correspondingly the control commands, based on the actual situation. Various types of robotic manipulators, such as spatial and tentacle-like, require motion planning, with considerable uncertainties arising from size limitations, compliance and collision-avoidance requirements, and sensing unreliability [6]. Thus, finding a suitable path depends on a variety of experimental factors, such as the presence of static or dynamic obstacles and other types of uncertainties, and must be handled using relevant control strategies, e.g., Dynamic Programming (DP) [7].

Perception of the surrounding environment by the robot may be achieved using various sensory modalities, such as sound signals. In this configuration, a set of audition capabilities enable the robot to locate a sound source, among other functionalities. For example, a microphone array installed on a robot may be utilized to measure the angle of arrival, thereby estimating the location of the sound source, where the front–back ambiguity may be alleviated based on the robot's motion, even in noisy and reverberant environments, potentially with moving sources, using, e.g., Kalman filtering [8].

The results include a set of locations, i.e., via points, at which the autonomous system may be positioned, as well as rule sets intended to constrain the movements of the au-

tonomous system between them [9]. Generally, path planning aims to find the intermediate points, such that the final pose can be reached within the shortest possible period, while taking into account an environmental and spatial model [10].

Path planning may be performed either in the joint space or in the operating space. It is limited to kinematic planning, and the criteria for evaluating performance mostly concern the computational costs, meaning that it does not account for the physics-based requirements of the problem [11]. Therefore, subsequently, trajectory planning needs to be performed, which incorporates the time information into the plan, to determine the dynamic aspects of the motion, as well as the time at which each of the via points are passed. The resulting system is often called a Kinodynamic motion Planning (KP) algorithm.

The importance of such concepts stems from the fact that inertial forces and torques are affected by the robot's acceleration, where jerk, which is mathematically represented as the derivative of acceleration, has a direct impact on mechanical vibrations. Thus, a trajectory planning method may concentrate on optimizing one or several factors, including execution time, energy consumption, jerk, route length, safety, computation time, and smoothness [12], and this can be achieved using, e.g., an artificial bee colony algorithm for a local search, as well as evolutionary programming for refining the path [13]. While planning a trajectory, the relevant constraints, including temporal, physical, geometric, and friction-related, as well as joint limits and equilibrium requirements, where applicable, should also be satisfied [2,10,14]. Mathematically, a broad category of classical problems concern motion planning under differential constraints [15].

Motion planning considering dynamics can be significantly challenging; since, due to the lack of a local planner, the only primitive toward the state space comes from the controls being forward-propagated. This has been alleviated in the literature through employing propagation of random controls in each iteration, utilizing tree Sampling-Based Motion Planning (SBMP) techniques, leading to asymptotic optimality, i.e., ultimate optimality in terms of the cost of the path, as the number of points grows toward infinity. Nevertheless, the convergence of the resulting approaches to suitable trajectories may be unacceptably slow [16,17].

Adjustable links and joints can be considered to model trajectories for improved flexibility against unexpected dynamic obstacles, such that the trajectory can be conveniently modified by creating multiple obstacle trajectories based on the relationships between them and using queries along the link chains [18].

Despite the above geometric perspectives, practical use cases may entail technical complexities of different sorts, including functional uncertainties and noise during sensing and control, probabilistic incompleteness, nonholonomic constraints, optimality requirements such as time or vibration minimization [19], dynamic constraints, error recovery concerns, route caching preconditions, and a lack of essential information about the scene and obstacles. Therefore, during the last four decades, researchers from several communities, such as computational geometry and robotics, have been engaged in developing suitable frameworks for this purpose [3].

A certain aspect of the problem concerns robot autonomy, which is necessary for enhancing the robot's safety and its performance, in terms of delivering insightful information about the scene it is supposed to navigate through [20]. This, in turn, gives rise to extra factors and requirements regarding the fact that the robot has to be capable of capturing and processing some or all of the information for determining its actions.

Another challenge underlying the task of motion planning arises from the intricacy of achieving onboard implementation of the procedures required for handling the dynamics of using the robot's computational resources [21], as well as the lack of sensory capabilities that are required for a consistent performance across different situations, i.e., where the robot needs to make inferences about possible valid configurations [22]. Additionally, different configuration parameters may affect the quality of the trajectory returned by a motion planning algorithm, and these are usually adjusted beforehand [23].

As a common practice, the problem of motion planning is converted from the Cartesian space (workspace) to the higher-dimensional configuration space, which is often referred to as the C-space. The goal is to reduce the problem of checking for 3D collisions between spatial objects to that of simpler point-like tests. Nevertheless, making the required map in the C-space may become intractable computationally and memory-wise with the higher dimensions of the C-space. This may stem from factors such as high numbers of Degrees of Freedom (DoF), or from the time-dependency of the environment, e.g., because of obstacles dynamically appearing, disappearing, or moving, and requirements for being incorporated online into the C-space [24].

Other challenges may also be presented by the necessity of finding a probabilistic compromise between safety and speed in collision checking at the edges of the sensing field [25]. It should, however, be noted that forcing a robot to perform as fast as possible may undermine its accuracy and repeatability. Therefore, it is vital to plan trajectories that can be performed quickly and highly efficiently, and which are smooth enough not to require excessive acceleration or extreme actions, which would otherwise damage mechanical parts such as actuators or control modules, because of, e.g., extreme vibrations [12].

Once a trajectory has been planned, a low-level controller needs to be employed so the autonomous system follows it, while maintaining the safety and compensating for tracking errors using, e.g., sum-of-squares programming for upper-bound limits of the tracking error [26].

Furthermore, completeness and exactness are considered essential properties of any motion planning algorithm [27]. Probabilistic completeness refers to the property of an algorithm, where the probability of it being unable to return a solution if at least one exists decays to zero as the number of samples grows to infinity [28]. Similarly, a motion plan needs to be singularity-free, meaning that the end-effector should not reach regions of the workspace where it is not able to move along one or more of its DoF or where it could move irrespective of the input joint positions. It is worth noting that avoiding local coordinates may be useful for tackling singularities or ambiguities [29].

Generally, a navigation package consists of two main modules: a global planner that finds the optimal path according to prior knowledge of the environment and the static obstacles, and a local planner that refines the path, to avoid collisions with dynamic obstacles [30].

Motion planning algorithms involve loosely coupled multilayered designs that aim at dynamically feasible solutions, which are found quickly enough to be suitable for online planning. Lower-dimensional projection spaces are used for approximating a lead path from the start to the goal configuration. The results are then employed as initial guesses for the second planner, considering the dynamic constraints and possible cases of Inevitable Collision State (ICS). This is also fundamental when it comes to leveraging the computational cost by performing calculations up to a reasonable horizon at each interval, and focusing on regions where high-quality solutions are deemed most likely to be found [21].

SBMP, which will be further discussed in the upcoming sections, is highly reliable, in terms of avoiding obstacles, and is therefore useful in cases where the task must be performed while inferring and trying to satisfy a set of dynamic constraints. Instead of specifically and numerically programming a robot, planning its motions and performing a certain task may be accomplished through learning the actions and constraints. For example, expert demonstrations may be used within a Demonstration-Guided Motion Planning (DGMP) framework for tasks involving, e.g., holding an object in an upright or horizontal position while moving it or cleaning a surface [31].

Motion planning algorithms have been extended and refined to improve their efficiency, reliability, energy and time consumption, computational cost, physical feasibility, robustness, and stability. However, most types of robot are not yet capable of working in close contact, while cooperating, and collaborating with each other and with human beings. Thus, efficiently incorporating them into practical contexts involves unresolved

challenges. For example, social robots should behave seamlessly, smoothly, and similarly to humans. This necessitates additional performance indicators measuring their aptness for social interactions.

Motion planning is of great value in, e.g., search and rescue, Simultaneous Localization And Mapping (SLAM) exploration, teleoperation, inspection, construction, and architecture. For example, it could be utilized for searching collapsed buildings that are considered either hazardous or unreachable. Applications of robots in home and office environments also abound. Given the fact that teleoperation may become temporarily unavailable, it is essential to equip a robot with autonomy, where distances are predicted using regression and ranking with a steer function, and the two-point boundary value problem is solved using, e.g., non-linear parametric models benefiting from constant-time inferences. Each motion planning problem aimed at optimizing a robot's performance needs to be set up based on the mission's representative indicators. For example, when it comes to SLAM, the goal is to determine a sequence of motions that yield a set of viewpoints suitable for making reasonably accurate models, suffering the least from known obstacles, such as holes [32,33].

The remainder of this review is organized as follows: First, in Section 2, relevant studies and taxonomies are reviewed, to clarify the differences and advantages of the present survey in comparison with similar surveys. Then, a brief overview of a set of motion planning techniques widely applied to robots in the existing scientific and technical literature is presented in Section 3. Next, we review some of the most fundamental aspects of their practical applicability to different types of robot in Section 4. Finally, a list of concluding remarks and hints for possible future research and development directions are outlined in Section 5.

Overall, the contributions of the present review are as follows:

- By analyzing the frequency of use of conventional, heuristic, and learning-based algorithms, we demonstrate the current transition, whereby more and more studies have recently embraced learning-based models instead of older heuristic approaches, while the conventional techniques are still being utilized at a similar ratio as before;
- We provide two different categorizations of these algorithms, one based on being conventional or heuristic, and the other based on being global or local. To the authors' knowledge, our categorizations are the most comprehensive proposed in the literature to date, thereby enabling a sound and quick judgment of the role, importance, and relevance of each technique for a given application;
- By identifying and studying the most common motion planning pipelines of current practical significance, we exclude algorithms that are no longer being actively employed as of the time of writing or that have not yet proven useful in real-world applications. Considering the large amount of literature being published on the topic of motion planning every year, the materials presented throughout this survey are of essential value for readers whose purpose is to grasp an overall understanding of the field, as opposed to deeply analyzing the mathematical and theoretical backbones.

2. Related Work and Taxonomy

The taxonomies developed to classify motion planning techniques include different types. In this section, a selection of the most recent reviews will be briefly discussed, to provide a broad insight into the taxonomies, as well as their relevance and applicability.

From the broadest perspective, collision-avoiding motion planning techniques belong to two main categories; namely, SBMP methods and trajectory optimization approaches [34]. Historically, each motion planning method can considered classical or heuristic, the former being represented by RoadMap (RM), cell decomposition, subgoal networks, and potential fields, as popular examples. These are deemed to be capable of handling simplified motions and environments only [17]. It is widely perceived that the classical methods are simple, but their main disadvantages are being computationally expensive or intractable, and they may fail to tackle uncertainties appropriately [4].

Motion planning has always been dependent on computer science, aiming at devising solutions that ensure a good level of reliability, accuracy, and precision. Nevertheless, one of the predicaments in motion control is that, in modeling, different frameworks are required for motion planning and obstacle avoidance [27].

Assuming that proper safety considerations are incorporated, robots may improve various factors, such as mobility, transportation convenience, and efficiency in people's daily lives. This applies to numerous types of robots and self-driving vehicles. More clearly, a system needs to be planned to work smoothly, safely, conveniently, and efficiently, e.g., in terms of energy consumption, within a dynamic environment that may include objects, agents, and live beings. Actions are either decided or rethought based on the feedback received about the present state. For an informative review of motion planning and control techniques aimed at urban environments, see [35]. By discussing and contrasting the associated motion models and the extent of applicability of the methods for certain types of environments and computational resources, we provide insights into possibilities for improving the system design.

Onboard sensors and networks communicating information between the agents and the environment, as well as between the agents themselves, can play essential roles in automating the functionalities of intelligent vehicles. This gives rise to the possibility, and in a broader context, the necessity, of having agents cooperate for the application to be viable within complex real-world environments involving Vulnerable Road User (VRU)s. For a comparative study of motion planning frameworks for intelligent vehicles, see [36].

A survey of 3D path planning techniques for robots was provided in [10]. Algorithmic motion planning techniques dealing with simple problems, as opposed to heuristic practical techniques, were reviewed in [3], which offers a theoretical analysis of the methods from the point of view of computational and combinatorial geometry, as applied to surfaces and curves, as well as the worst-case asymptotic bounds.

From another perspective, the problem of motion planning can be viewed focusing on whether the autonomous system targeted is holonomic or nonholonomic. In simple words, if the number of controllable DoF an autonomous system possesses is the same as its overall number of DoF, then it is considered holonomic, and if not, it is nonholonomic. Due to the extremely high level of complications arising in nonholonomic motion planning, it has attracted significantly more attention. A survey of motion planning and obstacle avoidance techniques for nonholonomic mobile robots was reported in [27].

Due to the recent increasing interest in autonomous driving, numerous studies have surveyed the state of the art of self-driving vehicles. An informative review of motion planning techniques for on-road intelligent vehicles and transportation systems was provided in [1]. It first explored the related concepts from the point of view of motion planning for mobile robots and then clarified the nuances and necessary considerations for employing the inferences in the context of autonomous vehicles.

Another survey of motion planning and control techniques for intelligent vehicles in urban environments was reported in [35], where different strategies toward mobility model and various environment structures were discussed, along with the relevant implications regarding the computational aspects, aiming at providing a more insightful perspective on the role of choices made at the system design level.

In [36], a review of motion planning for self-driving cars was presented, covering overtaking maneuvers. A study of the feasibility and optimality of different methods under various environmental conditions was provided in [37].

Randomized path planning algorithms may be either single-query or multiple-query. The methods falling under the former category try to solve a single path planning problem quickly, without performing or requiring preprocessing, while the methods belonging to the latter intend to solve multiple path planning problems in the same environment. Thus, in the case of multiple-query path planning methods, data structures containing information from preprocessing could help achieve a faster performance [38]. The basic Rapidly-exploring Random Trees (RRT) construction procedure is summarized in Algorithm 1,

where the EXTEND operation is illustrated in Figure 2. Moreover, the RRT-connect is shown in Algorithm 2.

Algorithm 1: Basic RRT construction. This algorithm was taken from [38].

```
Function BUILD_RRT(q_init)
    T.init(q_init);
    for k = 1 to K do
        q_rand ←RANDOM_CONFIG();
        EXTEND(T, q_rand);
    end
    return T

Function EXTEND(T, q)
    q_near ←NEAREST_NEIGHBOR(q, T);
    if NEW_CONFIG(q, q_near, q_new) then
        T.add_vertex(q_new);
        T.add_edge(q_near, q_new);
        if q_new = q then
            return Reached
        else
            return Advanced
        end
    return Trapped
```

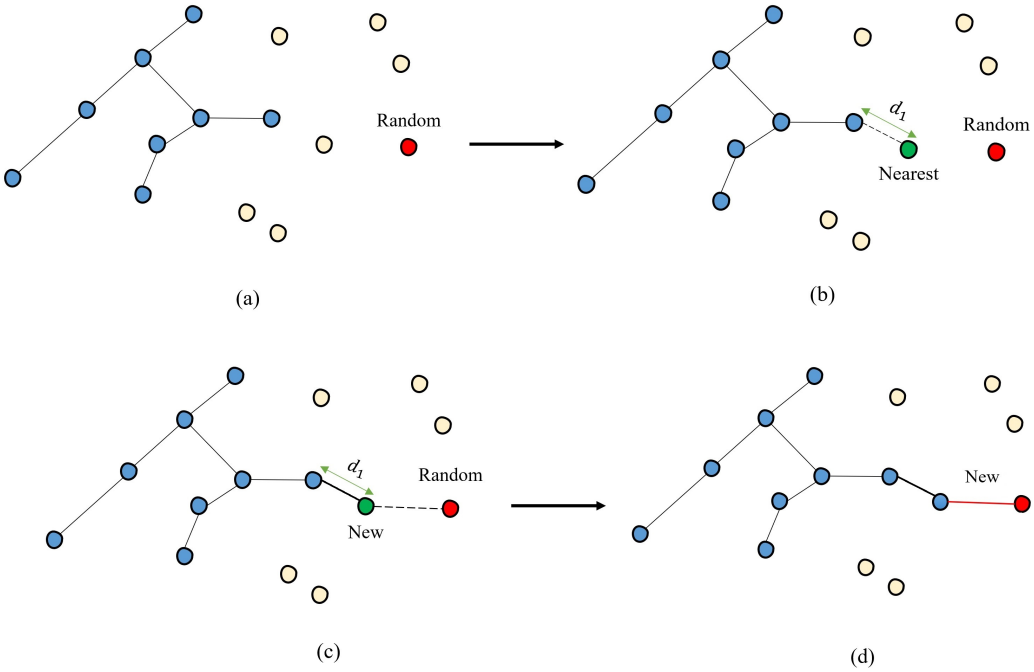

Figure 2. The four steps of the EXTEND operation. The search tree is shown in its original form, i.e., at the time of starting to find a path to a random point (**a**). Then the nearest neighbors are determined (**b**), followed by making a new node (**c**), and finally, finding the random point (**d**).

Algorithm 2: The RRT-Connect algorithm, which was taken from [38].

Function CONNECT($\mathcal{T}, q$)
  repeat
    $S \leftarrow$EXTEND($\mathcal{T}, q$);
  until ($S \neq Advanced$);
  return S

Function RRT_CONNECT_PLANNER(q_{init}, q_{goal})
  $\mathcal{T}_a$.init(q_{init}); $\mathcal{T}_b$.init(q_{goal});
  for $k = 1$ **to** K **do**
    $q_{rand} \leftarrow$RANDOM_CONFIG();
    if $EXTEND(\mathcal{T}_a, q_{rand}) \neq Trapped$ **then**
      if $CONNECT(\mathcal{T}_b, q_{new}) = Reached$ **then**
        return PATH($\mathcal{T}_a$,$\mathcal{T}_b$);
    SWAP($\mathcal{T}_a$,$\mathcal{T}_b$);
  end
  return $Failure$

Optimality guarantees for SBMP algorithms under differential constraints were theoretically evaluated and compared in [15], concentrating on drift-less control-affine dynamical models, where the concept of converging with probability has been introduced, resulting in a higher flexibility and convergence rate bounds, in contrast to sure converging.

The nature-inspired approaches, fuzzy logic, Neural Network (NN)s, and hybrid methods utilized in heuristic-based path planning were reviewed in [4]. Another general survey of motion planning was provided in [12].

A review of the 3D motion planning techniques widely utilized for underwater, ground, and aerial robots was conducted in [10], where the exploration mechanisms were divided into five categories and compared in terms of their implementable areas and time efficiencies.

The advantages and disadvantages of path planning techniques from the point of view of special robotic operations were reviewed in [39]. Several approaches to analyzing the safety and feasibility in path planning were investigated and compared, considering their role in contexts involving underwater vehicles, Unmanned Aerial Vehicle (UAV)s, Autonomous Guided Vehicle (AGV)s, and industrial robots, with applications in mountainous areas, warehouses, and production lines. Examples concerning AGVs in medical and industrial scenarios were specifically discussed, followed by a detailed analysis of a certain robotic spray painting problem.

As aforementioned, the technical aspects of motion planning problems and pipelines have been well studied from various perspectives in the existing literature, which seems to have overlooked the requirements associated with the practical assimilation of these technologies and strategies. For example, much more could be achieved in the sense of fast, safe, and reliable communication, cooperation, and collaboration with other agents and human beings present in the same environment. The taxonomy employed in the present survey, therefore, concentrates on the application side, aiming at shedding some light on additional considerations for, e.g., social, mobile, and humanoid robots, as well as real-world frameworks involving, e.g., self-driving, object manipulation, and multi-robot teams working in cooperation. From a technical viewpoint, on top of the path planning and trajectory optimization components of motion planning algorithms, typical online replanning capabilities are reviewed, along with other topics with a functional impact, including Demonstration-Based Learning (DBL), scene uncertainties and dynamics, stability, and computational feasibility.

As the focus of the present review is on incorporating motion planning techniques into practical contexts, the articles' suitability and importance were judged according to their

significance in the field, as measured based on their number of citations, which reflects their impact on how the relevant applications are evolving. Therefore, the basic list of studies for this survey consisted of 200 articles, being compiled by searching the keyword "motion planning" on Google Scholar, to ensure that no prior bias or subjective preference would affect the type of motion planning algorithms being picked up and explored.

The period of publication for the above list was set to cover the period from 2015 to 2019, so that each article could be deemed to have had enough time to collect citations from the time of publication, but, at the same time, would not be too old to present notions with current practical implications. Nevertheless, some of the papers were excluded from the basic list, due to their lack of direct relevance to present applications with real-world usage.

However, some older studies have also been included where necessary for presenting theoretical foundations. On the other hand, newer studies have also been covered where a similar search for articles published after 2019 indicated significantly high numbers of citations within a short period. More clearly, these papers were considered to have made a considerable contribution, in terms of either introducing a new approach or showing great potential to emerge as one.

A summary of the motion planning algorithms reviewed in this survey, along with their important characteristics, is provided in Table 1, which also lists selected references for each approach, together with the frequency of the articles covered based on the publication year. Similarly, a graphical overview is shown in Figure 3. Moreover, the most significant criteria involved in judging the suitability of each motion planning algorithm are schematically listed in Figure 4.

Figure 3. A graphical summary of the motion planning algorithms reviewed in the present survey.

Table 1. A summary of the most essential properties of motion planning algorithms discussed throughout this survey, and their selected references, along with the number of articles covered for each publication year. The criteria used while judging the advantages and disadvantages of each algorithm are illustrated in Figure 4.

Category	Description	Advantages	Disadvantages	Selected References
Converntional				
SBMP	Analyzing alternative trajectories based on path length, by low-dispersion or random sampling, using, e.g., RRT, RMP, FMT, FMT*, RRT*, RRT$^{\#}$, RRT^{X}, RRT-Connect, RRT*-Connect, RRG, CBF-RRT, LBT-RRT, RRT2.0, BTT, BIT*, or MC-RRM	Not requiring a model of the environment, high reliability in avoiding obstacles, as well as inferring and satisfying dynamic constraints, the possibility of achieving asymptotic optimality and probabilistic completeness, simplicity, and quick performance	Slow convergence, unreliability in cases involving complex motions or environments, runtime computational heaviness and intractability, especially in the presence of obstacles, and poor safety because of randomness	[6,8,15,17,28,32,38,40–67]
RM	Multi-query search using a pose graph to find feasible motions based on path length or accumulated pose uncertainty, using, e.g., cell decomposition, a sub-goal network, PRM, PRM*, visibility graph, or spanners	Simplicity, probabilistic completeness, asymptotic optimality, the possibility of parallelization of collision detection over the RM edges, and reliability in tackling dynamic obstacles and constraints	Poor performance for complex motions or in unstructured environments, high runtime computational cost and intractability, weak safety due to randomness, and the possibility of excessive growth or failure in the presence of uncertainties	[68–71]
Potential field	Guiding the motion based on repulsion from obstacles and attraction toward the goal, with respect to the distances, thereby handling dynamics, as well as utilizing velocity and orientation information	Real-time performance, simplicity, high safety, asymptotic optimally	Probabilistic incompleteness, high computational cost and intractability, unreliability against uncertainties, narrow paths, and dynamic environments, and the possibility of local minima	[72,73]
Heuristic				
Bio-inspired	Using multi-query search algorithms, such as bee colony, GA, PSO, BBO, or the bat algorithm	Handling discrete functions and achieving convergence for complex problems, and ease of combining with other methods	Probabilistic incompleteness, high computation cost, slowness, weakness against dynamics and probabilities, and the possibility of local minima	[13,74–82]
NN	Making search trees by predicting the cost, based on dynamics, heuristics, and obstacles, using ANNs, or LSTM	High generalizability	Challenging dataset preparation and the possibility of becoming stuck in local minima	[16,83–86]
Learning-based				
DBL	Using human requests based on a kinetic model to learn near-optimal heuristics representing the operator's choice from path planning, and making cost maps and features, possibly combined with RL or IRL	Handling unseen or unstructured environments by generalizing from primitive motions to more sophisticated ones, without explicit task constraints, and predicting human motions for improved speed	Requiring large datasets and powerful online adaptation	[31,87–96]
DL	Automatic selective sampling using, e.g., OracleNet, DeepSMP, MPNet, GANs, or CoMPNetX	Theoretical worst-case guarantees, real-time performance, and handling higher dimensionalities, as well as cluttered or unfamiliar environments	Challenging requirements for training datasets	[97–101]
RL	Sequential decision-making through LPV state-space representations, using, e.g., PPO, TD3, EASE, SAC-based methods, DDPG, or other dual architectures	Actively exploring the domain, handling unfamiliar environments and high-dimensionalities, and tackling dynamic obstacles accurately by identifying their boundaries and distances to the robot	Requiring high numbers of samples, training difficulties due to convergence and robustness issues caused by ambiguities between Cartesian and joint spaces, continuous workspaces, and redundant DoF	[102–114]

Frequency by year										
<2015	2015	2016	2017	2018	2019	2020	2021	2022	2023	Total
17	29	37	40	28	31	7	11	8	5	213

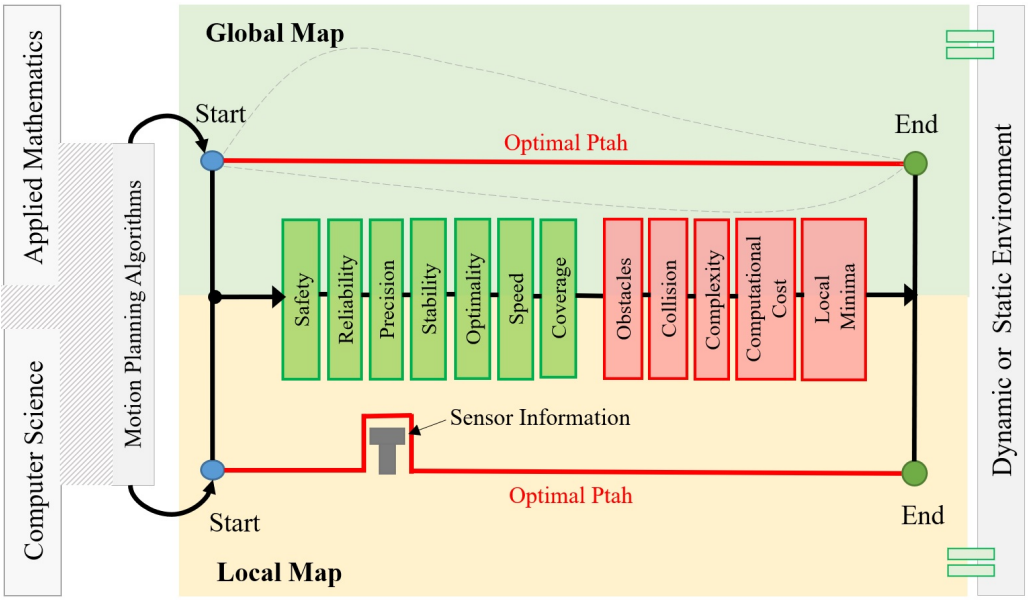

Figure 4. A schematic list of the main criteria considered while evaluating the suitability of motion planning algorithms, where green and red boxes indicate positive and negative factors, respectively. Computer science and applied mathematics constitute the theoretical backbones of motion planning, where the and the goal is to find the optimal path from the starting point to the goal using a local or global planner, within a dynamic or static environment.

3. Motion Planning Pipelines

Motion planning is part of a more general problem referred to as Task and Motion Planning (TMP), which typically takes place under partial observability. A symbolic decision tree is employed, which may undergo additional branching based on new observations. Independent optimization of the symbolic trajectories is performed using approximate path costs, followed by picking up the best policy and optimizing a joint trajectory tree [115,116]. In what follows, the essential aspects of the most common approaches to motion planning in robotics will be concisely discussed.

3.1. SBMP

Not requiring an explicit model of the obstacles present in environments, SBMP algorithms utilize a collision-checking module, which analyzes the practicability of alternative trajectories. Through connecting a set of collision-free configurations in space, they build a graph of feasible trajectories [40]. On the other hand, the set of connected edges thought to lead to the shortest path from the initial to the final configuration is called the shortest-path-to-goal sub-tree [41]. The continuity of the algorithm depends on the configuration space's topological complexity [117].

A fundamental strength of data-driven SBMP algorithms arises from their relatively low reliance on explicit mathematical models, which are not available from the outset in most practical scenarios [42].

RRT is a well-known example of this type of motion planning algorithm. It has engaged hundreds of researchers, each of whom has tried to verify and demonstrate its applicability to an application of a certain kind. Consequently, investigating the performance of RRT has been the subject of a substantial number of studies, where making modifications aiming at improving the speed, fluency, accuracy, and stability were among the main factors in evaluating the resulting functionality [43].

SBMP methods mainly depend on uninformed path sampling procedures, whose examples include low-dispersion and random sampling. Thus, the information required for collision testing needs to be accessible to the planner. However, this might not be feasible in many scenarios [44].

Occupancy maps are commonly associated with SBMP methods. Nevertheless, they suffer from the deficiency that the resulting cost estimation is limited to the search heuristic utilized over the unknown environment. Thus, only intermediate objectives along the frontiers can be considered [118].

Multi-Component Rapidly-exploring RoadMap (MC-RRM) is a SBMP motion planning method that aims to optimize a learned cost metric. It is useful for incrementally computing a motion plan according to DBL [31], which will be further discussed in Section 3.5.

RRT* and Rapidly-exploring Random Graph (RRG) are examples of algorithms that present slight differences with RRT. They lead to asymptotic optimality but are slower than RRT. Performing continuous interpolation between the fast RRT and its asymptotically optimal alternatives, with a cost function representing the path length, brings the ability to quickly solve path planning problems, while maintaining asymptotic optimality. Taking advantage of a sub-graph of the RM returned by RRG and an auxiliary lower-bound graph, a parameter may be utilized and adjusted to specify how similar to RRG or RRT* the behavior of the algorithm needs to be.

For example, Lower Bound Tree (LBT)-RRT is an efficient single-query SBMP method taking advantage of the above strategy. It offers an asymptotically near-optimal solution within an approximation factor $1 + \epsilon$ of the optimal solution. It makes use of an approximation factor, which if set to 1 or infinity, makes the algorithm behave like RRG or RRT, respectively, and for any other value, yields a compromise between the speed of RRT and the path quality, i.e., asymptotic optimality, of RRT*. The LBT-RRT algorithm applies to problems of DoF ranging from 3 to 12 [45]. Moreover, probabilistic belief networks resulting from pose SLAM could be utilized as belief RM for optimal collision-free trajectory planning by searching through the pose graph, regardless of the map reference frame, and considering the accumulated robot pose uncertainty as one of the criteria [46,68].

On the other hand, running multiple independent RRTs could be considered in a context where the aim is to achieve asymptotic convergence to a minimal Collision Probability (CP) [6]. For complex motion planning problems that are in high-dimensional configuration spaces or for problems that involve a high number of obstacles for which it is computationally expensive to check collisions, lazy dynamic programming recursion [47] on a prescribed number of samples that have been probabilistically determined helps deal with the problem, through dividing it into smaller sub-problems, as well as allowing skipping collision-checks when evaluating local connections, thereby making the problem more tractable. Moreover, to further improve efficiency, the incremental growth of the tree of candidate paths can be one-pass, i.e., it could take advantage of a heapsort technique to determine the appropriate sample point systematically, and consequently, grow only in the outward direction, thereby avoiding backtracking over sample points that have already been evaluated.

To improve mathematical flexibility, aiming at making it possible to impose a lower bound on the convergence rate, instead of almost sure convergence, the concept of convergence in probability can be considered. The Fast Marching Tree (FMT)* algorithm is a viable example of such a strategy, which can offer asymptotic optimality, while still converging on the optimal solution faster than alternatives such as Probabilistic RoadMap (PRM)* and RRT*. Under a set of considerations for the configuration spaces and tuning parameters, a lower bound of the order $O\left(n^{-1/d+\rho}\right)$ can be achieved for the convergence rate, with n, d, and ρ being the number of sampled points, the dimension of the configuration space, and an arbitrarily small constant, respectively. It has been shown that the FMT* algorithm is asymptotically optimal, even in cases where a non-uniform sampling of the configuration space is taken into account, a general cost function is used instead of the prevalent path length, or the connections are determined according to the number of nearest neighbors, e.g., by using the k-Nearest-Neighbor (kNN) algorithm, instead of considering a predetermined connection radius [28].

Probabilistic SBMP algorithms, such as PRM and RRT, have proven extremely successful, mainly because of their favorable mathematical advantages, including probabilistic

completeness and asymptotic optimality, as well as their practical usefulness. Their property of being probabilistic stems from the notion that they find a path through connecting independently and identically distributed (i.i.d.) random points within the robot's configuration space. However, they also entail drawbacks. For example, they may not be able to guarantee safety and do not allow offline computations, which could otherwise help alleviate the runtime computational intensiveness.

To counteract the above phenomena caused by the randomness of probabilistic SBMP methods, one may think of ways to incorporate deterministic characteristics into the system. Using deterministic low-dispersion sampling sequences, along with a suitable configuration of tuning parameters, PRM leads to deterministic asymptotic optimality. The convergence rate, which is represented by a sub-optimality factor, may be controlled through an upper bound, in terms of the $l2$-dispersion of the sampling sequence and PRM's connection radius. It is worth mentioning that $l2$-dispersion stands for how well a set of points covers the space. Geometrically, this represents the largest Euclidean ball that does not touch any of the points.

The above framework is associated with a space and computational complexity that could be arbitrarily close to $O(n)$, which is the theoretical minimum, where n is the number of points in the sequence. It is possible to achieve similar performances, i.e., with reasonable success rates and path costs, on differentially-constrained problems [48].

In dynamic environments, performing offline computations to obtain prior information may be impossible. Therefore, the ability to carry out fast replanning, while maintaining asymptotic optimality under a single-query framework, is crucial. As soon as it is found that the appearance or motion of an obstacle will make it collide with the robot while following the shortest-path-to-goal sub-tree, replanning needs to be performed to modify the graph, such that collisions are avoided. Older single-query replanning algorithms performed the task in sch a way that the branches disconnected as a consequence of this operation are pruned away, followed by regrowing parts or all of the graph.

However, being able to transfer information quickly is essential for a timely reaction in dynamic environments involving unpredictable changes in the presence or location of obstacles. Thus, the computation time could be considerably enhanced by trying to modify and repair the existing graph instead. This may be achieved by making use of a fast graph rewiring cascade, which incorporates new information and accordingly updates the shortest-path-to-goal sub-tree.

Moreover, the graph and the shortest-path-to-goal sub-tree could be built in the state-space of the robot itself, to ensure that the motion complies with the kinematics of the robot, and gradually improve throughout the navigation. The RRT$^{\mathrm{X}}$ algorithm uses this strategy. It offers probabilistic completeness. RRT$^{\mathrm{X}}$ does not perform global and local planning separately, but it is still suitable for real-time performance in dynamic environments, due to its capability of quickly reacting to the emergence and movement of obstacles.

A major factor affecting the competence of an algorithm to quickly react to changes in obstacles is the information transfer time, which is the amount of time the algorithm needs to transfer information regarding a decrease in the cost associated with a node to the relevant parts of the graph. The time required to transfer information regarding an ϵ-cost reduction to a graph of size n is $O(n \log n)$, $\Omega\left(n\left(\frac{n}{\log n}\right)^{1/D}\right)$, and $\omega\left(n \log^2 n\right)$ for RRT$^{\mathrm{X}}$, RRT* and RRT$^{\#}$, respectively. These are all asymptotically optimal single-query algorithms.

RRT and RRT* have a $O(\log n)$ amortized iteration time in static environments. This is $\omega\left(\log^2 n\right)$ in the case of RRT$^{\#}$. For RRT$^{\mathrm{X}}$ to achieve a $O(\log n)$ amortized iteration time in static environments, each node can bear a set of $O(\log n)$ expected neighbors and the graph may retain ϵ-consistency for a prescribed ϵ [41].

Combining the virtues of some of the aforementioned randomized algorithms, new ones such as RRT*-connect have been devised for solving single-query problems using a bidirectional search. They are faster than RRT* but , despite RRT-Connect, are capable of converging to the theoretical optimum, i.e., offering asymptotic optimality. Such a strategy

is particularly useful for a faster performance in applications such as autonomous driving in complex environments [49].

Asymptotic optimality or near-optimality of a broad category of SBMP algorithms depends on the connection radius around a configuration q, which is expected to be connected to all configurations inside the ball. To accelerate the functionality, in contrast to algorithms operating based on connecting to the k nearest neighbors, the foregoing notion could be adapted to non-Euclidean spaces. This is often seen in practical applications [50].

Alternatively, motion planning may be performed using the Monte Carlo algorithm, based on criteria such as Shannon entropy or the standard deviation of the estimated belief for the source location [8].

Despite the RRT algorithm, RRT2.0 is asymptotically optimal. Avoiding Boundary Value Problem (BVP), which is required to establish connections between the nodes within a tree for algorithms such as RRT*, FMT*, and Batch-Informed Trees (BIT)*, it utilizes forward propagation, where every point in the configuration space is represented in an augmented fashion through supplying the associated cost-to-come, which represents the aggregation of the costs incurred by the edges constituting the path. RRT2.0 offers optimality under the assumption that the objective function and the dynamics are Lipschitz-continuous, where a trajectory with positive clearance from the obstacles and a piecewise-constant control function can be approximated [51].

When it comes to SBMP algorithms in the Euclidean space with uniform random sampling, e.g., in the case of methods such as BoTtleneck Tree (BTT), FMT*, and RRG, as well as PRM-based methods, given n samples and d dimensions, connection radiuses less than a critical value proportional to $\Theta\left(n^{-1/d}\right)$ lead to failure of the algorithm in returning asymptotically (near) optimal results. For greater values, the probability of success will be at least $1 - O\left(n^{-1}\right)$. Additionally, instead of $\Theta(\log n)$ induced by a radius of order $\left(\frac{\log n}{n}\right)^{1/d}$, only $\Theta(1)$ connections will be required [52].

Among notable KP algorithms, Kinodynamic motion Planning by Interior-Exterior Cell Exploration (KPIECE) has a relatively high success rate in finding time-optimal solutions, and Synergistic Combination of Layers of Planning (SyCLoP) yields comparatively high levels of power-optimality [11].

Although numerous SBMP strategies offer asymptotic optimality, they may suffer from a slow convergence, which could be alleviated through intelligence contributed by certain heuristics aimed at faster exploration. More clearly, once a tentative solution is found, further sample points can be taken from a collection of points commonly referred to as the "informed set". The heuristic needs to be chosen paying due attention to providing a reasonable estimate of the cost associated with the solution. Further focusing the search is possible based on metrics such as cost-to-come, thereby determining more limited point sets such as the "relevant region" proposed in [53].

3.2. Trajectory Optimization

Although SBMP algorithms may be able to offer feasible solutions to the problem under high-dimensional configuration spaces, the resulting solutions might entail undesired properties, such as unnecessary movements or jerk, which is further worsened in the presence of tight navigation constraints or sparsely scattered obstacles. As a remedy for the foregoing shortcomings, by analyzing the problem from a probabilistic viewpoint, smooth continuous-time trajectories can be obtained as samples from a Gaussian Process (GP) [119].

Trajectory optimization may target criteria such as the time required to follow the trajectory, velocity, jerk, curvature, or a combination of these, while the step-time is optimized simultaneously [46].

Complex motion planning tasks for many-DoF robots can be handled using functional gradient algorithms, which are capable of finding an optimal, smooth trajectory within a search space, while also taking into account geometric requirements such as smoothness. They rely on a finite parameterization of the trajectories represented by a list of waypoints.

It is worth noting that the performance may suffer from drawbacks such as an extremely small step size or a high number of iterations, which can be alleviated in adaptively lower dimensions through avoiding waypoints by representing trajectories as linear combinations of kernel functions within Reproducing Kernel Hilbert Spaces (RKHSs).

In the above context, optimization may be performed using different kernels such as Basis splines (B-splines), as well as Laplacian or Gaussian Radial Basis Functions (RBFs) [120]. Splines are useful for reducing a problem's dimensions and the number of constraints, where solving the problem with a receding horizon helps handle modeling errors and environmental uncertainties [121]. The occupancy gradients may then be optimized considering an update rule defined based on a stochastic process [34,122]. As a relevant point, an online adaptation of the aforementioned plans based on experience requires differentiation [23].

Numerous adaptive bio-inspired strategies such as the bat algorithm [74] also exist, which may be utilized for optimal motion planning.

In higher-dimensional state spaces, trajectory optimization can also be combined with other approaches to locomotion, such as graph search, which leads to high flexibility, dynamical feasibility, and probabilistic optimality [123].

Piecewise linear paths may be further improved in terms of curvature, tangential or acceleration continuity, and boundedness, which are important for avoiding undesired phenomena such as vibrations, as well as confined chord error, by utilizing interpolation and performing optimization for the Curvature Variation Energy (CVE). Furthermore, additional computational costs can be alleviated using mixed linear and quartic Bezier segments [124].

Various criteria may need to be considered and accounted for while formulating a trajectory optimization problem. These include path continuity, infinite-norm velocities, the joints' maximum angles and velocities, and the possible drifts caused by redundancies. For example, the constraints arising from the requirement that the speeds amount to zero at the end of a task may be dealt with using frameworks such as Infinity-Norm Velocity Minimization (INVM), Repetitive Motion Planning (RMP), or linear variational inequalities.

3.3. Bio-Inspired Algorithms

Biological structures found in nature have been the source of inspiration for various optimization algorithms. Examples include the movement of ants and the work of honey bees, who do not normally interfere with each other's motion [78–80].

Genetic Algorithm (GA)-based approaches have been widely utilized in the literature for route design problems, due to their capability for handling discrete functions. Particle Swarm Optimization (PSO) frameworks are also inspired by biological patterns derived from the social life of birds and their coordinated movements [75,82].

Different types of objective functions can be optimized using such bio-inspired methods. However, they are not capable of producing high-quality solutions in real time, because of their notoriously high computational cost. Furthermore, they cannot tackle dynamic environments and probabilities, because they typically return only one solution and are prone to becoming trapped in local minima [81].

Nevertheless, they were still the foremost category of optimization techniques utilized in motion planning at the time of the widespread deployment of Convolutional Neural Network (CNN)s, which have dominated the field since the year 2015, and more conspicuously, since 2020 [5,125]. This will be discussed in more detail in Section 3.5.

3.4. Online Replanning

Despite the prior considerations and predictions, actual performance is usually affected by various factors that prevent a robot from correctly following the trajectory planned. Fixing these issues requires performing execution-time replanning. On the other hand, there may be occasions where the newly captured sensory information renders the previous inferences invalid.

Other factors, such as uncertainties in the robot's kinematic model, sensor noise, and the obstacles' motion reinforce the necessity and importance of online replanning. A robot's success in carrying out its mission and the operation time are greatly dependent on its capability for replanning, especially in environments involving unstructured terrain or extreme occlusions.

The manner in which such problems are alleviated also depends on the source and cause of the uncertainties, among other factors. For example, a neural dynamics design may be employed through a pseudo-inverse-type formulation, to alleviate the impacts of time-varying noise [126]. Parallel instances of the same planner may be utilized to search through numerous heuristic weights and update them, thereby achieving fast replanning. High-Frequency Replanning (HFR) performs this task while letting the robot carry out the prior decided plan at the same time. Stochastic nonlinear dynamics are typically linearized, for the noise and uncertainties to become tractable. Assuming the availability of sufficient computational resources, asymptotic optimality may also be achieved [54,55].

3.5. DBL

In numerous motion planning tasks, the ideal behavior is supposed to be the that presented by a human expert. However, due to the complexity of deriving formal descriptions of the reward functions underlying expert behaviors systematically, a suitable strategy consists in employing nonlinear Inverse Reinforcement Learning (IRL) based on maximum entropy. In other words, it tries to extract approximate reward functions, which however may not match the inherent unknown one perfectly, leading to a similar behavior under similar conditions [87]. This can be accomplished by making cost maps from demonstrations. For example, expert driving behaviors in a complex urban environment can be learned based on a large number of demonstrations. This may be achieved using raw sensor data that are converted to cost maps and features directly, without requiring manual interference or processing. Using a Fully Convolutional neural Network (FCN) makes it possible to handle large datasets and complex behaviors efficiently [88].

Personal robots could also learn how to plan their motions to perform simple household tasks, such as cleaning the surface of a table or moving a spoon of sugar from a bowl to a cup, based on demonstrations [31]. Raw sensor data captured from expert demonstrations, e.g., 2D laser range findings, could be learned and directly mapped to robotic motion plans represented in the form of steering commands, in an end-to-end fashion. The foregoing procedure would need to be performed using supervised learning aimed at an existing motion planner, which may be employed for seen or unseen environments of a virtual or real nature, using e.g., the grid-based global approach [89].

In SBMP, task constraints aimed at specific missions, e.g., holding a glass of water upright to avoid spilling, need to be explicitly programmed. However, DBL provides the possibility of automatically deriving task constraints, which may be hard to express or determine manually, from the principal patterns representing the constraints, entailing less variation across different demonstration instances [31].

Despite these challenges, predicting human motion is feasible, at least under predefined task descriptions, such as single-arm reaching motions for manufacturing, and this is very beneficial for fast and efficient performance. Assumptions include the fact that human motion is optimal with respect to an unknown cost function and that online iterative replanning is capable of capturing the adaptation of a human's motion to that of a robot. Upon detecting sample motion trajectories, inverse optimal control may be utilized to learn the aforementioned cost function based on a human kinetic model using Stochastic Trajectory Optimizer for Motion Planning (STOMP) [90]. Safety requirements may be ensured by applying upper bounds to the collision probabilities based on the Gaussian distributions according to which the predicted motion is represented [91].

On the other hand, the operator's intentions can appear in the form of explicit requests, which need to be disambiguated and interpreted in terms of specific motion commands, through variable grounding based on the likelihoods of valid choices within a goal re-

gion [92]. For more complicated tasks, user preferences in terms of, e.g., the desired temporal or spatial constraints, may be learned and iteratively revised based on the choice of trajectory among the existing options [93].

In various contexts, a robot's performance may be affected by its capability for generalizing from basic motions and actions to more sophisticated ones. This would be of substantial benefit in terms of having the robot take on responsibilities it has not been specifically programmed to do. It would also make the robot adaptable, significantly beyond carrying out repetitive tasks, which is not sufficient for operating in unstructured environments. A promising approach to the foregoing task consists in combining kinesthetic user guidance with Reinforcement Learning (RL). More clearly, once the features of the primitive actions are captured based on the screw transformation of the end-effector, they are mapped and employed to compose the reward function for RL by training a new motion planning policy. The system developed in [96] takes advantage of this strategy and offers the option of requesting further learning of basic motions if the tasks available from the outset do not suffice for considering the new constraints. A more detailed discussion of the role of RL in motion planning will be presented in Section 3.7.

3.6. Deep Learning

As aforementioned, SBMP may incur a significantly high computational complexity, which grows excessive with increased problem dimensionality. One way to alleviate the foregoing shortcoming is to sample the configuration space selectively and focus only on regions that stand a higher chance of leading to the optimal path. Adaptive sampling may be performed using either manual heuristics or learning. Nevertheless, the former cannot handle higher dimensionalities and is not suitable for cluttered or unfamiliar environments [97].

Deep Learning (DL) networks, however, do not suffer from such drawbacks and may be trained to produce heuristics. They are capable of creating sample nodes. This enhances the computational efficiency and helps accommodate higher dimensionalities, where the resulting hybrid approach retains theoretical worst-case guarantees. DL can handle kinematic constraints and produce near-optimal trajectories in real time.

The stepping Recurrent Neural Network (RNN) OracleNet [99], Deep Sampling-based Motion Planner (DeepSMP) [97], Motion Planning Networks (MPNet) [98], Generative Adversarial Network (GAN)s [100], and the gradients-based Constrained Motion planning Networks x (CoMPNetX) [101] are examples of NNs that have been successfully utilized for DL-based motion planning.

For example, DeepSMP consists of a contractive autoencoder that uses point clouds to encode workspaces, followed by a stochastic deep feedforward NN with dropout that recursively creates node samples for an optimal end-to-end path based on the starting and goal configurations.

Nevertheless, preparing suitable datasets enabling the network to achieve this is typically not straightforward. This may be alleviated by decreasing the number of samples involved in training through active learning, where only certain samples chosen by the DL network itself are supplied in each round of training [5,97,98].

3.7. Reinforcement Learning

Similarly to DL, RL, which is aimed at sequential decision-making, can be employed for motion planning in unfamiliar environments. It can resolve high-dimensional problems involving dynamic obstacles by taking into account their location over a limited number of timestamps within the past horizon [108].

RL methods such as the Proximal Policy Optimization (PPO) algorithm [105] actively explore the domain. They are more accurate than supervised learning approaches, which lack data on the boundaries of obstacles [112]. However, RL suffers from the shortcoming that the number of samples needed for training may be too high. This undermines their

practical usefulness. Nevertheless, an efficient reward function may help reduce the amount of data required [5,104].

Examples of the RL-based algorithms utilized in motion planning include the twin delayed deep deterministic policy gradient (TD3) [107] and Exploitation of Abstract Symmetry of Environments (EASE). The latter relies on locally adopting spatial symmetry abstractions obtained from naïvely trained agents [108].

Soft Actor Critic (SAC)-based methods constitute another category of such approaches, where optimizing several NNs based on maximum entropy [110] can be applied for an improved sampling efficiency. Moreover, Prioritized Experience Replay (PER) enables a more efficient use of data for training by changing the weights of samples based on Temporal-Difference (TD) error. Nonetheless, it should be noted that PER is computationally expensive, and inefficiently adjusting its hyperparameters may undermine the performance of RL [111,113].

Other schemes, such as Deep Deterministic Policy Gradient (DDPG), have also been successfully deployed [112]. Meta-learning can be accompanied by experience replay separation based on a success/failure discrimination within separate buffers randomly sampled at a ratio determined through NN-based learning [111]. Hindsight Experience Replay (HER) is another technique for utilizing training data more efficiently [106].

As a common approach to motion planning based on RL, a Linear Parameter Varying (LPV) state-space representation can be used to tackle dynamic obstacles. More specifically, a switching mechanism is embedded within a dual architecture, where the mode is determined based on the distance between the robot and the obstacles. The two modes may differ from each other in terms of, e.g., whether or not the joint positions are directly controlled, whereby the RL planner takes control when obstacles are perceived to have become too close to the robot [109].

As aforementioned, training RL models is challenging in terms of convergence and robustness. The reasons for this include ambiguities in the relationship between the Cartesian and joint spaces, continuous workspaces, and redundant DoF, which result in unnecessary explorations [107]. This could be alleviated using a NN to produce an initial policy for guiding the training of the RL framework [105].

3.8. Scene Uncertainties and Dynamics

Scene characteristics constitute one of the main factors influencing the performance of motion planning frameworks. This signifies the importance of finding out unknown or unexpected phenomena, and updating existing information regarding the environment.This may be acquired using an autonomous system's sensory equipment from the rest of the autonomous systems available within a reasonably close vicinity, through communication networks connecting them via Vehicle-to-Vehicle (V2V) protocols, or from management centers, via Vehicle-to-Infrastructure (V2I) protocols [36]. Doing so requires further incorporation of transportation engineering concepts [37].

A principal component of such information concerns obstacles and their properties, including their position and shape. These may be dynamic or even nonexistent from the point of view of the planner, until they appear. In cases where the constraints, e.g., the ones related to the obstacles, need to be captured and dealt with in real time, i.e., during the runtime, they may emerge in an even more complex form, especially when affected by sensing noise [127]. If the terrain is harsh or uneven or involves a clutter of static or dynamic obstacles, this can result in an unstructured environment [20].

The navigational complexity of a topological space may be evaluated based on the underlying homotopy [128]. Uneven or irregular surfaces are often encountered in the course of carrying out search and rescue tasks and conducting space missions. They require properties such as modularity, with which the robot adjusts its configuration according to the terrain properties. The terrain can be extremely rough and involve various obstacles, in which case, the area reachable by the robot is significantly limited, demanding the generation of a higher number of locomotion gates, e.g., walking or crawling, and

thereby enabling the robot to switch between different motion primitives, considering the requirements arising from the number and properties of obstacles [56].

Motion planning for modular robots can greatly benefit from elementary motion primitives. This may be realized using locomotion generators, leading to flexibility in the shape of the robot, the environmental conditions, and the number of modules, where the necessity for reconfiguration is also obviated [56].

Terrain dynamics can be affected by media penetration, resulting in deformations appearing as, e.g., dirt or sand. This negatively affects the reliability of maneuvers involving walking or jumping. Closed-loop dynamics may be modeled using an added-mass description of the grain motions, taking into account the hydrodynamic and hydrostatic effects [129].

Foothold and motion planning could be coupled using a parameterized dynamic model, to efficiently analyze, at whole-body level, torque and kinematic limits, as well as the friction cones, at the same time as trajectory optimization. This enables considering the effect of topology based on terrain normals [130].

Irregular surfaces also result in complications such as stochastic bouncing and nonlinear gravity, which may be handled by, e.g. solving Lambert's orbital boundary value problems, followed by propagating model and control uncertainties to compute landing distributions, which are, in turn, utilized to figure out an energy- or time-optimal hopping strategy based on a policy gradient. The final step consists in performing sequential planning through a Markov decision process using Least Squares Policy Iteration (LSPI), leading to an off-policy, off-line, and sample-efficient RL strategy [114].

Approximate Clearance Evaluation (ACE) is particularly useful for handling uneven terrain, where the terrain height bounds are utilized to find out the state bounds of articulated suspension systems, e.g., in the case of planetary rovers, thereby evaluating the worst possible safety indicators based on constraint violation levels. Although ACE may lead to extreme conservatism, a newer variant proposed in [131] has a favorable computational cost and alleviates the pessimism, while still retaining empirically estimated probabilistic safety guarantees based on the distributions calculated over the states.

Additionally, avoiding skidding and slippage requires proper modeling of the surface and producing sufficient amounts of torque to carry out curvilinear motions based on energy-efficient motion planning. Possible changes to the surface need to be accounted for and incorporated into the model using online learning through, e.g., NNs and Extended Kalman Filter (EKF) for the dynamic and kinematic aspects, respectively. The former has importance in estimating the pose, and the latter in determining the constraints on the turn radius and for producing estimates of energy consumption [20,83].

Alternatively, reasoning about the contact and motion may be handled simultaneously through Mixed-Integer Convex Programming (MICP), which is effective in tackling other complications such as uneven terrain, friction, and certain types of non-convexity [132]. Similarly, the motion within a free space can evolve toward motion while in contact by letting the tree grow in the combined space through factoring it into a belief over the configuration and the contact state [133].

Motion planning for legged robots is generally easier than for wheeled robots, due to the fact that they provide a wider range of mobility. Nevertheless, even in the case of legged robots, duly accounting for all the conditions and uncertainties arising from the terrain's unevenness is not straightforward, since it requires full-body motion plans made within a high-dimensional space based on the limited information available about the terrain [130].

It is also worth noting that collaboration with human operators may make a significant contribution in handling uncertainties, e.g., occlusions disrupting the robot's perception of the target object to be grasped. This can take place either by providing knowledge or taking part in the execution of higher-level tasks, where a tree-shaped set of feasible plans is created using geometric reasoning [134].

3.9. Stability

Uncertainties of various types, including disturbances, inaccuracies of parametric models, and environmental parameters, as well as complex geometry, may affect the performance of a motion planning algorithm [127]. Therefore, path planning and trajectory optimization may need to be followed by feedback control, to ensure stability and to overcome the nonlinear dynamics and uncertainties. The output of such a module is a control policy that stabilizes the system, from a bounded set of initial conditions to a goal state. As a result, a tree of trajectories could be created and stabilized using feedback.

By exploring a bounded set based on random samples, the possible necessity of deriving additional trajectories can be determined. Subsequently, funnels can be found iteratively, until they cover the bounded set, meaning that the control policies can stabilize the bounded set to the goal. The task of approximating the funnels can be performed using, e.g., sums-of-squares verification or through sampling followed by falsification based on simulation [57]. On the other hand, various post-processing procedures can be employed for smoothing and improving a trajectory, as well as replanning to avoid dynamic obstacles using model-based predictions [43]. For example, linear-quadratic-Gaussian controllers yield reliable reference-tracking functionalities [58].

Consideration of the dynamics of a problem can be realized by performing asymptotically optimal KP using two-point BVP-solvers based on optimization and numerical optimal control methods, such as Sequential Quadratic Programming (SQP) [46]. Artificial potential fields realized through, e.g., Improved Rapidly-exploring Random Trees (I-RRT)* constitute a viable alternative [135–137].

With information about the distance from obstacles, using the Monte Carlo (MC) algorithm, CP may be approximated with asymptotic correctness, which enhances the chance of finding a feasible plan.The CP can be found for each iteration for the estimated optimal path, followed by inflating or deflating the obstacles to make the CP reach a target value. Statistical variance-reduction techniques such as Control Variates (CV) and Importance Sampling (IS) contribute to a faster performance or real-time functionality through parallelization [58].

The results of CP approximation need to be transformed to the configuration space, to estimate the location of the points with the highest CP. This could be performed using a Newton method to determine the closest point that may cause a collision [6]. For safety-critical tasks, Control Barrier Function (CBF) can be used to incorporate obstacle-avoidance constraints through Quadratic Programming (QP) [59]. It is also possible to devise a random search tree reconstruction framework by predicting the cost using NNs. This results in asymptotic near-optimality [84].

Non-convexities in the cost functions and obstacles, as well as control and state constraints, are handled in energy-optimal motion planning by decomposing the decision variables and creating a convex representation of collision constraints. This results in two QP problems, accommodating the decoupled Linear Time-Invariant (LTI) dynamics in each axis. Assuming that an initial feasible trajectory exists, the foregoing problem may be solved iteratively using an Alternating Quadratic Programming (AQP) algorithm, which converges to a homotopic local optimum with respect to the initial guess faster than the MICP alternatives [138].

Applying splines to the resulting paths, to generate a possibly complex shape while maintaining parametric continuity, is another approach for improving the quality, in terms of, e.g., reduced lateral accelerations, enhanced robustness against disturbances, or improved tracking performance. In autonomous driving, additional benefits may include enhanced passenger comfort, as well as reduced tire wear and mechanical failure [17].

3.10. Computational Feasibility

Physics-based motion planning, state sampling, and state steering are computationally expensive, due to the costs of state propagation of real-time replanning performance [20], especially for high-DoF robots [119]. High-dimensional dynamics and external disturbances

lead to extremely high computational costs, which are often achieved for efficient, real-time planning by sacrificing dynamical feasibility and safety. Various efforts have been made to come up with a compromise between safety and planning efficiency, resulting in, e.g., the modular Fast and Safe Tracking (FaSTrack) system [139], which can be used for tracking the multi-dimensional paths devised by different trajectory optimizers. On the other hand, using factor graphs and GP interpolation has been widely reported to improve the computation time required for making inferences [122].

Moreover, the computational load of creating a funnel of trajectories may be alleviated using convex optimization, through sums-of-squares programming [127]. For TMP algorithms, this could be leveraged by enabling add-and-remove constraints on motion feasibility at the task level, a successful example of which is the Iteratively Deepened Task and Motion Planning (IDTMP) proposed in [140], offering probabilistic completeness and scalability for plans involving several objects.

PRM* is known to minimize the computational cost of generating RMs. Nevertheless, it may be slow, and result in excessive growth, undermining online query resolution and storage. Incremental sparse sub-graph-based approaches, referred to as spanners, have alternatively been proposed with asymptotic near-optimality. They yield a lower path quality but alleviate the aforementioned shortcomings, and improve the RM density [69]. Examples of computationally heavy functions that increase overheads include checking for nearest neighbors and possible collisions in sampling, while using, e.g., RRT. This may be obviated by utilizing the CBF-RRT algorithm instead.

It has been stated that by discretizing the reachable state space and solving it numerically, the state sampling and nearest-neighbor search speeds can be improved. The former can be performed online and results in a reachable map, which reduces the number of states involved in sampling, as well as unsuccessful motion validity check queries. Moreover, states that are not reachable or can be reached only after a certain horizon are excluded from the nearest-neighbor search [60]. On the other hand, for problems such as motion-specific self-calibration, using EKF may accelerate the process of approximating model parameters or dynamics, to achieve the maximum belief informativeness based on a prescribed budget. This is beneficial in SBMP, which is required to react to certain circumstances promptly.

For a set of robots moving in parallel and with separability, assuming that each of them is expected to move at most d units from their initial location, the distance traveled by each robot is $O(d)$, i.e., constant stretch, meaning that a constant-factor approximation is possible for the optimization problem. Moreover, NP-hardness can be maintained, even if the robot positions are restricted to a regular grid. If separability is not guaranteed, e.g., for densely-packed disks that may not be well separated, the stretch factor increases to $\Omega\left(N^{1/4}\right)$, and with certain considerations, to $O\left(N^{1/2}\right)$ [141].

The distance metric is not necessarily Euclidean, and this choice greatly affects the computational cost of RRT-based motion planning. It also influences the path quality and space coverage. Determining this appropriately, e.g., based on learning, is essential for achieving a reasonable computational cost [32].

Exploiting data-driven approaches for predicting the map in an unknown environment, as opposed to relying on frontier selection heuristics, may help expedite planning [118]. In numerous applications requiring dense mapping of an unknown environment, e.g., for agile robots possessing visual sensors and constrained embedded computational capabilities, leveraging the computational cost may be accomplished by using only parts of the existing sensory information and combining the mapping, i.e., perception, and planning problems. In this manner, by switching between the two iteratively, each of them is updated based on the results of the other [142].

When it comes to problems involving human motion tracking, it is basically assumed that the subject moves according to a plan devised with a cost function. However, such algorithms are prone to computational intractability [143]. A remedy would be to utilize NNs learning near-optimal heuristics from path planning. For instance, the MPNet can learn continually and actively from expert demonstrations. This reduces the amount of

learning data required. By recursively constructing connectable paths, the workspace is encoded based on point cloud measurements. Incorporating classical path planning algorithms for a hybrid approach leads to theoretical worst-case guarantees, while maintaining the computational and optimality advantages [94].

However, it should also be noted that long-term planning using NNs, e.g., applying Deep Reinforcement Learning (DRL) to the Linear Temporal Logic (LTL) or Metric Interval Temporal Logic (MITL) specifications [85], may incur computational drawbacks. This can be alleviated through an MC tree search [86], where Model Predictive Control (MPC) is used under a linear program for convex optimization, to achieve further robustness against noise. This is true for scenarios involving a high number of dynamic objects and may be improved by using Long Short-Term Memory (LSTM) [102].

Parameterized probabilistic models of human behavior can be utilized to predict and prepare for future actions. This allows adapting a model to circumstances and observations, based on a distribution over the model parameters, obviating the necessity for prior determination. Other alternatives include Bayesian inference and worst-case forward reachable sets. The additional computational burden may be leveraged by solving a stochastic reachability problem in the joint space of the human and the belief for the model parameters. These can be deterministically approximated using a Hamilton–Jacobi framework. This approach needs to be implemented allowing only a set of actions, as opposed to the aforementioned distribution, while maintaining the belief as an explicit state.

For robots with numerous continuous states, improved scalability can be achieved by decomposing the problem into smaller sub-problems using a lazy satisfiability modulo theory approach. At each iteration, a coarse discretization of the workspace is carried out, taking into account Boolean constraints, while capturing the low-level continuous dynamics, to produce high-level discrete plans. Iterations are performed until a feasible plan is found. Possible infeasibilities include the transitions between the workspace regions [144].

From the point of view of software development, frameworks realized using utilities such as Motion Planning Templates (MPT) take advantage of compile-time polymorphism. Such approaches may help to more efficiently handle the usual lack of computational resources for small robots running on batteries, by producing individual codes tailored to the specifications of the robot and motion planning problem, which can determine the data structure. Although this deprives the robot of runtime flexibility, it provides extended compile-time capabilities, such as storing robot-specific information in the resulting motion planning graphs, creating firmly packed data structures, and modifying the scalar precision [145].

A common practice is to use a unified representation on the basis of a discrete, finite motion space and to split the problem into task and motion planning components. The former decides what to do, and the latter determines the geometric feasibility based on finer details. Width-based search algorithms and AI planning languages that represent the high-dimensional problem in compact form are essential components of such frameworks. Moreover, some problem representations enable the prediction of constraints and the transfer of knowledge from one instance of the problem to another. For example, the score space proposed in [146] provides a representation according to the performance of solutions applied previously, where similarity indices are used sequentially [147].

Although collision detection is widely believed to be the main computational bottleneck in asymptotically optimal SBMP, a nearest-neighbor search may be computationally even more expensive, depending on the experimental scenario. This has motivated the development of more efficient nearest-neighbor search strategies and data structures that take into account the mission specifications [50].

Given an online preparation of a set of random controls that is large enough, the computational cost may be reduced by finding a balance between exploration and exploitation. Machine learning techniques can further contribute by producing choices of maneuvers based on the dynamics, as well as the heuristics and particularities related to obstacles [16].

On the other hand, incorporating loop-closure constraints may help reduce the configuration space to a manifold within the joint ambient space, which is higher-dimensional [61].

In numerous recent studies, using a many-core Graphics Processing Unit (GPU) has been reported to allow fast motion planning [148,149]. Nevertheless, robot-specific circuitry could be built to achieve scalability in motion planning and parallelization of collision detection over the RM edges [70]. Using semi-infinite nonlinear optimization, the constraints and their gradients are calculated simultaneously, and obtaining polynomial approximations over time helps satisfy the constraints continuously. Parallelizability arises from the fact that the constraints and their gradients are analyzed for each time interval independently. The rest of the computational elements, such as kinematics, dynamics, and geometry, are also essentially parallelizable [149]. Continuous-state Partially Observable Markov Decision Process (POMDP) problems can also be solved faster using a GPU or a hybrid of a GPU and Central Processing Unit (CPU), considering a multi-level formulation of the Monte Carlo Value Iteration (MCVI) method. Balancing the workload and interleaving the data and computations constitute the main backbones of parallelization in this context, where the CPU prepares the data, and the GPU handles the simulations [150].

Last but not least, one may resort to a multi-resolution architecture for managing the depth information characterizing the obstacles, where the range measurements are represented using a 2.5-dimensional projective data structure, namely an egospace, within which motion primitives can also be expressed [151].

4. Applications

Motion planning algorithms are useful in many scenarios involving robots or autonomous cars, e.g., trajectory planning for multi-legged walking robots for planetary exploration, or for autonomous cars inside parking garages, as well as for highly actuated service robots in mobile manipulation [49], and even in precision medicine [75]. In the following sections, applications of motion planning algorithms realized using a certain type of robot will be briefly reviewed, to make their importance in practical contexts more tangible, as well as touching upon some of the relevant open problems.

4.1. Social Robots

Social robots have gained increased popularity. In particular, omnidirectional robots are capable of performing social interactions relatively efficiently, due to their ability to reorient their body independently from the direction of motion. Nevertheless, apart from general objectives typically considered while framing the problem, assimilating a robot into the social context necessitates accounting for additional requirements. Prominently, the robot should behave cooperatively and in a similar manner to human beings, e.g., in crowded or narrow passages. Thus, certain constraints may be derived from human subjects, and imposing these while devising the translation and rotation profiles helps make the robot resemble humans in a more realistic fashion [62,152].

In the absence of vocal clues implying the user's intentions or expectations, a robot should preferably be able to understand and handle a situation by observing the users' behavior and predicting their actions [153]. On the other hand, mimicking and trying to replicate the motion profiles adopted by human beings may contribute to developing socially acceptable and cooperative behaviors for mobile robots, small-sized transportation vehicles, and wheelchairs, thereby improving operational efficiency and safety using an acceleration-based *social force model* [154].

A robot's perceptual and decision-making capabilities should enable it to behave well by, e.g., choosing a reasonable trajectory. Statistical models of density and velocity profiles can be made and integrated into the robot's resources. These are produced by analyzing and representing the subjects' attitudes using, e.g., a Boltzmann factor defined based on the comfort attributed to their distance from walls and their possible preference for walking on a certain side within a pedestrian corridor. This, in turn, provides further insights into

the authenticity and reliability of theoretical and simulation-based pedestrian distributions and collision avoidance models [154,155].

Modeling human behaviors is among the factors that may contribute to better motion planning for social robots. This may not be easy, due to the underlying stochasticity. Nevertheless, a robot should ideally be able to plan and perform its motions such that they are predictable for nearby pedestrians [156]. More clearly, feature-matching techniques, possibly based on maximum entropy, may fail to model the path followed by humans under scenarios involving different subjects or different instances of the same path being followed by a given subject. Therefore, utilizing DRL may be considered to avoid violations of social norms, as well as for achieving motion at normal human walking speed, while avoiding extended computation times [103].

Assuming that a motion is planned and performed in a human-like manner, the above facilitates the prediction of the robot's motions by the human, thereby enabling more convenient joint navigation. Nevertheless, the principles governing Human-Human Interaction (HHI) are unlikely to be readily applicable to Human-Robot Interaction (HRI), due to the different circumstances arising from the usual interest presented by humans in getting to know more about various aspects of a robot's behavior [156].

HRI may be enhanced in an adaptive and online manner in dynamic and unknown environments, through utilizing communicative means of locomotion, where, based on the Artificial Potential Field (APF), the velocity and orientation information is utilized to take advantage of patterns represented by the motions of household animals [72].

It is worth noting that human-likeness is measured based on social spatial attention models indicating the capture time and direction as a consequence of a cue, e.g., a head turn [157]. When it comes to replicating human motions intended to resemble an emotion, analyzing upper-body motions inspired by human gait, along with the trajectory and the center-of-mass, has been shown to have a significant influence from the vertical oscillations of the human subjects' perceptions of the emotional states and the corresponding confidence levels [158].

On the other hand, the motion diversity produced independently from the subject's intentions and incorporated into the android's behavior could contribute to the soundness of the resulting impressions, in terms of the interpersonal skills required for natural interaction [159]. The congruence based on the balancing motions of the robot, which could be achieved by, e.g., a human-like inverted pendulum mechanism, could greatly affect these impressions and swaying [160]. Last but not least, motion planning for persons with reduced mobility, such as those with visual impairment, may require further considerations; e.g., of the constraints imposed based on their capability for handling crowded areas, wet floors or roadblocks, or avoiding undesirable items, such as stairs.

4.2. Self-Driving Cars

In principle, self-driving cars are a type of mobile robot, as they take advantage of similar technology in terms of autonomy. Likewise, the problem of avoiding hitting other cars or pedestrians is, in fact, tantamount to that of avoiding dynamic obstacles dealt with in the motion planning for mobile or humanoid robots [1].

They also need to perceive the world, make maps and update them, and plan their motion with several layers of hierarchy. Nevertheless, unlike mobile robots, which may be utilized under considerably more flexible setups, the motions of self-driving cars are restricted to predefined roads and are further constrained by traffic regulations and good practices of driving, e.g., while navigating intersections [86]. This leads to both advantages and disadvantages in motion planning [1].

Improved convenience, efficiency, accessibility, and safety, as well as reduced overall costs and manpower, constitute the significant advantages of self-driving cars in automotive transportation settings [35], where, from the point of view of urban administration, they also contribute to tackling congestion and reducing emissions [37]. Moreover, due to their

capability for utilizing redundant sensors, they reduce the risk of accidents due to possible late reactions, faulty perceptions, or a lack of attentiveness from the human driver [1].

Tasks for a self-driving car can be extremely safety-critical, and performing them may demand making use of feedback control [35]. Real-time speed control of navigating a self-driving car through an occluded crosswalk may be achieved by modeling the problem as a partially observable Markov decision process and optimizing the longitudinal acceleration using dynamic programming, on the basis of the belief of the crosswalk [161].

In order to enhance the speed for reliable performance under real-world scenarios, rule-template sets may be created. These make it possible to determine an appropriate decision for the traffic scene at hand. The CommonRoad benchmark was proposed in [162], where for each real or artificial experimental scenario, constraints, goals, the cost function, and a vehicle model are provided. For example, autonomous parking is modeled as a dynamic optimization problem and solved using an interior-point simultaneous algorithm [163].

Algorithms such as A^*, which are designed to work in unstructured environments, are useful for handling highly constrained environments where on-road trajectory planning may not suffice. Node expansion algorithms are utilized to obtain trajectories that are subsequently refined using a pure-pursuit controller producing edges, as short motion primitives, which ensure the general adherence of the vehicle to the trajectory. Using this strategy, a vehicle can exploit explorative advantages in challenging scenarios, e.g., reversing to pass another vehicle that suddenly stops.

After extracting motion primitives from real-world driving data, they can be connected, followed by applying Expectation-Maximization (EM) and an initial segmentation. The Dynamic Movement Primitives (DMPs) may then be inferred probabilistically, thereby correlating the independent motion plans for a smooth transition and improving the tracking accuracy [164].

Finally, it should be noted that more complex maneuvers, such as changing lanes, parking, and overtaking, also require modeling the reactions of human drivers and are computationally disproportionately heavy. This could be remedied using prior knowledge obtained by training neural networks [95].

4.3. Humanoid Robots

Humanoid robots are characterized by their relatively high DoF. Classically, their high-level motion plans resulted from path planning, and their trajectory optimization used to be transformed into a dynamic locomotion plan for the whole body in a later stage. More recent approaches have tired to take advantage of simultaneous optimization of the low-level and high-level motion specifications where applicable. This will be further elaborated later in this section. While analyzing the generalized inverse kinematics, the joint limits, foot positions, and balance need to be considered. A motion planner should be able to manipulate massive objects, take appropriate actions with the unforeseen appearances of obstacles in cluttered environments, and plan footsteps by means of variable kinematic modeling of the footholds [165]. On the other hand, state space representations are used to apply the existing SBMP techniques with biased sampling to humanoid robots, followed by steering using inverse kinematics [62].

Motion planning for humanoid robots may be performed using a hierarchical strategy, such that, first, a collision-free path is obtained for the End-Effector (EE), followed by searching for collision-free points in the case of each of the via-points for the elbow, resulting in an asymptotically-optimal raw path consisting of straight lines, further refined for kinematic smoothness and feasibility using an online Cartesian calculator and controller [24].

The Contact-Invariant Optimization (CIO) approach aims to optimize whether or not contact forces should be active during each phase of presenting a behavior, simultaneously with the behavior itself. This could be utilized in conjunction with ensembles of perturbed models to achieve robustness against model uncertainties while performing behaviors such as turning, as well as sideways and forward walking [166]. Moreover, human experience

may be exploited by considering multiple operational phases and optimizing the specific contact configurations separately [2].

Gait primitives for a humanoid robot may be extracted as limit-cycle behaviors, then being arranged into a switched, discrete-time system. Nevertheless, the frequency of switching needs to be limited, in order to achieve fluent performance. This can then be utilized in conjunction with Hybrid Zero Dynamics (HZD) to enhance the stability through dimensional reduction and sums-of-squares programming [167].

Motions limited to the sagittal plane constitute the main element in pick-and-place tasks [168]. A more challenging scenario would be to move toward a target stance pose, and after reaching it, perform a prescribed manipulation task. The complexity arises from requirements such as finding a stable stance pose that not only is collision-free but also leads to a full-body configuration that is robustly maintained during manipulation. Using inverse dynamic reachability maps and assuming that the feet are positioned close to each other on a flat surface, a solution can be found with a success rate dependent on the coverage density of the sampling space. Obviously, the map size should not exceed the memory capacity.

Nevertheless, the map may not offer samples with a high enough variety from the high-dimensional configuration space to handle real-world applications involving, e.g., uneven terrain. Therefore, a paired forward-inverse dynamic reachability map needs to be utilized to achieve a wider modularity, if the robot's kinematic structure allows. This would help achieve a given number of composed configurations, while requiring storing a smaller number of samples. For a given memory capacity, this would lead to a more diverse set of possible configurations. Consequently, taking advantage of whole-body redundancy, a higher level of flexibility would be achieved in planning the motion to reach an end-pose in the presence of constraining factors such as obstacles and uneven terrain [169].

Humanoid robots are widely used in scenarios involving HRI. One of the challenges typically encountered in haptic interactions arises from the inevitable impact of the robot's own motions on the sensory data supposed to represent the user's tactile input. A possible partial remedy to the latter problem could be to estimate and subtract the effect of the robot's motions based on a sequence of joint values, through linearizing the underlying posture sub-spaces [170].

Various strategies using the safety of the operator may be accommodated within a motion planning framework that can be assessed based on, among other things, the technical specification ISO/TS 15066 [171] and the principle of energy absorption by the operator body. In this context, the regulated body movements are simulated for an experimental setup, bearing in mind the fundamental consideration that the relative velocity needs to be predicted, estimated, and bound to improve the operators' safety, by preventing the robot from harming them, according to the prescribed trajectory and the relevant velocity limits [172].

Multi-limbed robots can take part in missions such as climbing. For example, a six-legged robot could climb up two walls, where numerous constraints on parameters such as posture, contact force, and torque need to be taken into account, in order for the robot to be able to perform the task using the friction between its end-effectors and the wall surfaces. One approach to tackle this problem consists in utilizing a NonLinear Programming (NLP) solver. Alternatively, the problem could be decoupled into parts: one concerning the torso posture, and the other dealing with contact forces. The former could be solved by means of NLP or MICP, while the latter needs to be worked out as a series of convex optimization problems. Introducing angled walls, uneven surfaces, and obstacles could be considered as further challenges to expose the robot to [173].

4.4. Object Manipulation

Multi-arm robotic systems require task- and joint-level coordination, where the former is meant to make the arms coordinated with an object or with each other, and the latter is aimed at preventing self-occlusions. The space of free motion may be learned using sparse

non-linear kernel classification approaches. Whether the robots are expected to coordinate with each other to reach a moving object simultaneously or only to perform independent point-to-point motions determines the synchrony requirements, which need to be properly considered, especially for possible transitions between different scenarios [174].

The higher-dimensionality and multi-modality of the configuration and search spaces involved in motion planning problems for multi-object manipulation tasks give rise to additional challenges, due to kinematic and geometric constraints. To tackle different placements of objects, FastForward (FF) planners such as FFRob use multi-query RM structures to assess reachability. They handle rearrangement planning problems using an Extended Action Specification (EAS) representation suitable for conditions involving, e.g., arbitrary predicates. They are capable of solving strip planning problems, e.g., delete-relaxations. More specifically, by means of batch sampling of the manipulation primitives, the motion planning problem is iteratively discretized, resulting in a probabilistically complete planner with the finite runtime required for estimating the distance to goal [71,175]. The primitives may be generated using a graph-search under a linear quadratic minimum time problem [176].

In cluttered environments with objects blocking the path to grasping an object, it may be necessary to move the objects first. This may require robot–object and object–object interactions of a dynamical multibody nature, involving varying object poses, and using algorithms such as KPIECE and p-KPIECE [63,64].

4.5. Cooperation and Multi-Robot Motion Planning

Autonomous systems may benefit greatly from communication with their peers, as well as management centers. This would help them make more efficient decisions. For example, self-driving cars may park close to each other, which saves space, or cooperate with each other to alleviate congestion [1]. Moreover, a self-driving car can be equipped with sensors and processing units enabling it to obtain and share valuable information regarding, e.g., the duration of each color of traffic lights, as well as their locations. Cloud-based networks enabling such communications could help them to calculate an Estimated Time of Arrival (ETA) more efficiently. Similarly, the information could be analyzed to allow an improved understanding of traffic patterns and alternative routes [177].

Although setting up global restrictions on robots' movements based on worst-case scenarios may suffice for ensuring safety, this may result in longer operation times. This could be alleviated by incorporating safety considerations at the motion planning level [172,178].

Multi-Robot (MR) teams are useful where, e.g., a robot's own reachable workspace or payload does not suffice for handling large loads [179], and for missions such as patrolling rooms, or in handling deadlocks, e.g., where a moving obstacle hinders the robot's motion [180]. LTL formulas are widely utilized in this context. To this end, robot dynamics and specifications can be captured for convex and Boolean constraints within a feasibility problem, to be decomposed and solved using, e.g., Satisfiability Modulo Convex (SMC) [181].

Even the aerial transportation of large loads, including cable-suspended, could be achieved using teams of robots, where, e.g., Parametric Dynamic Movement Primitives (PDMPs) are used to coordinate the robots and their manipulators. This could be subsequently transformed into an explicit parameterization of motions using statistical approaches with a Bayesian property, such as Gaussian Process Regression (GPR) [66]. Constraints can be imposed on the relative positions of the robots, where onboard, real-time localization detects the relationships between neighboring robots [42,67].

Hybrid approaches only take advantage of centralized planning at a global level based on a consensus, where distributed controls are applied to individual robots to avoid obstacles and transfer forces to the rest of the robots through the object being manipulated. Typically, a receding horizon planner is employed to achieve velocity control via a convex optimization problem [182]. In cooperative missions, a network topology controller is required to ensure connectivity [73].

Centralized strategies may be adopted under optimal control formulations. Although computationally more expensive, these yield higher quality solutions [183]. They

also improve the robustness against possible changes in the environment, avoid deadlocks, and obviate the necessity of decoupling. This problem may be discretized using, e.g., orthogonal collocation direct transcription, resulting in a large-scale NLP [184].

In unlabeled Multi-Robot Motion Planning (MRMP) scenarios, robots need to be placed interchangeably, such that each of the prescribed positions is filled by one of them. In contrast, in labeled cases, each position is associated with a specific robot. For unit-square robots maneuvering within an environment involving polygonal obstacles, this problem is PSPACE-hard [185].

For optimal formation trajectory tracking, finite-term performance can be achieved by optimizing a cost function in the presence of control and state-variable constraints, as well as invertible conditions, over Euclidean rigid body motions, using Pontryagin's maximum principle for Lie groups. The optimality of the performance index is guaranteed if, and only if, the linear dynamic systems are controllable [186,187].

Complexity issues may be partially alleviated by utilizing a priority assignment algorithm to divide a team of robots into smaller groups. This can be more robustly dealt with using incremental approaches, based on, e.g., Satisfiability Modulo Theories (SMT) solvers. Robots with the same priority are in the same group, and, regarding motion planning, handled simultaneously. Robots taking higher priorities are treated as dynamic obstacles, collisions with which are avoided using a relatively small delay [188]. Priorities are also assigned dynamically to avoid collisions between the robots, using a Minimum Linear Ordering Problem (MLOP).

The total length, i.e., the summation of the lengths of the individual paths can be considered as an optimization criterion. This results in a computational complexity of $\tilde{O}\left(m^4 + m^2 n^2\right)$, with m and n standing for the number of robots and the workspace's total complexity, respectively. The maximum length is OPT$+4m$, where OPT denotes the cost of the optimal solution [189].

Population-based stochastic optimization approaches such as PSO or Biogeography-Based Optimization (BBO), which are inspired by group behaviors observed in wildlife, may be utilized to find a global solution, i.e., to avoid local minima, through position updating strategies applied to diverse populations. The Normalized Step Cost (NSC) can be used to initialize the optimization procedure, which may also benefit from previously optimized parameters. Upon violation of the constraints, the parameters are replaced with those of the solution that is the closest to the query vector [76,77]. Moreover, path-finding through geometrically embedded discrete graphs can be combined with an implicit representation of the RMs for a reliable performance [65].

The metric utilized for finding the nearest neighbors within a certain configuration may have a considerable impact on the perception of connectivity of the RMs, as well as the path quality. Recently, other metrics have been adopted from the field of shape matching. Combining different metrics might lead to a preferable number of vertices within the RMs [190].

When it comes to computations, the cost functions related to collision testing can be enhanced by summarizing the most salient data through probabilistic modeling [44]. Directly searching a dense motion planning RM is run in $O(V \log V + E) \approx O\left(n^2\right)$ time, where n is the number of vertices. For a large n, it can be alleviated through successively shortening the path by making the r-disk sub-graphs increasingly dense over a low-dispersion deterministic sequence.

In MRMP, potential targets are captured using, e.g., visual sensors, where probabilities are modeled using, e.g., a Bayesian likelihood ratio tracker, which is recursively updated as a probability density function. Each robot's motion plan is dynamically determined according to the probability of each target for the corresponding location, and its distance from neighboring robots. Considering the inverse log-likelihood ratio as the temperature, the robot moves along the negative gradient of the temperature surface. The interaction with the local agents is performed using a Lennard–Jones potential, practically maximizing the mutual information between the target state and the measurements [191].

Consensus among the agents can be achieved in diverse ways. In the absence of a leader, distributed consensus protocols may be produced using, e.g., Pontryagin's principle, where the relative information is measured at sampling instants. To tackle the inadequacy in bandwidth, the continuity of the information may be improved using a distributed framework, taking advantage of multiple leaders. Distributed containment protocols are followed, such that the remaining robots converge, in terms of position and velocity, to the convex hull resulting from the leaders. Control gains, communication topologies, and sampling periods are devised independently, which provide further freedom for the controller [192].

5. Discussion and Future Work

We have provided a brief review of motion planning frameworks in the context of robotics, along with some of the essential considerations for their efficient practical utilization. In this section, to give a quick overview of the motion planning algorithms discussed throughout the present review, we categorize them from two different points of view. We also list the most prevalent approaches for each, followed by a discussion of which frameworks have been omitted and why, and then the concluding remarks.

5.1. Categorizations

The first categorization is based on the property of being conventional or heuristic and is shown in Figures 5 and 6, respectively. However, based on the studies covered in the present survey, we have clearly shown that with the emergence of Artificial Neural Network (ANN)s and machine learning algorithms, especially Deep Neural Network (DNN)s, there has been a noticeable transition in the trend of research and development activities within the field of motion planning. Owing to their power and efficiency in solving problems such as optimization and route search, etc., AI-based methods, especially DL, have emerged as an independent category of techniques. Since 2015, they have occupied the largest proportion of related works.

The second categorization is on the basis of being global or local, and this is shown in Figure 7. Global path planning uses the global geographical or a static map to find an optimal path [193,194], whereas local path planning requires a constant supply of sensory information to compute a collision-free path within the robot's immediate vicinity. Unlike local planners, global ones may struggle with real-time performance [195–197]. A combinatory interpretation using the two types of categorization shown in Figure 5 and Figure 7 provides an advantageous insight into which choice of motion planning algorithm would best suit a certain application.

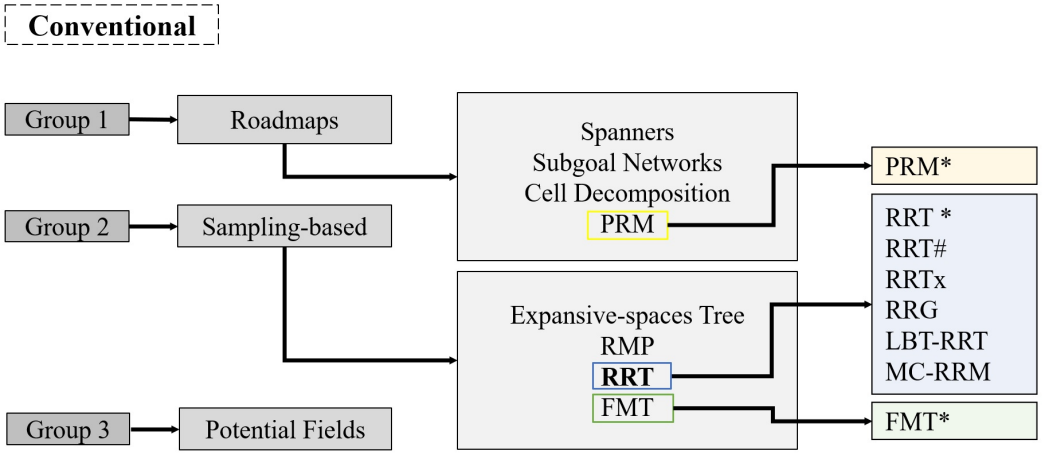

Figure 5. Conventional motion planning frameworks covered in this review.

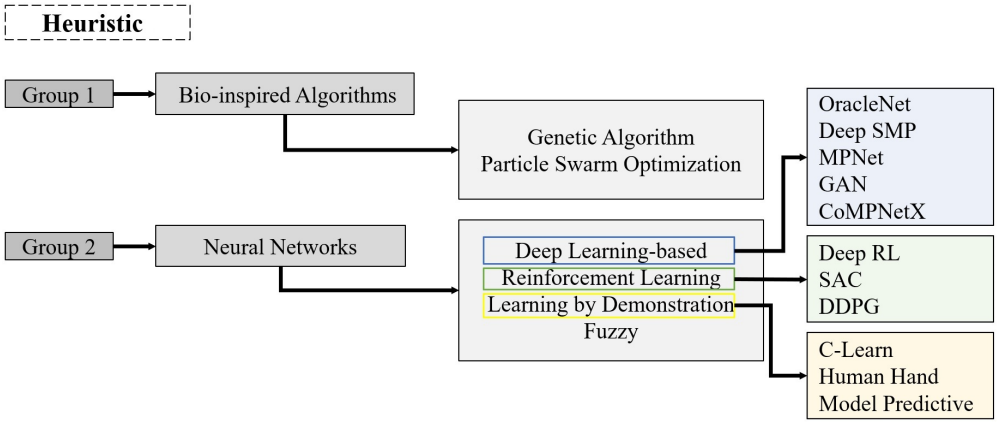

Figure 6. Heuristic motion planning approaches reviewed throughout the present survey.

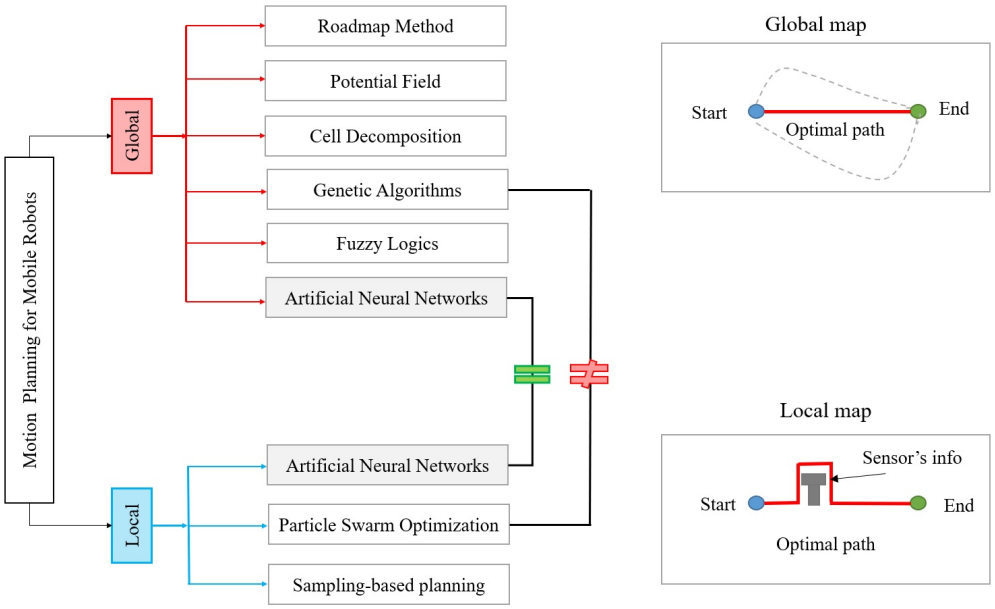

Figure 7. Categorization of the motion planning algorithms discussed in this review based on being global or local.

When it comes to older methods, one may clearly see that over time, heuristic approaches have gained a wider utilization for solving motion design problems. This is shown in the diagrams illustrated in Figure 8, which indicate the percentage usage of conventional and heuristic techniques for three consecutive decades spanning the 1980s through the 2000s. These statistics were taken from [198].

However, in the case of more recent years, the share of motion planning pipelines relying on learning has risen significantly. For DL networks, RL, and DBL, this has changed from 6% to 27%. This has taken place from the period 2010–2015 to 2015–2022 [5]. The ratios of the utilization of these strategies are shown, along with those of heuristic and conventional approaches in Figure 9. A more detailed representation is shown in Figure 10, which indicates the specific ratios for, not only learning-based methods, but also for the bio-inspired, potential field, and sampling-based ones. It should be noted that Figure 10 was adopted from [5], which was also the source of the above statistics.

These figures demonstrate the remarkable pace at which the category of DNN-based motion planning systems has been expanding with respect to the others, and it may well continue doing so in the coming years. While classical methods seem to have maintained their share, classical heuristic structures appear to be rapidly losing their competitiveness against learning-based mechanisms. This can be attributed to learning-based mechanisms'

strong robustness and scalability, which enables them to handle the challenging motion-planning requirements arising from the growing complexities of modern robots, as well as those of the environments they are expected to operate within [199].

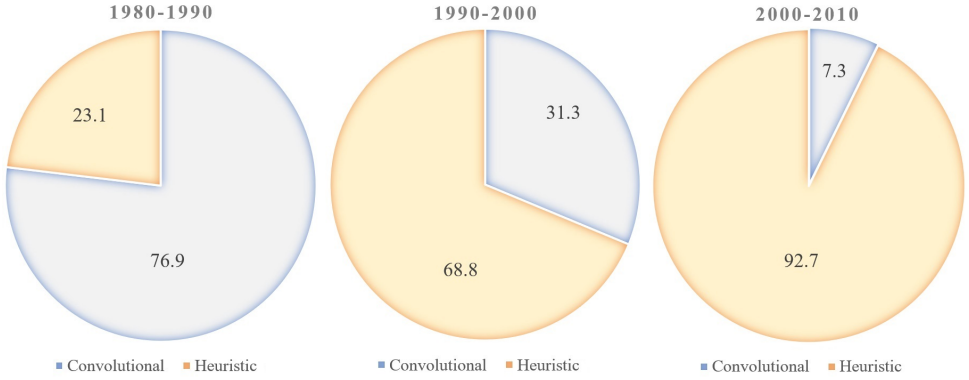

Figure 8. Visual comparison of percentages, indicating the trend of heuristic motion planning methods surpassing conventional ones over time. These values were taken from [198]. The diagrams cover the period from 1980 until 2010.

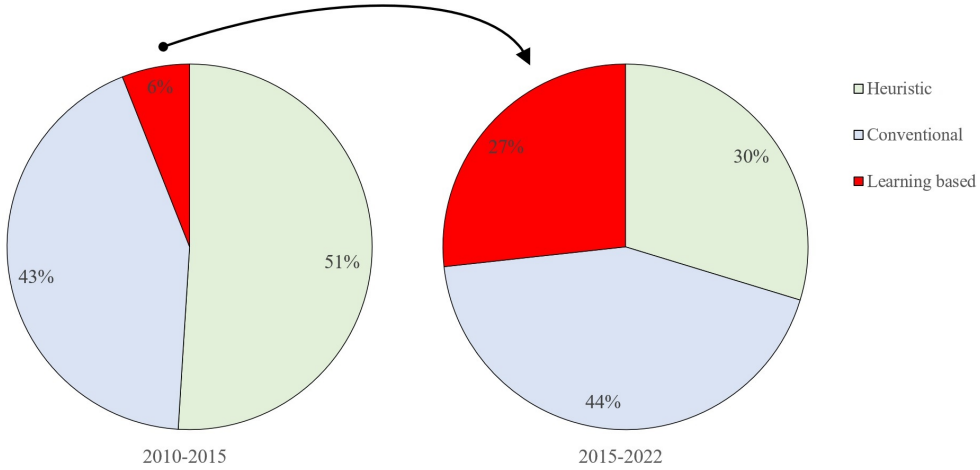

Figure 9. The trend in use of learning-based motion planning frameworks in recent years compared to heuristic and conventional methods, represented as percentages. More detailed statistics are shown in Figure 10.

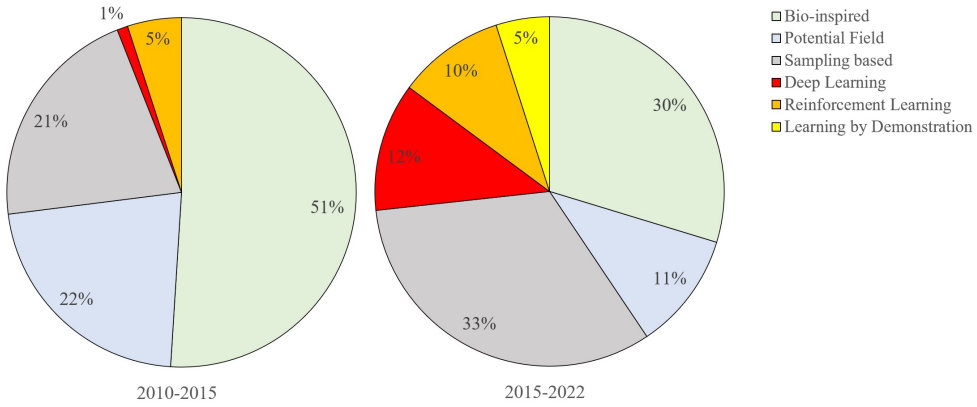

Figure 10. The trend in popularity of bio-inspired, potential field, and sampling-based motion planning frameworks compared to learning-based techniques taking advantage of DL, RL, and DBL, shown as percentages. The figure was adopted from [5].

It should be noted that in this survey, for the sake of coherence and maintaining brevity, not all motion planning pipelines have been covered. More clearly, to enable an in-depth analysis of the current state of the art and its shortcomings, this review has focused on frameworks of ongoing significance in practical contexts, where this judgment was made based on the frequency of practical usage of each technique in the recent robotics literature. More comprehensive categorizations of conventional and heuristic motion planning approaches are shown in Figures 11 and 12, respectively.

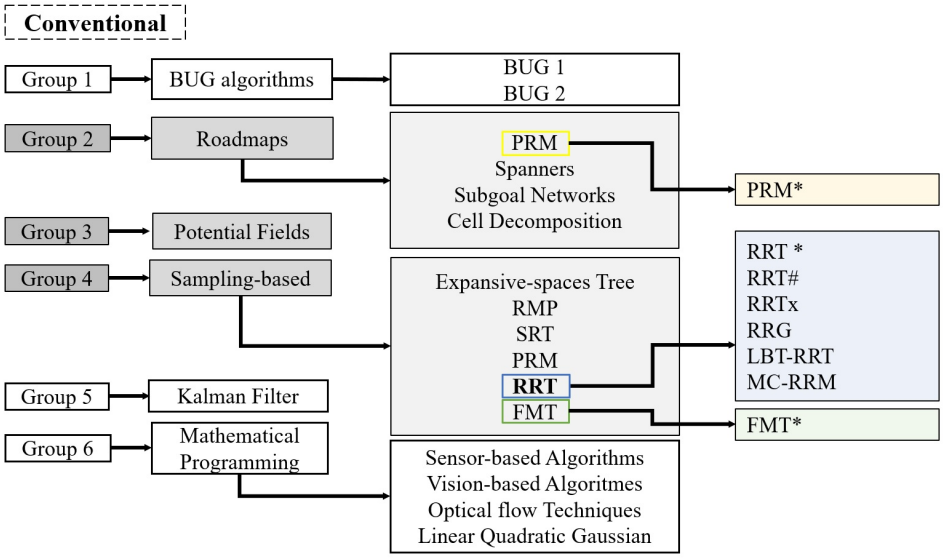

Figure 11. A more comprehensive categorization of conventional motion planning algorithms, with those not covered in the present survey shown within boxes with a plain background. Specifically, BUG algorithms, Kalman filter, mathematical programming, and geometric-based approaches were studied in this survey.

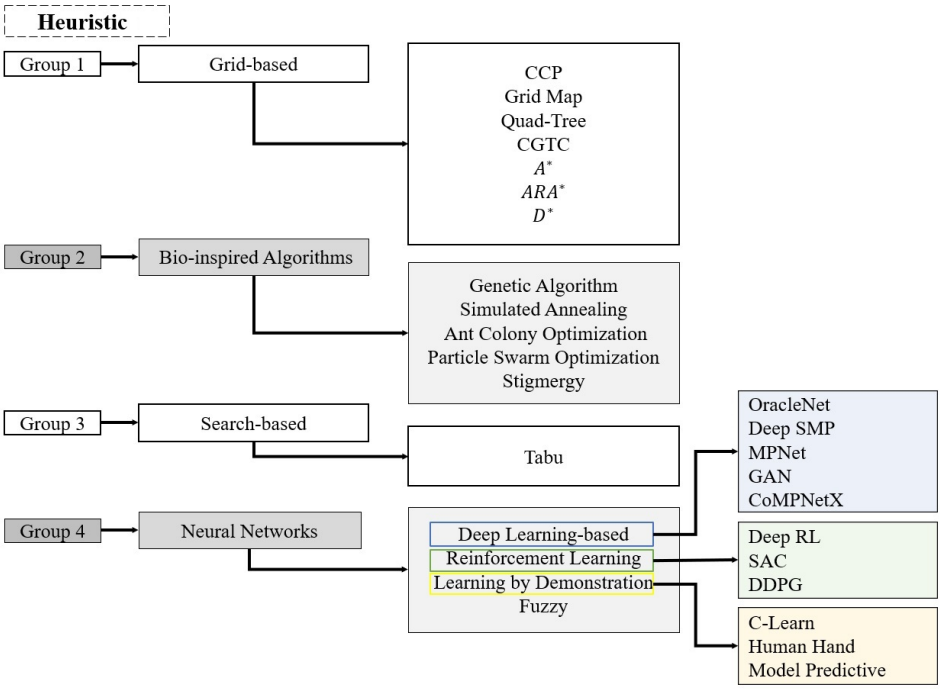

Figure 12. A more inclusive categorization of heuristic motion planning techniques, including grid- and search-based methods, which were not explored in the present review and are placed within boxes with a plain background in the figure.

5.2. Categories Excluded from the Review

For quick reference, each category of methods excluded from the survey will be concisely discussed.

The BUG 1 algorithm is based on the idea of having a robot follow an obstacle's edge, i.e., its contour, until it reaches a full cycle around the obstacle. Meanwhile, the distance to the goal is constantly updated to find the point on the edge which minimizes it. This point will then be used to start the next move toward the goal, which will take place within the next cycle. BUG 2 is faster, as it does not require the robot to undertake a full cycle and instead, makes it depart from the obstacle as soon as the slope of the line from the robot to the goal is the same as that of the line connecting the initial position to the goal [78].

Kalman Filter (KF) is useful in contexts where the tasks of map building and motion planning need to be handled simultaneously through scanning the terrain. Doing so requires obtaining depth information using detectors, where the uncertainties are handled by fusing the data. A fuzzy adaptive KF may help update the map and hierarchically optimize the scanning procedure according to the terrain type by adjusting the filter gain [200].

In mathematical programming, obstacle-avoidance constraints are modeled as inequalities imposed on the configuration parameters, where due to the underlying nonlinearity and the high number of obstacles, typically, a numerical approach is employed to solve the resulting optimization problem [198].

Grid-based motion planning covers a diverse range of algorithms. Complete Coverage Planning (CCP) considers the whole mapping area and the grid-based coverage method is the most popular example, which is suitable for indoor environments. However, its memory usage may be significantly high, due to the fact that the map resolution is not adjusted with respect to the actual environment's complexity, including the shapes and positions of the obstacles. Moreover, it is prone to nonoptimality, due to possible inefficient organization of the cells [201]. Other variants of grid-based algorithms include quadtree [202], Circle Grid Trajectory Cell (CGTC) [203], A* [203,204], Anytime Repairing A* (ARA*) [205,206], and D* [207].

Search-based algorithms work based on a neighborhood search over several waypoints, aiming at real-time, simultaneous map building and path planning. For example, in Tabu search, to avoid becoming stuck in local minima, a Tabu list is made to store nodes that have been evaluated previously [208].

5.3. Future Work

Various applications of motion planning are worthy of further investigation and are still at a primitive stage. Therefore, in what follows, some open challenges that need to be more rigorously addressed in upcoming studies will be concisely discussed, to provide insights into potential future research directions

Although some studies have suggested projecting the problem of motion planning in unstructured environments onto the 2D domain, this may not be practically sufficient. On the other hand, dealing with the problem in the actual 3D domain might fail to accommodate all requirements [20]. Therefore, it is recommended to seek solutions that provide a compromise between the two, using high-level motion primitives in an alternative space, thereby tracing down lower-level characteristics, to devise more detailed configurations over the preliminary outcomes projected onto the other space.

As far as intention-aware and DBL motion planning techniques are concerned, a robot should ideally resemble the intended patterns as closely as possible and perform robustly in the presence of possible errors. This is also the case for self-driving cars [88], which are expected to present a human-like driving behavior. Therefore, larger and more comprehensive representation databases offering more inclusive samples that resemble appropriate reactions under different situations may make of great contribution to a robotic system's fluency.

For example, self-driving cars possess limited sensory equipment but are meant to tackle occlusions in highly cluttered urban areas, while ensuring convenience and driving fluency [25]. Thus, further investigation and development are needed to devise motion planning standards that prevent over-cautiousness but do not undermine safety.

Even in the case of humanoid robots, their performance shows a significant depreciation in practical situations compared to simulated environments, even in the presence of local feedback drawn from an optimizer. In extreme cases, they may be unable to physically meet practical demands, where e.g., trying to apply the simulated trajectory could break the robot apart [166]. This indicates the importance of the balance between the advancements targeted within the two realms, i.e., the theoretical and practical sides.

Avoiding collisions in Human-Robot Collaboration (HRC), e.g., in manufacturing, robotic assembly, and robotic disassembly, requires keeping a distance between humans and robots. This may undermine performance and prevent the robot from reaching its goal quickly enough. Context awareness helps a robot to realize the human operator's intentions and enables it to ensure safety within shorter distances. Human pose recognition and collision sensing calibration are among the factors that could contribute to obtaining such an awareness [209]. On the other hand, the size of dynamic safety zones and bounding volumes may be minimized online according to torque constraints and robot dynamics, where the possibility of collision is assessed and avoided using smooth stop trajectories based on intersections, thereby optimizing the robot's speed [210,211]. These concepts play a critical role in improving efficiency and fluency, while ensuring safety, and are worth further investigation and development in upcoming studies.

Finally, shape and motion both play major roles in contacts between a robot and other objects and agents. Although this could be considered at the design or control stages of motion planning; in the literature, they are not usually considered simultaneously, i.e., through a synergy [212], which would be worth dedicating further effort to in the course of future work.

Funding: This work was partly supported by the Center for Research-based Innovation SmartForest: Bringing Industry 4.0 to the Norwegian forest sector (NFR SFI project no. 309671, smartforest.no); the European Social Fund via IT Academy programme; and the Estonian Research Council (projects no. COVSG24 and PSG605).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Acronyms

ACE	Approximate Clearance Evaluation
AGV	Autonomous Guided Vehicle
AI	Artificial Intelligence
ANN	Artificial Neural Network
APF	Artificial Potential Field
AQP	Alternating Quadratic Programming
ARA*	Anytime Repairing A*
ATA	Aggressive Turn-Around
BBO	Biogeography-Based Optimization
BIT	Batch-Informed Trees
B-spline	Basis spline
BTT	BoTtleneck Tree

BVP	Boundary Value Problem
CAM	Computer Aided Manufacturing
CBF	Control Barrier Function
CCP	Complete Coverage Planning
CGTC	Circle Grid Trajectory Cell
CIO	Contact-Invariant Optimization
CNC	Computer Numerical Control
CNN	Convolutional Neural Network
CoMPNetX	Constrained Motion planning Networks x
CP	Collision Probability
CPU	Central Processing Unit
CRISP	Continuum Reconfigurable Incisionless Surgical Parallel
CV	Control Variates
CVE	Curvature Variation Energy
DBL	Demonstration-Based Learning
DDPG	Deep Deterministic Policy Gradient
DeepSMP	Deep Sampling-based Motion Planner
DGMP	Demonstration-Guided Motion Planning
DL	Deep Learning
DMPs	Dynamic Movement Primitives
DNN	Deep Neural Network
DoF	Degrees of Freedom
DP	Dynamic Programming
DRL	Deep Reinforcement Learning
EAS	Extended Action Specification
EASE	Exploitation of Abstract Symmetry of Environments
EE	End-Effector
EKF	Extended Kalman Filter
EM	Expectation-Maximization
ETA	Estimated Time of Arrival
FaSTrack	Fast and Safe Tracking
FCN	Fully Convolutional neural Network
FEA	Finite Element Analysis
FF	FastForward
FMT	Fast Marching Tree
GA	Genetic Algorithm
GAN	Generative Adversarial Network
GP	Gaussian Process
GPM	Gauss Pseudospectral Method
GPR	Gaussian Process Regression
GPU	Graphics Processing Unit
HER	Hindsight Experience Replay
HFR	High-Frequency Replanning
HHI	Human-Human Interaction
HRC	Human-Robot Collaboration
HRI	Human-Robot Interaction
HZD	Hybrid Zero Dynamics
ICS	Inevitable Collision State
IDTMP	Iteratively Deepened Task and Motion Planning
i.i.d.	independently and identically distributed
INVM	Infinity-Norm Velocity Minimization
IRL	Inverse Reinforcement Learning
I-RRT	Improved Rapidly-exploring Random Trees
IS	Importance Sampling
KF	Kalman Filter
kNN	k-Nearest-Neighbor
KP	Kinodynamic motion Planning
KPIECE	Kinodynamic motion Planning by Interior-Exterior Cell Exploration

LBT	Lower Bound Tree
LPV	Linear Parameter Varying
LRPP	Learned Reactive Planning Problem
LSPI	Least Squares Policy Iteration
LSTM	Long Short-Term Memory
LTI	Linear Time-Invariant
LTL	Linear Temporal Logic
MC	Monte Carlo
MC-RRM	Multi-Component Rapidly-exploring RoadMap
MCVI	Monte Carlo Value Iteration
MICP	Mixed-Integer Convex Programming
MITL	Metric Interval Temporal Logic
MLOP	Minimum Linear Ordering Problem
MPC	Model Predictive Control
MPNet	Motion Planning Networks
MPT	Motion Planning Templates
MR	Multi-Robot
MRMP	Multi-Robot Motion Planning
MSTTMR	Multi-Steering Tractor-Trailer Mobile Robot
NLP	NonLinear Programming
NN	Neural Network
NSC	Normalized Step Cost
PER	Prioritized Experience Replay
PDMPs	Parametric Dynamic Movement Primitives
POMDP	Partially Observable Markov Decision Process
pose	position and orientation
PPO	Proximal Policy Optimization
PSO	Particle Swarm Optimization
QP	Quadratic Programming
RBF	Radial Basis Function
RKHSs	Reproducing Kernel Hilbert Spaces
RL	Reinforcement Learning
RM	RoadMap
RMP	Repetitive Motion Planning
RRG	Rapidly-exploring Random Graph
PRM	Probabilistic RoadMap
RNN	Recurrent Neural Network
RRT	Rapidly-exploring Random Trees
SAC	Soft Actor Critic
SLAM	Simultaneous Localization And Mapping
SBMP	Sampling-Based Motion Planning
SMC	Satisfiability Modulo Convex
SMT	Satisfiability Modulo Theories
SQP	Sequential Quadratic Programming
STOMP	Stochastic Trajectory Optimizer for Motion Planning
SyCLoP	Synergistic Combination of Layers of Planning
TD	Temporal-Difference
TMP	Task and Motion Planning
UAV	Unmanned Aerial Vehicle
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VRU	Vulnerable Road User

References

1. Kala, R. *On-Road Intelligent Vehicles: Motion Planning for Intelligent Transportation Systems*; Butterworth-Heinemann: Oxford, UK, 2016.
2. Liu, C.; Atkeson, C.G.; Feng, S.; Xinjilefu, X. Full-body motion planning and control for the car egress task of the DARPA robotics challenge. In Proceedings of the 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), Seoul, South Korea, 3–5 November 2015; pp. 527–532.

3. Halperin, D.; Salzman, O.; Sharir, M. Algorithmic motion planning. In *Handbook of Discrete and Computational Geometry*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2017; pp. 1311–1342.
4. Mac, T.T.; Copot, C.; Tran, D.T.; De Keyser, R. Heuristic approaches in robot path planning: A survey. *Robot. Auton. Syst.* **2016**, *86*, 13–28. [CrossRef]
5. Tamizi, M.G.; Yaghoubi, M.; Najjaran, H. A review of recent trend in motion planning of industrial robots. *Int. J. Intell. Robot. Appl.* **2023**, *7*, 253–274. [CrossRef]
6. Sun, W.; Torres, L.G.; Van Den Berg, J.; Alterovitz, R. Safe motion planning for imprecise robotic manipulators by minimizing probability of collision. In *Robotics Research*; Springer: Cham, Switzerland, 2016; pp. 685–701.
7. Roozegar, M.; Mahjoob, M.; Jahromi, M. Optimal motion planning and control of a nonholonomic spherical robot using dynamic programming approach: Simulation and experimental results. *Mechatronics* **2016**, *39*, 174–184. [CrossRef]
8. Nguyen, Q.V.; Colas, F.; Vincent, E.; Charpillet, F. Motion planning for robot audition. *Auton. Robot.* **2019**, *43*, 2293–2317. [CrossRef]
9. Mccrackin, D.C.; Johnson, S.W. Syntactic Inferential Motion Planning Method for Robotic Systems. U.S. Patent 9,855,657, 2 January 2018.
10. Yang, L.; Qi, J.; Song, D.; Xiao, J.; Han, J.; Xia, Y. Survey of robot 3D path planning algorithms. *J. Control Sci. Eng.* **2016**, *2016*, 7426913. [CrossRef]
11. Gillani, M.; Akbari, A.; Rosell, J. Physics-based motion planning: Evaluation criteria and benchmarking. In Proceedings of the Robot 2015: Second Iberian Robotics Conference, Lisbon, Portugal, 19–21 November 2015; pp. 43–55.
12. Gasparetto, A.; Boscariol, P.; Lanzutti, A.; Vidoni, R. Path planning and trajectory planning algorithms: A general overview. In *Motion and Operation Planning of Robotic Systems*; Springer: Cham, Switzerland, 2015; pp. 3–27.
13. Contreras-Cruz, M.A.; Ayala-Ramirez, V.; Hernandez-Belmonte, U.H. Mobile robot path planning using artificial bee colony and evolutionary programming. *Appl. Soft Comput.* **2015**, *30*, 319–328. [CrossRef]
14. Holladay, R.; Lozano-Pérez, T.; Rodriguez, A. Force-and-Motion Constrained Planning for Tool Use. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019.
15. Schmerling, E.; Janson, L.; Pavone, M. Optimal sampling-based motion planning under differential constraints: The driftless case. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 2368–2375.
16. Sivaramakrishnan, A.; Littlefield, Z.; Bekris, K.E. Towards Learning Efficient Maneuver Sets for Kinodynamic Motion Planning. *arXiv* **2019**, arXiv:1907.07876.
17. Elbanhawi, M. Randomised Parameterisation Motion Planning for Autonomous Cars. Ph.D. Thesis, RMIT University, Melbourne, Australia, 2016.
18. Simon, B.; Riegl, P.; Gaull, A. Motion Planning With Flexible Trajectory Chains. In Proceedings of the 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES), Madrid, Spain, 12–14 September 2018; pp. 1–8.
19. Geng, J.; Arakelian, V. Design of Partially Balanced Planar 5R Symmetrical Parallel Manipulators via an Optimal Motion Planning. In Proceedings of the IFToMM World Congress on Mechanism and Machine Science, Krakow, Poland, 15–18 July 2019; pp. 2211–2220.
20. Ferri, F. Computing Fast Search Heuristics for Physics-based Mobile Robot Motion Planning. 2018. Available online: https://iris.uniroma1.it/handle/11573/1137741 (accessed on 3 April 2023).
21. Vidal, E.; Moll, M.; Palomeras, N.; Hernández, J.D.; Carreras, M.; Kavraki, L.E. Online multilayered motion planning with dynamic constraints for autonomous underwater vehicles. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 8936–8942.
22. Choudhury, S.; Srinivasa, S.S. A Bayesian Active Learning Approach to Adaptive Motion Planning. In Proceedings of the International Symposium on Robotics Research, Puerto Varas, Chile, 11–14 December 2017.
23. Bhardwaj, M.; Boots, B.; Mukadam, M. Differentiable Gaussian process motion planning. *arXiv* **2019**, arXiv:1907.09591.
24. Nguyen, P.D.; Hoffmann, M.; Pattacini, U.; Metta, G. A fast heuristic Cartesian space motion planning algorithm for many-DoF robotic manipulators in dynamic environments. In Proceedings of the 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), Cancun, Mexico, 15–17 November 2016; pp. 884–891.
25. Naumann, M.; Konigshof, H.; Lauer, M.; Stiller, C. Safe but not Overcautious Motion Planning under Occlusions and Limited Sensor Range. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 140–145.
26. Ravanbakhsh, H.; Laine, F.; Seshia, S.A. Real-time Funnel Generation for Restricted Motion Planning. *arXiv* **2019**, arXiv:1911.01532.
27. Minguez, J.; Lamiraux, F.; Laumond, J.P. Motion planning and obstacle avoidance. In *Springer Handbook of Robotics*; Springer: Cham, Switzerland, 2016; pp. 1177–1202.
28. Janson, L.; Schmerling, E.; Clark, A.; Pavone, M. Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. *Int. J. Robot. Res.* **2015**, *34*, 883–921. [CrossRef]
29. Rehan, M.; Reyhanoglu, M.; McClamroch, H. Motion planning for a knife-edge on the surface of a hyperboloid. In Proceedings of the 2017 11th Asian Control Conference (ASCC), Gold Coast, QLD, Australia, 17–20 December 2017; pp. 1326–1330.
30. Marin-Plaza, P.; Hussein, A.; Martin, D.; Escalera, A.D.L. Global and local path planning study in a ros-based research platform for autonomous vehicles. *J. Adv. Transp.* **2018**, *2018*, 6392697. [CrossRef]
31. Ye, G.; Alterovitz, R. Guided motion planning. In *Robotics Research*; Springer: Cham, Switzerland, 2017; pp. 291–307.

32. Palmieri, L.; Arras, K.O. Distance metric learning for RRT-based motion planning with constant-time inference. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 637–643.
33. Ramanagopal, M.S.; Phu-Van Nguyen, A.; Le Ny, J. A motion planning strategy for the active vision-based mapping of ground-level structures. *IEEE Trans. Autom. Sci. Eng.* **2017**, *15*, 356–368. [CrossRef]
34. Francis, G.; Ott, L.; Ramos, F. Stochastic functional gradient for motion planning in continuous occupancy maps. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3778–3785.
35. Paden, B.; Čáp, M.; Yong, S.Z.; Yershov, D.; Frazzoli, E. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Trans. Intell. Veh.* **2016**, *1*, 33–55. [CrossRef]
36. González, D.; Pérez, J.; Milanés, V.; Nashashibi, F. A review of motion planning techniques for automated vehicles. *IEEE Trans. Intell. Transp. Syst.* **2015**, *17*, 1135–1145. [CrossRef]
37. Katrakazas, C.; Quddus, M.; Chen, W.H.; Deka, L. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transp. Res. Part C Emerg. Technol.* **2015**, *60*, 416–442. [CrossRef]
38. Kuffner, J.J.; LaValle, S.M. RRT-connect: An efficient approach to single-query path planning. In Proceedings of the Proceedings 2000 ICRA, Millennium Conference, IEEE International Conference on Robotics and Automation, Symposia Proceedings (Cat. No. 00CH37065), San Francisco, CA, USA, 24–28 April 2000; Volume 2, pp. 995–1001.
39. Boscariol, P.; Gasparetto, A.; Scalera, L. Path Planning for Special Robotic Operations. In *Robot Design: From Theory to Service Applications*; Carbone, G., Laribi, M.A., Eds.; Springer International Publishing: Cham, Switzerland, 2023; Volume 123, pp. 69–95.
40. Karaman, S.; Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **2011**, *30*, 846–894. [CrossRef]
41. Otte, M.; Frazzoli, E. RRTX: Asymptotically optimal single-query sampling-based motion planning with quick replanning. *Int. J. Robot. Res.* **2016**, *35*, 797–822. [CrossRef]
42. Okadome, Y.; Nakamura, Y.; Ishiguro, H. Sampling-Based Motion Planning with a Prediction Model using Fast Gaussian Process Regression. *Electron. Commun. Jpn.* **2017**, *100*, 24–34. [CrossRef]
43. Ma, L.; Xue, J.; Kawabata, K.; Zhu, J.; Ma, C.; Zheng, N. Efficient sampling-based motion planning for on-road autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 1961–1976. [CrossRef]
44. Knepper, R.A.; Mason, M.T. Realtime informed path sampling for motion planning search. In *Robotics Research*; Springer: Cham, Switzerland, 2017; pp. 401–417.
45. Salzman, O.; Halperin, D. Asymptotically near-optimal RRT for fast, high-quality motion planning. *IEEE Trans. Robot.* **2016**, *32*, 473–483. [CrossRef]
46. Xie, C.; van den Berg, J.; Patil, S.; Abbeel, P. Toward asymptotically optimal motion planning for kinodynamic systems using a two-point boundary value problem solver. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 4187–4194.
47. Tang, Y.; You, R.; Kan, H.; Tithi, J.J.; Ganapathi, P.; Chowdhury, R.A. Cache-oblivious wavefront: Improving parallelism of recursive dynamic programming algorithms without losing cache-efficiency. In Proceedings of the ACM SIGPLAN Notices, San Francisco, CA, USA, 7–11 February 2015; Volume 50, pp. 205–214.
48. Janson, L.; Ichter, B.; Pavone, M. Deterministic sampling-based motion planning: Optimality, complexity, and performance. *Int. J. Robot. Res.* **2018**, *37*, 46–61. [CrossRef]
49. Klemm, S.; Oberländer, J.; Hermann, A.; Roennau, A.; Schamm, T.; Zollner, J.M.; Dillmann, R. RRT*-Connect: Faster, asymptotically optimal motion planning. In Proceedings of the 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, China, 6–9 December 2015; pp. 1670–1677.
50. Kleinbort, M.; Salzman, O.; Halperin, D. Collision detection or nearest-neighbor search? On the computational bottleneck in sampling-based motion planning. *arXiv* **2016**, arXiv:1607.04800.
51. Kleinbort, M.; Solovey, K.; Bonalli, R.; Bekris, K.E.; Halperin, D. RRT2.0 for Fast and Optimal Kinodynamic Sampling-Based Motion Planning. *arXiv* **2019**, arXiv:1909.05569.
52. Solovey, K.; Kleinbort, M. The critical radius in sampling-based motion planning. *Int. J. Robot. Res.* **2020**, *39*, 266–285. [CrossRef]
53. Joshi, S.S.; Tsiotras, P. Relevant Region Exploration On General Cost-maps For Sampling-Based Motion Planning. *arXiv* **2019**, arXiv:1910.05361.
54. Harper, M.Y.; Ordonez, C.; Collins, E.G.; Erlebacher, G. Parallel approach to motion planning in uncertain environments. In Proceedings of the Unmanned Systems Technology XX International Society for Optics and Photonics, Orlando, FL, USA, 15–19 April 2018; Volume 10640, p. 106400H.
55. Sun, W.; Patil, S.; Alterovitz, R. High-frequency replanning under uncertainty using parallel sampling-based motion planning. *IEEE Trans. Robot.* **2015**, *31*, 104–116. [CrossRef]
56. Vonásek, V.; Saska, M.; Winkler, L.; Přeučil, L. High-level motion planning for CPG-driven modular robots. *Robot. Auton. Syst.* **2015**, *68*, 116–128. [CrossRef]
57. Reist, P.; Preiswerk, P.; Tedrake, R. Feedback-motion-planning with simulation-based LQR-trees. *Int. J. Robot. Res.* **2016**, *35*, 1393–1416. [CrossRef]
58. Janson, L.; Schmerling, E.; Pavone, M. Monte Carlo motion planning for robot trajectory optimization under uncertainty. In *Robotics Research*; Springer: Cham, Switzerland, 2018; pp. 343–361.
59. Yang, G.; Vang, B.; Serlin, Z.; Belta, C.; Tron, R. Sampling-based Motion Planning via Control Barrier Functions. *arXiv* **2019**, arXiv:1907.06722.

60. Pendleton, S.D.; Liu, W.; Andersen, H.; Eng, Y.H.; Frazzoli, E.; Rus, D.; Ang, M.H. Numerical approach to reachability-guided sampling-based motion planning under differential constraints. *IEEE Robot. Autom. Lett.* **2017**, *2*, 1232–1239. [CrossRef]
61. Jaillet, L.; Porta, J.M. Path planning with loop closure constraints using an atlas-based RRT. In *Robotics Research*; Springer: Cham, Switzerland, 2017; pp. 345–362.
62. Yang, Y.; Ivan, V.; Merkt, W.; Vijayakumar, S. Scaling sampling-based motion planning to humanoid robots. In Proceedings of the 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO), Qingdao, China, 3–7 December 2016; pp. 1448–1454.
63. Sucan, I.A.; Kavraki, L.E. A sampling-based tree planner for systems with complex dynamics. *IEEE Trans. Robot.* **2011**, *28*, 116–131. [CrossRef]
64. Muhayyuddin; Moll, M.; Kavraki, L.; Rosell, J. Randomized physics-based motion planning for grasping in cluttered and uncertain environments. *IEEE Robot. Autom. Lett.* **2017**, *3*, 712–719. [CrossRef]
65. Solovey, K.; Salzman, O.; Halperin, D. Finding a needle in an exponential haystack: Discrete RRT for exploration of implicit roadmaps in multi-robot motion planning. *Int. J. Robot. Res.* **2016**, *35*, 501–513. [CrossRef]
66. Kim, H.; Lee, H.; Choi, S.; Noh, Y.K.; Kim, H.J. Motion planning with movement primitives for cooperative aerial transportation in obstacle environment. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 2328–2334.
67. Spurny, V.; Petrlik, M.; Vonasek, V.; Saska, M. Cooperative transport of large objects by a pair of unmanned aerial systems using sampling-based motion planning. In Proceedings of the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, 10–13 September 2019; pp. 955–962.
68. Valencia, R.; Andrade-Cetto, J. Path planning in belief space with pose SLAM. In *Mapping, Planning and Exploration with Pose SLAM*; Springer: Cham, Switzerland, 2018; pp. 53–87.
69. Marble, J.D.; Bekris, K.E. Asymptotically near-optimal is good enough for motion planning. In *Robotics Research*; Springer: Cham, Switzerland, 2017; pp. 419–436.
70. Murray, S.; Floyd-Jones, W.; Qi, Y.; Sorin, D.J.; Konidaris, G. Robot Motion Planning on a Chip. In Proceedings of the Robotics: Science and Systems, Ann Arbor, MI, USA, 18–22 June 2016.
71. Garrett, C.R.; Lozano-Pérez, T.; Kaelbling, L.P. Ffrob: An efficient heuristic for task and motion planning. In *Algorithmic Foundations of Robotics XI*; Springer: Cham, Switzerland, 2015; pp. 179–195.
72. Kovács, B.; Szayer, G.; Tajti, F.; Burdelis, M.; Korondi, P. A novel potential field method for path planning of mobile robots by adapting animal motion attributes. *Robot. Auton. Syst.* **2016**, *82*, 24–34. [CrossRef]
73. Di, B.; Zhou, R.; Duan, H. Potential field based receding horizon motion planning for centrality-aware multiple UAV cooperative surveillance. *Aerosp. Sci. Technol.* **2015**, *46*, 386–397. [CrossRef]
74. Roy, A.G.; Rakshit, P. Motion planning of non-holonomic wheeled robots using modified bat algorithm. In *Nature-Inspired Algorithms for Big Data Frameworks*; IGI Global: Hershey, PA, USA, 2019; pp. 94–123.
75. Meng, K.; Jia, Y.; Yang, H.; Niu, F.; Wang, Y.; Sun, D. Motion Planning and Robust Control for the Endovascular Navigation of a Microrobot. *IEEE Trans. Ind. Inform.* **2019**, *16*, 4557–4566. [CrossRef]
76. Kim, J.J.; Lee, J.J. Trajectory optimization with particle swarm optimization for manipulator motion planning. *IEEE Trans. Ind. Inform.* **2015**, *11*, 620–631. [CrossRef]
77. Mo, H.; Xu, L. Research of biogeography particle swarm optimization for robot path planning. *Neurocomputing* **2015**, *148*, 91–99. [CrossRef]
78. Campbell, S.; O'Mahony, N.; Carvalho, A.; Krpalkova, L.; Riordan, D.; Walsh, J. Path planning techniques for mobile robots a review. In Proceedings of the 2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE), Barcelona, Spain, 12–15 February 2020.
79. Ghorbani, A.; Shiry, S.; Nodehi, A. Using genetic algorithm for a mobile robot path planning. In Proceedings of the 2009 International Conference on Future Computer and Communication, Kuala Lumpar, Malaysia, 3–5 April 2009.
80. Gerke, M. Genetic path planning for mobile robots. In Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251), San Diego, CA, USA, 2–4 June 1999; Volume 4.
81. Nonoyama, K.; Liu, Z.; Fujiwara, T.; Alam, M.; Nishi, T. Energy-efficient robot configuration and motion planning using genetic algorithm and particle swarm optimization. *Energies* **2022**, *15*, 2074. [CrossRef]
82. Pamosoaji, A.K.; Piao, M.; Hong, K.-S. PSO-based minimum-time motion planning for multiple vehicles under acceleration and velocity limitations. *Int. J. Control Autom. Syst.* **2019**, *17*, 2610–2623. [CrossRef]
83. Ordonez, C.; Gupta, N.; Reese, B.; Seegmiller, N.; Kelly, A.; Collins, E.G., Jr. Learning of skid-steered kinematic and dynamic models for motion planning. *Robot. Auton. Syst.* **2017**, *95*, 207–221. [CrossRef]
84. Li, Y.; Cui, R.; Li, Z.; Xu, D. Neural network approximation based near-optimal motion planning with kinodynamic constraints using RRT. *IEEE Trans. Ind. Electron.* **2018**, *65*, 8718–8729. [CrossRef]
85. Lindemann, L.; Dimarogonas, D.V. Robust motion planning employing signal temporal logic. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; pp. 2950–2955.
86. Paxton, C.; Raman, V.; Hager, G.D.; Kobilarov, M. Combining neural networks and tree search for task and motion planning in challenging environments. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 6059–6066.

87. Abbeel, P.; Ng, A.Y. Apprenticeship learning via inverse reinforcement learning. In Proceedings of the Twenty-First International Conference on Machine Learning, Banff, AB, Canada, 4–8 July 2004; p. 1.
88. Wulfmeier, M.; Wang, D.Z.; Posner, I. Watch this: Scalable cost-function learning for path planning in urban environments. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, South Korea, 9–14 October 2016; pp. 2089–2095.
89. Pfeiffer, M.; Schaeuble, M.; Nieto, J.; Siegwart, R.; Cadena, C. From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 1527–1533.
90. Mainprice, J.; Hayne, R.; Berenson, D. Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative replanning. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26-30 May 2015; pp. 885–892.
91. Park, J.S.; Park, C.; Manocha, D. Intention-Aware Motion Planning Using Learning Based Human Motion Prediction. In Proceedings of the Robotics: Science and Systems, Cambridge, MA, USA, 12–16 July 2017.
92. Hernández, J.D.; Moll, M.; Kavraki, L.E. Lazy Evaluation of Goal Specifications Guided by Motion Planning. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019.
93. Wilde, N.; Kulić, D.; Smith, S.L. Learning user preferences in robot motion planning through interaction. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 619–626.
94. Qureshi, A.H.; Simeonov, A.; Bency, M.J.; Yip, M.C. Motion planning networks. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 2118–2124.
95. Lenz, D. Motion Planning for Highly-Automated Vehicles under Uncertainties and Interactions with Human Drivers. Ph.D. Thesis, Technische Universität München, München, Germany, 2018.
96. Tian, Y.; Chang, Q. Motion Planning for Human-Robot Collaboration based on Reinforcement Learning. In Proceedings of the 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE), Mexico City, Mexico, 20–24 August 2022.
97. Qureshi, A.H.; Yip, M.C. Deeply informed neural sampling for robot motion planning. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018.
98. Qureshi, A.H.; Miao, Y.; Simeonov, A.; Yip, M.C. Motion planning networks: Bridging the gap between learning-based and classical motion planners. *IEEE Trans. Robot.* **2020**, *37*, 48–66. [CrossRef]
99. Bency, M.J.; Qureshi, A.H.; Yip, M.C. Neural path planning: Fixed time, near-optimal path generation via oracle imitation. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019.
100. Lembono, T.S.; Pignat, E.; Jankowski, J.; Calinon, S. Learning constrained distributions of robot configurations with generative adversarial network. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4233–4240. [CrossRef]
101. Qureshi, A.H.; Dong, J.; Baig, A.; Yip, M.C. Constrained motion planning networks x. *IEEE Trans. Robot.* **2021**, *38*, 868–886. [CrossRef]
102. Everett, M.; Chen, Y.F.; How, J.P. Motion planning among dynamic, decision-making agents with deep reinforcement learning. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 3052–3059.
103. Chen, Y.F.; Everett, M.; Liu, M.; How, J.P. Socially aware motion planning with deep reinforcement learning. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 1343–1350.
104. Sangiovanni, B.; Incremona, G.P.; Piastra, M.; Ferrara, A. Self-configuring robot path planning with obstacle avoidance via deep reinforcement learning. *IEEE Control Syst. Lett.* **2020**, *5*, 397–402. [CrossRef]
105. Zhou, D.; Jia, R.; Yao, H.; Xie, M. Robotic arm motion planning based on residual reinforcement learning. In Proceedings of the 2021 13th International Conference on Computer and Automation Engineering (ICCAE), Melbourne, Australia, 20–22 March 2021.
106. Prianto, E.; Park, J.-H.; Bae, J.-H.; Kim, J.-S. Deep reinforcement learning-based path planning for multi-arm manipulators with periodically moving obstacles. *Appl. Sci.* **2021**, *11*, 2587. [CrossRef]
107. Guo, M.; Wang, Y.; Liang, B.; Chen, Z.; Lin, J.; Huang, K. Robot Path Planning via Deep Reinforcement Learning with Improved Reward Function. In *Proceedings of 2021 Chinese Intelligent Systems Conference: Volume III*; Springer: Singapore, 2022.
108. Gupta, K.; Najjaran, H. Exploiting Abstract Symmetries in Reinforcement Learning for Complex Environments. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022.
109. Incremona, G.P.; Sacchi, N.; Sangiovanni, B.; Ferrara, A. Experimental Assessment of Deep Reinforcement Learning for Robot Obstacle Avoidance: A LPV Control Perspective. *IFAC-PapersOnLine* **2021**, *54*, 89–94. [CrossRef]
110. Wang, S.; Cao, Y.; Zheng, X.; Zhang, T. An end-to-end trajectory planning strategy for free-floating space robots. In Proceedings of the 2021 40th Chinese Control Conference (CCC), Shanghai, China, 26–28 July 2021.
111. Yang, J.; Peng, G. DDPG with Meta-Learning-Based Experience Replay Separation for Robot Trajectory Planning. In Proceedings of the 2021 7th International Conference on Control, Automation and Robotics (ICCAR), Singapore, 23–26 April 2021.
112. Jurgenson, T.; Tamar, A. Harnessing reinforcement learning for neural motion planning. *arXiv* **2019**, arXiv:1906.00214.
113. Chen, P.; Pei, J.; Lu, W.; Li, M. A deep reinforcement learning based method for real-time path planning and dynamic obstacle avoidance. *Neurocomputing* **2022**, *497*, 64–75. [CrossRef]

114. Hockman, B.; Pavone, M. Stochastic motion planning for hopping rovers on small solar system bodies. In Proceedings of the International Symposium on Robotics Research (ISRR), Puerto Varas, Chile, 11–14 December 2017.
115. Phiquepal, C.; Toussaint, M. Combined task and motion planning under partial observability: An optimization-based approach. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 9000–9006.
116. Dantam, N.T.; Chaudhuri, S.; Kavraki, L.E. The Task-Motion Kit: An Open Source, General-Purpose Task and Motion-Planning Framework. *IEEE Robot. Autom. Mag.* **2018**, *25*, 61–70. [CrossRef]
117. Farber, M. Configuration spaces and robot motion planning algorithms. *arXiv* **2017**, arXiv:1701.02083.
118. Elhafsi, A.; Ivanovic, B.; Janson, L.; Pavone, M. Map-Predictive Motion Planning in Unknown Environments. *arXiv* **2019**, arXiv:1910.08184.
119. Dong, J.; Mukadam, M.; Dellaert, F.; Boots, B. Motion Planning as Probabilistic Inference using Gaussian Processes and Factor Graphs. In Proceedings of the Robotics: Science and Systems, Ann Arbor, MI, USA, 18–22 June 2016; Volume 12, p. 4.
120. Marinho, Z.; Dragan, A.; Byravan, A.; Boots, B.; Srinivasa, S.; Gordon, G. Functional gradient motion planning in reproducing kernel hilbert spaces. *arXiv* **2016**, arXiv:1601.03648.
121. Mercy, T.; Van Loock, W.; Pipeleers, G. Real-time motion planning in the presence of moving obstacles. In Proceedings of the 2016 European Control Conference (ECC), Aalborg, Denmark, 29 June–1 July 2016; pp. 1586–1591.
122. Mukadam, M.; Yan, X.; Boots, B. Gaussian process motion planning. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 9–15.
123. Bitar, G.; Martinsen, A.B.; Lekkas, A.M.; Breivik, M. Two-Stage Optimized Trajectory Planning for ASVs Under Polygonal Obstacle Constraints: Theory and Experiments. *IEEE Access* **2020**, *8*, 199953–199969. [CrossRef]
124. Fan, W.; Lee, C.H.; Chen, J.H. A realtime curvature-smooth interpolation scheme and motion planning for CNC machining of short line segments. *Int. J. Mach. Tools Manuf.* **2015**, *96*, 27–46. [CrossRef]
125. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
126. Guo, D.; Xu, F.; Yan, L.; Nie, Z.; Shao, H. A new noise-tolerant obstacle avoidance scheme for motion planning of redundant robot manipulators. *Front. Neurorobot.* **2018**, *12*, 51. [CrossRef] [PubMed]
127. Majumdar, A.; Tedrake, R. Funnel libraries for real-time robust feedback motion planning. *Int. J. Robot. Res.* **2017**, *36*, 947–982. [CrossRef]
128. Cohen, D.C.; Vandembroucq, L. Motion planning in connected sums of real projective spaces. *arXiv* **2018**, arXiv:1807.09947.
129. Hubicki, C.M.; Aguilar, J.J.; Goldman, D.I.; Ames, A.D. Tractable terrain-aware motion planning on granular media: An impulsive jumping study. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, South Korea, 9–14 October 2016; pp. 3887–3892.
130. Mastalli, C.; Havoutis, I.; Focchi, M.; Caldwell, D.; Semini, C. Motion planning for challenging locomotion: A study of decoupled and coupled approaches. 2017. Available online: https://cmastalli.github.io/publications/locomotion20tro.pdf (accessed on 3 April 2023).
131. Ghosh, S.; Otsu, K.; Ono, M. Probabilistic Kinematic State Estimation for Motion Planning of Planetary Rovers. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 5148–5154.
132. Aceituno-Cabezas, B.; Mastalli, C.; Dai, H.; Focchi, M.; Radulescu, A.; Caldwell, D.G.; Cappelletto, J.; Grieco, J.C.; Fernández-López, G.; Semini, C. Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization. *IEEE Robot. Autom. Lett.* **2017**, *3*, 2531–2538.
133. Sieverling, A.; Eppner, C.; Wolff, F.; Brock, O. Interleaving motion in contact and in free space for planning under uncertainty. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 4011–4073.
134. Akbari, A.; Diab, M.; Rosell, J. Contingent task and motion planning under uncertainty for human–robot interactions. *Appl. Sci.* **2020**, *10*, 1665. [CrossRef]
135. Xu, X.; Yang, Y.; Pan, S. Motion Planning for Mobile Robots. In *Advanced Path Planning for Mobile Entities*; IntechOpen: Rijeka, Croatia, 2018; p. 145.
136. Wang, H.; Huang, Y.; Khajepour, A.; Zhang, Y.; Rasekhipour, Y.; Cao, D. Crash mitigation in motion planning for autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 3313–3323. [CrossRef]
137. Simon, B.; Franke, F.; Riegl, P.; Gaull, A. Motion Planning for Collision Mitigation via FEM–Based Crash Severity Maps. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 2187–2194.
138. Zhao, Y.; Wang, Y.; Zhou, M.; Wu, J. Energy-Optimal Collision-Free Motion Planning for Multiaxis Motion Systems: An Alternating Quadratic Programming Approach. *IEEE Trans. Autom. Sci. Eng.* **2018**, *16*, 327–338. [CrossRef]
139. Herbert, S.L.; Chen, M.; Han, S.; Bansal, S.; Fisac, J.F.; Tomlin, C.J. FaSTrack: A modular framework for fast and guaranteed safe motion planning. In Proceedings of the 2017 IEEE 56th Annual Conference on Decision and Control (CDC), Melbourne, VIC, Australia, 12–15 December 2017; pp. 1517–1522.
140. Dantam, N.T.; Kingston, Z.K.; Chaudhuri, S.; Kavraki, L.E. Incremental Task and Motion Planning: A Constraint-Based Approach. In Proceedings of the Robotics: Science and Systems, Ann Arbor, MI, USA, 18–22 June 2016; Volume 12, p. 00052.

141. Becker, A.T.; Fekete, S.P.; Keldenich, P.; Konitzny, M.; Lin, L.; Scheffer, C. Coordinated Motion Planning: The Video (Multimedia Exposition). In Proceedings of the 34th International Symposium on Computational Geometry (SoCG 2018), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Budapest, Hungary, 11–14 June 2018.
142. Sayre-McCord, T.; Karaman, S. Perception-driven sparse graphs for optimal motion planning. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 8110–8117.
143. Vasquez, D. Novel planning-based algorithms for human motion prediction. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 3317–3322.
144. Shoukry, Y.; Nuzzo, P.; Saha, I.; Sangiovanni-Vincentelli, A.L.; Seshia, S.A.; Pappas, G.J.; Tabuada, P. Scalable motion planning using lazy smt-based solving. In Proceedings of the 55th IEEE Conference on Decision and Control (IEEE CDC), Las Vegas, NV, USA, 12–14 December 2016; pp. 6683–6688.
145. Ichnowski, J.; Alterovitz, R. Motion Planning Templates: A Motion Planning Framework for Robots with Low-power CPUs. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019.
146. Kim, B.; Kaelbling, L.P.; Lozano-Pérez, T. Learning to guide task and motion planning using score-space representation. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 2810–2817.
147. Ferrer-Mestres, J.; Frances, G.; Geffner, H. Combined task and motion planning as classical AI planning. *arXiv* **2017**, arXiv:1706.06927.
148. Lin, C.J.; Chen, C.S.; Yu, S. A GPU-based evolution algorithm for motion planning of a redundant robot. *Int. Robot. Autom. J.* **2017**, *2*, 1–14. [CrossRef]
149. Chretien, B.; Escande, A.; Kheddar, A. GPU robot motion planning using semi-infinite nonlinear programming. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *27*, 2926–2939. [CrossRef]
150. Lee, T.; Kim, Y.J. Massively parallel motion planning algorithms under uncertainty using POMDP. *Int. J. Robot. Res.* **2016**, *35*, 928–942. [CrossRef]
151. Fragoso, A.T.; Matthies, L.H.; Murray, R.M. A fast motion planning representation for configuration flat robots with applications to micro air vehicles. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; pp. 4195–4202.
152. Kitagawa, R.; Liu, Y.; Kanda, T. Human-inspired Motion Planning for Omni-Directional Social Robots. In Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction, New York, NY, USA, 8–11 March 2021; pp. 34–42.
153. Jung, J.; Kanda, T.; Kim, M.S. Guidelines for contextual motion design of a humanoid robot. *Int. J. Soc. Robot.* **2013**, *5*, 153–169. [CrossRef]
154. Saiki, L.Y.M.; Kanda, T. Social Group Motion in Robots. In *Social Robotics: 9th International Conference, ICSR 2017, Tsukuba, 22–24 Japan, November 2017, Proceedings*; Springer: Cham, Switzerland, 2017; Volume 10652, p. 474.
155. Zanlungo, F.; Chigodo, Y.; Ikeda, T.; Kanda, T. Experimental study and modelling of pedestrian space occupation and motion pattern in a real world environment. In *Pedestrian and Evacuation Dynamics 2012*; Springer: Cham, Switzerland, 2014; pp. 289–304.
156. Pfeiffer, M.; Schwesinger, U.; Sommer, H.; Galceran, E.; Siegwart, R. Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, South Korea, 9–14 October 2016; pp. 2096–2101.
157. Li, X.A.; Florendo, M.; Miller, E.L.; Ishiguro, H.; Saygin, P.A. Robot form and motion influences social attention. In Proceedings of the 2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI), New York, NY, USA, 2–5 March 2015; pp. 43–50.
158. Yagi, S.; Ise, N.; Yu, S.; Nakata, Y.; Nakamura, Y.; Ishiguro, H. Perception of emotional gait-like motion of mobile humanoid robot using vertical oscillation. In Proceedings of the Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction, New York, NY, USA, 23–26 March 2020; pp. 529–531.
159. Minato, T.; Ishiguro, H. Construction and evaluation of a model of natural human motion based on motion diversity. In Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction, New York, NY, USA, 12–15 March 2008; pp. 65–72.
160. Masayuki, K.; Takahiro, M.; Noriaki, M.; Hiroshi, I.; Norihiro, H. How does a Balancing Motion of a Humanoid Robots Affect a Human Motion and Impression? In Proceedings of the 2006 6th IEEE-RAS International Conference on Humanoid Robots, Genova, Italy, 4–6 December 2006; pp. 252–257.
161. Thornton, S.M.; Lewis, F.E.; Zhang, V.; Kochenderfer, M.J.; Gerdes, J.C. Value sensitive design for autonomous vehicle motion planning. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1157–1162.
162. Althoff, M.; Koschi, M.; Manzinger, S. CommonRoad: Composable benchmarks for motion planning on roads. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 719–726.
163. Li, B.; Shao, Z. A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles. *Knowl.-Based Syst.* **2015**, *86*, 11–20. [CrossRef]
164. Wang, B.; Gong, J.; Chen, H. Motion Primitives Representation, Extraction and Connection for Automated Vehicle Motion Planning Applications. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3931–3945. [CrossRef]
165. Yoshida, E.; Kanehiro, F.; Laumond, J.P. Whole-body Motion Planning. In *Humanoid Robotics: A Reference*; Springer: Dordrecht, The Netherlands, 2017.

166. Mordatch, I.; Lowrey, K.; Todorov, E. Ensemble-cio: Full-body dynamic motion planning that transfers to physical humanoids. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 5307–5314.
167. Motahar, M.S.; Veer, S.; Poulakakis, I. Composing limit cycles for motion planning of 3D bipedal walkers. In Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 12–14 December 2016; pp. 6368–6374.
168. Zielinska, T.; Zimin, L.; Szumowski, M.; Ge, W. Motion Planning for a Humanoid Robot with Task Dependent Constraints. In Proceedings of the IFToMM World Congress on Mechanism and Machine Science, Krakow, Poland, 15–18 July 2019; pp. 1681–1690.
169. Yang, Y.; Merkt, W.; Ferrolho, H.; Ivan, V.; Vijayakumar, S. Efficient humanoid motion planning on uneven terrain using paired forward-inverse dynamic reachability maps. *IEEE Robot. Autom. Lett.* **2017**, *2*, 2279–2286. [CrossRef]
170. Tajika, T.; Miyashita, T.; Ishiguro, H.; Hagita, N. Reducing influence of robot's motion on tactile sensor based on partially linear model. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 512–517.
171. Rosenstrauch, M.J.; Krüger, J. Safe human-robot-collaboration-introduction and experiment using ISO/TS 15066. In Proceedings of the 2017 3rd International Conference on Control, Automation and Robotics (ICCAR), Nagoya, Japan, 24–26 April 2017; pp. 740–744.
172. Vysockỳ, A.; Wada, H.; Kinugawa, J.; Kosuge, K. Motion Planning Analysis According to ISO/TS 15066 in Human–Robot Collaboration Environment. In Proceedings of the 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Hong Kong, China, 8–12 July 2019; pp. 151–156.
173. Lin, X.; Zhang, J.; Shen, J.; Fernandez, G.; Hong, D.W. Optimization Based Motion Planning for Multi-Limbed Vertical Climbing Robots. *arXiv* **2019**, arXiv:1909.06339.
174. Mirrazavi Salehian, S.S.; Figueroa, N.; Billard, A. A unified framework for coordinated multi-arm motion planning. *Int. J. Robot. Res.* **2018**, *37*, 1205–1232. [CrossRef]
175. Garrett, C.R.; Lozano-Perez, T.; Kaelbling, L.P. FFRob: Leveraging symbolic planning for efficient task and motion planning. *Int. J. Robot. Res.* **2018**, *37*, 104–136. [CrossRef]
176. Liu, S.; Atanasov, N.; Mohta, K.; Kumar, V. Search-based motion planning for quadrotors using linear quadratic minimum time control. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 2872–2879.
177. Singhal, A.; Aharony, N.; Liang, A.T.; Muralidhar, G. Adaptive Route and Motion Planning Based on Learned External and Internal Vehicle Environment. U.S. Patent 10,234,302, 19 March 2019.
178. Dragan, A.D.; Bauman, S.; Forlizzi, J.; Srinivasa, S.S. Effects of robot motion on human-robot collaboration. In Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction, New York, NY, USA, 2–5 March 2015; pp. 51–58.
179. Sina Mirrazavi Salehian, S.; Figueroa, N.; Billard, A. Coordinated multi-arm motion planning: Reaching for moving objects in the face of uncertainty. In Proceedings of the Robotics: Science and Systems, Ann Arbor, MI, USA, 18–22 June 2016.
180. Alonso-Mora, J.; DeCastro, J.A.; Raman, V.; Rus, D.; Kress-Gazit, H. Reactive mission and motion planning with deadlock resolution avoiding dynamic obstacles. *Auton. Robot.* **2018**, *42*, 801–824. [CrossRef]
181. Shoukry, Y.; Nuzzo, P.; Balkan, A.; Saha, I.; Sangiovanni-Vincentelli, A.L.; Seshia, S.A.; Pappas, G.J.; Tabuada, P. Linear temporal logic motion planning for teams of underactuated robots using satisfiability modulo convex programming. In Proceedings of the 2017 IEEE 56th Annual Conference on Decision and Control (CDC), Melbourne, VIC, Australia, 12–15 December 2017; pp. 1132–1137.
182. Alonso-Mora, J.; Knepper, R.; Siegwart, R.; Rus, D. Local motion planning for collaborative multi-robot manipulation of deformable objects. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 5495–5502.
183. Ma, X.; Jiao, Z.; Wang, Z.; Panagou, D. Decentralized prioritized motion planning for multiple autonomous UAVs in 3D polygonal obstacle environments. In Proceedings of the 2016 International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 7–10 June 2016; pp. 292–300.
184. Li, B.; Liu, H.; Xiao, D.; Yu, G.; Zhang, Y. Centralized and optimal motion planning for large-scale AGV systems: A generic approach. *Adv. Eng. Softw.* **2017**, *106*, 33–46. [CrossRef]
185. Solovey, K.; Halperin, D. On the hardness of unlabeled multi-robot motion planning. *Int. J. Robot. Res.* **2016**, *35*, 1750–1759. [CrossRef]
186. Liu, Y.; Zhao, Y.; Chen, G. Finite-time formation tracking control for multiple vehicles: A motion planning approach. *Int. J. Robust Nonlinear Control* **2016**, *26*, 3130–3149. [CrossRef]
187. Liu, Y.; Geng, Z. Finite-time formation control for linear multi-agent systems: A motion planning approach. *Syst. Control Lett.* **2015**, *85*, 54–60. [CrossRef]
188. Saha, I.; Ramaithitima, R.; Kumar, V.; Pappas, G.J.; Seshia, S.A. Implan: Scalable incremental motion planning for multi-robot systems. In Proceedings of the 2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS), Vienna, Austria, 11–14 April 2016; pp. 1–10.
189. Solovey, K.; Yu, J.; Zamir, O.; Halperin, D. Motion planning for unlabeled discs with optimality guarantees. *arXiv* **2015**, arXiv:1504.05218.

190. Atias, A.; Solovey, K.; Salzman, O.; Halperin, D. Effective metrics for multi-robot motion-planning. *Int. J. Robot. Res.* **2018**, *37*, 1741–1759. [CrossRef]
191. Sydney, N.; Paley, D.A.; Sofge, D. Physics-inspired motion planning for information-theoretic target detection using multiple aerial robots. *Auton. Robot.* **2017**, *41*, 231–241. [CrossRef]
192. Liu, Y.; Zhao, Y.; Chen, G. Sampled-data-based consensus and containment control of multiple harmonic oscillators: A motion-planning approach. *Chaos Interdiscip. J. Nonlinear Sci.* **2016**, *26*, 116303. [CrossRef]
193. Mohanty, P.K.; Singh, A.K.; Kumar, A.; Mahto, M.K.; Kundu, S. Path Planning Techniques for Mobile Robots: A Review. In *Proceedings of the International Conference on Soft Computing and Pattern Recognition, 2021*; Springer International Publishing: Cham, Switzerland, 15–17 December 2021.
194. Lu, Y.; Xue, Z.; Xia, G.-S.; Zhang, L. A survey on vision-based UAV navigation. *Geo-Spat. Inf. Sci.* **2018**, *21*, 21–32. [CrossRef]
195. Wang, Q.; Wieghardt, C.S.; Jiang, Y.; Gong, J.; Wagner, B. Generalized path corridor-based local path planning for nonholonomic mobile robot. In Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Gran Canaria, Spain, 15–18 September 2015.
196. Huy, Q.; Mita, S.; Nejad, H.T.N.; Han, L. Dynamic and safe path planning based on support vector machine among multi moving obstacles for autonomous vehicles. *IEICE Trans. Inf. Syst.* **2013**, *96*, 314–328.
197. Liu, L.; Wang, X.; Yang, X.; Liu, H.; Li, J.; Wang, P. Path planning techniques for mobile robots: Review and prospect. *Expert Syst. Appl.* **2023**, *227*, 120254. [CrossRef]
198. Tang, S.H.; Khaksar, W.; Ismail, N.; Ariffin, M. A review on robot motion planning approaches. *Pertanika J. Sci. Technol.* **2012**, *20*, 15–29.
199. Xiao, X.; Liu, B.; Warnell, G.; Stone, P. Motion planning and control for mobile robot navigation using machine learning: A survey. *Auton. Robot.* **2022**, *46*, 569–597. [CrossRef]
200. Najjaran, H.; Goldenberg, A. Real-time motion planning of an autonomous mobile manipulator using a fuzzy adaptive Kalman filter. *Robot. Auton. Syst.* **2007**, *55*, 96–106. [CrossRef]
201. Aydemir, H.; Tekerek, M.; Gök, M. Complete coverage planning with clustering method for autonomous mobile robots. *Concurr. Comput. Pract. Exp.* **2023**, e7830. [CrossRef]
202. Hwang, J.Y.; Kim, J.S.; Lim, S.S.; Park, K.H. A fast path planning by path graph optimization. *IEEE Trans. Syst. Man Cybern.-Part A Syst. Hum.* **2003**, *33*, 121–128. [CrossRef]
203. Zhu, M.; Xiao, C.; Gu, S.; Du, Z.; Wen, Y. A circle grid-based approach for obstacle avoidance motion planning of unmanned surface vehicles. *Proc. Inst. Mech. Eng. Part J. Eng. Marit. Environ.* **2023**, *237*, 132–152. [CrossRef]
204. Song, R.; Liu, Y.; Bucknall, R. Smoothed A* algorithm for practical unmanned surface vehicle path planning. *Appl. Ocean Res.* **2018**, *83*, 9–20. [CrossRef]
205. Likhachev, M.; Gordon, G.J.; Thrun, S. ARA*: Anytime A* with provable bounds on sub-optimality. *Adv. Neural Inf. Process. Syst.* **2003**, *16*, 767–774.
206. Zhang, Y.; Wang, H.; Yin, M.; Wang, J.; Hua, C. Bi-AM-RRT*: A Fast and Efficient Sampling-Based Motion Planning Algorithm in Dynamic Environments. *arXiv* **2023**, arXiv:2301.11816.
207. Stentz, A. The focussed d* algorithm for real-time replanning. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Montreal, QC, Canada, 20–25 August 1995; Volume 95.
208. Balan, K.; Luo, C. Optimal trajectory planning for multiple waypoint path planning using tabu search. In Proceedings of the 2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 8–10 November 2018.
209. Liu, H.; Wang, L. Collision-free human-robot collaboration based on context awareness. *Robot. Comput.-Integr. Manuf.* **2021**, *67*, 101997. [CrossRef]
210. Scalera, L.; Vidoni, R.; Giusti, A. Optimal scaling of dynamic safety zones for collaborative robotics. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021.
211. Scalera, L.; Giusti, A.; Vidoni, R.; Gasparetto, A. Enhancing fluency and productivity in human-robot collaboration through online scaling of dynamic safety zones. *Int. J. Adv. Manuf. Technol.* **2022**, *121*, 6783–6798. [CrossRef]
212. Taylor, O.; Rodriguez, A. Optimal shape and motion planning for dynamic planar manipulation. *Auton. Robot.* **2019**, *43*, 327–344. [CrossRef]