Robotics and Autonomous Systems 107 (2018) 20–44

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Motion planning survey for autonomous mobile manipulators underwater manipulator case study

Dina Youakim *, Pere Ridao

Computer Vision & Robotics Research Institute, University of Girona, Spain

ARTICLE INFO

Article history:
Received 17 January 2018
Received in revised form 14 April 2018
Accepted 7 May 2018
Available online 1 June 2018

Keywords:
Motion planning
Survey
MoveIt!
Manipulation
Underwater
Autonomous
Obstacle avoidance

ABSTRACT

Sampling-based, search-based, and optimization-based motion planners are just some of the different approaches developed for motion planning problems. Given the wide variety of application tackled by autonomous mobile manipulators, the question "which planner to choose" may be tough. In this paper, we review the state of the art of the most common approaches, and present a set of benchmarks with the aim to provide not only a theoretical review but also a qualitative/quantitative comparison of the algorithms. Our purpose is to provide an insight and analyze their performance with respect to different metrics. The results are based on an Underwater Vehicle Manipulator System *UVMS*, although they can be extended to terrestrial and aerial robots as well. The paper uses these results to formalize a set of guidelines for the selection process of the most appropriate approach, for a given problem/requirements.

1. Introduction

Motion planning is a fundamental topic in robotics [1–3] which deals with the problem of finding a collision-free path going from an initial to a target configuration. Its basic form is *the piano mover's problem* which assumes a robot is a point. It has evolved through time to address a number of variations, depending on the specifications of the system and the environment in which it is expected to operate.

During the last few years several works have compared and analyzed the motion planning algorithms. The basic theory and the most common motion planning approaches have been reviewed in [4]. On the other hand, [5] and [6] each focused on just one common approach, being the sampling-based and heuristic-based respectively. Similarly [7] focused on the theory of classical and heuristic-based motion planning for navigation. In [8], the authors presented an analysis of probabilistic-based algorithms, discussing their performance regarding dynamic obstacles and narrow passages. Some other studies have been concerned with different robotics domains. For instance, the work presented in [9] focused on Unmanned Aerial Vehicles, [10] dealt with autonomous vehicles and [11] presented a more thorough algorithms taxonomy for 3D path planning. A quantitative comparison of motion planning algorithms for an aerial manipulator using *MoveIt!* was presented in [13] through a single test case. Similarly in [14], underwater bench-marking was covered. This paper focused on explaining a framework for simulating the environment using UWSIM [15] through one case study. Other surveys addressed different research problems related to motion planning. In this direction, [16] studied the coverage path-planning problem and [17] dealt with motion planning in dynamic environments, which is becoming a challenging topic of research.

Regarding benchmarking, some work has been done related to the quantitative evaluation of the techniques. For instance, in [18] the authors defined their own benchmarks to carry out a comparative study among a set of optimal motion planners. In [17], the focus was reviewing techniques for motion planning in dynamic environments. A ROS-based [19] framework using *MoveIt!* [12], was proposed in [20] for benchmarking sampling-based motion planning. Their purpose was to set-up guidelines for a terrestrial robots benchmarks database, along with the infrastructure to get comparative results. In contrast, the work in [21] is one of a kind, being focused on dual-arm manipulation; summarizing the state of the art of different approaches, ranging from low-level control to high level task planning and execution. Finally, [22] presented an early but unique work through an experimental evaluation of different collision detection mechanisms as well as their impact on the motion planning problem.

The intended contribution of the present work, with respect to the state-of-the-art surveys, is to advance beyond a theoretical-only review of the literature. This paper provides extensive comparisons based on kinematic simulations of the UVMS system. New statistical measures are proposed to provide a meaningful

* Corresponding author.
E-mail address: dina.isaac@udg.edu (D. Youakim).

https://doi.org/10.1016/j.robot.2018.05.006
0921-8890/

qualitative/quantitative comparison for existing/new planners. In addition, we present new underwater benchmarks, with the purpose of complementing those already available in the literature. Finally, to maintain a standardized and re-usable solution, the results presented are based on the latest ROS/*MoveIt!* framework for mobile manipulation. A total set of seventeen representative algorithms have been tested using five illustrative benchmarks and compared against seven metrics; with the purpose of providing clear and structured guidance for other researchers, on how to choose the most suitable motion planner for their application. To implement a full solution for mobile manipulators, it would be common to use hybrid frameworks (as will be summarized in Section 5) merging motion planning and reactive control techniques. For instance having two levels of obstacle avoidance: with reactive avoidance at the low level, in addition to the motion planning one. In this paper we will focus only on comparing the performance of the motion planning algorithms. It is worth noting that all the surveyed algorithms could be integrated with existing reactive control techniques.

The paper is organized as follows: Sections 2 and 3, present general motion planning concepts and requirements to be taken into account before deciding the approach to be used. Special attention is given to manipulation planning as explained in Section 4. Section 5 presents the theory of different motion planning approaches, with highlights of the pros and cons of each, based on the literature. In Section 7, we detail our own UVMS system as well as the tools and methodology we have followed. The main contribution of the work is in Section 8 where the comparative results and analysis are detailed. Finally, we conclude with guidelines and discussion in Section 9.

2. Common concepts

2.1. System properties

The feasibility of a motion planning approach is highly dependent on the characteristics of both the robot and the environment. Below are the common terminologies used to define a planning problem:

- **Configuration-Space (CS):** alternatively called Joint Space. A robot configuration is a vector of values representing the state of each movable joint, usually expressed as a vector of joint positions (either angular or prismatic): $q = (q_1, q_2, \ldots, q_n)$. The number of Degrees Of Freedom (DoFs) is the number of joints (n). The set of feasible configurations that avoids collision with obstacles is called the free space C_{free}. Its complement in CS is called the obstacle region $C_{obstacle}$. Planning in the Configuration Space may constitute a bottleneck for systems with many DoFs like mobile manipulators or humanoids.
- **Work-Space (WS):** interchangeably named End-Effector space, refers to the environment where the robot is allowed to move. It is either 2-D where the robot moves in a plane or 3-D which is equivalent to motion in the real world. A redundant system is a system for which the same end-effector pose may be reached with more than one valid configuration.
- **State-Space (SS):** a state represents the condition of the robot. It is the set of all feasible states and this is usually infinite. A state can be represented in either the configuration-space or the work-space, it can also be discrete or continuous.
- **Path:** in the *CS*, a path is a continuous curve C connecting two configurations q and q'. A *trajectory* is a path $C(t)$ parameterized by time t.

Any motion planning problem can be represented by an initial start state S_{start} and a goal state S_{goal}. A goal can be represented either in the *CS* where each joint has to attain a specific value, or in the *WS* where the end-effector of the robot has to reach a defined pose (2-D or 3-D). A plan constitutes a sequence of actions (either in the CS or WS depending on the planning space) to be taken to move the robot from the start to the goal, successfully taking into account one or more constraints. Section 3 will discuss the constraints in greater depth.

2.2. Motion planner properties

- **Computation Time:** the computational cost of the planner in terms of running time, i.e. the time spent by the planner to generate a valid plan.
- **Completeness:** An algorithm is complete if it finds a solution whenever one exists. Normally, it is a result of exact algorithms that build an exact representation of the world without losing any information. Usually completeness is traded-off against efficiency, as an accurate representation of the world may decrease efficiency. Two weaker notions exist to relax these conditions: "Resolution-Completeness" which means that the planner is able to find a plan if one exists, and if the resolution of the environment discretization is fine enough to capture relevant information. "Probabilistic-Completeness" planners refer to those that as more as they spend time planning, the probability of finding a solution, if one exists, increases to 1. Usually their performance is measured by their rate of convergence.
- **Optimality:** An algorithm is said to be optimal with respect to a certain criterion if it is able to find a plan that reaches the goal while optimizing such a criterion (e.g. length, execution time, energy consumption). As in the case of completeness, there are two weaker notions of optimality: "Resolution-Optimality" and "Probabilistic-Optimality".
- **Offline vs. Online:** An offline-planner is the one which does offline pre-processing to easily provide an online plan. On the other hand, the online planner incrementally processes the information to compute the plan at the time it is requested. Online planners can adapt to unexpected changes in the system/environment, an essential requirement for autonomous systems. Designing such a planner, and simultaneously keeping time-efficiency remains a challenge.
- **Local vs. Global:** local planners rely on local information in the neighborhood of the current state of the robot, contrarily global ones rely on system sensors to be able to perceive the global state of the environment and plan accordingly.

2.3. Path quality

An important aspect to be considered when evaluating and choosing a planner is the quality of the plans/paths produced. As in [23], the common metrics to evaluate path quality are:

- **Path Length:** typically, it is desirable to produce short paths, while maintaining planning efficiency. Since maintaining both is challenging, some techniques reduce the path length in a post-processing step after the plan generation.
- **Path Clearance:** the aim of any autonomous system is to generate collision-free paths. It is desirable that the generated path keeps at least a minimum distance away from obstacles. Moreover, traveling along high clearance paths reduces the chances of collisions due to various uncertainties (e.g. robot localization). This can also be treated as a post-processing step.