Nonlinear Analysis: Hybrid Systems 30 (2018) 266–284

Contents lists available at ScienceDirect

Nonlinear Analysis: Hybrid Systems

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

Event-driven optimal control for a robotic exploration, pick-up and delivery problem

Vladislav Nenchev [a,*], Christos G. Cassandras [b], Jörg Raisch [a,c]

[a] Control Systems Group, TU Berlin, Einsteinufer 17, 10587 Berlin, Germany
[b] Division of Systems Engineering, 15 St. Mary's St., Boston University, Brookline, MA 02446, USA
[c] Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany

ARTICLE INFO

Article history:
Received 7 May 2017
Accepted 26 June 2018
Available online xxxx

Keywords:
Optimal control
Hybrid systems
Motion control

ABSTRACT

This paper addresses an Optimal Control Problem (OCP) for a robot that has to find and collect a finite number of objects and move them to a depot in minimum time. The robot has fourth-order dynamics that change instantaneously at any pick-up or drop-off of an object. The objects are represented by point masses in a bounded two-dimensional space that may contain unknown obstacles. The OCP is formulated assuming either a worst-case positioning, or a uniform distribution of the objects (probabilistic case). Modeling the robotic problem by a hybrid system facilitates an event-driven receding horizon solution based on motion parameterization and gradient-based optimization. A comparison of the proposed methods to two simple heuristic approaches in simulation suggests that the event-driven approach offers significant advantages – a lower execution time (on average) and the ability to handle obstacles – over the simple solutions, at the price of a moderately increased computational effort. The methods are relevant for various robotic applications, e.g. the motion control of mobile manipulators for home-care, search and rescue, harvesting, manufacturing etc.

1. Introduction

One of the major challenges in autonomous robotic navigation is coping with uncertainties arising from limited a priori knowledge of the environment. Acquiring necessary information and achieving the overall goal are complementary subtasks that require adapting the motion of a robot during mission execution, typically accompanied by minimizing a performance criterion. In this work we address an Optimal Control Problem (OCP) for a robot with fourth-order dynamics that has to find, collect and move a finite number of objects to a designated spot in minimum time. The objects with a priori known masses are located in a bounded two-dimensional space, where the robot is capable of localizing itself using a state-of-the-art simultaneous localization and mapping (SLAM) system [1]. The challenging aspects of the problem at hand are (at least) threefold. One of them arises due to the discontinuity of the value function denoting the overall completion time, which makes it hard to obtain an explicit controller even for deterministic linear systems [2,3]. Fortunately, a wide range of approximate solutions has been proposed, including approaches based on numerical continuation [4], value set approximation [5], multi-parametric programming [6] etc. Another challenge follows from the requirement to collect a finite number of objects and drop them at the depot with minimal overall cost, which represents an instance of the well-known NP-hard Traveling Salesperson Problem (TSP) [7]. While similar Vehicle Routing Problems (VRPs) have been extensively

* Corresponding author.
E-mail addresses: nenchev@control.tu-berlin.de (V. Nenchev), cgc@bu.edu (C.G. Cassandras), raisch@control.tu-berlin.de (J. Raisch).

https://doi.org/10.1016/j.nahs.2018.06.004
1751-570X/

addressed in Operations Research [8], a distinguishing feature of the addressed problem is the continuous dynamics of the Salesperson (i.e. the robot) that change upon object pick-ups and drop-offs. While deterministic autonomously switching dynamics can be handled efficiently, e.g., by two-stage optimization [9,10] or relaxation [11], the complexity of most approaches for setups that involve uncertainties scales poorly with the problem size [12].

Further, optimal exploration of a limited space is an inherently difficult problem by itself. Minimizing the expected time for detecting a target located on a real line with a known probability distribution by a searcher that can change its motion direction instantaneously, has a bounded maximal velocity and starts at the origin, was originally addressed in [13] and extended in [14]. Different versions of this problem have received considerable attention from several research communities, e.g., as a "pursuit-evasion game" in game theory [15,16], as a "cow-path problem" in computer science [17] or as a "coverage problem" in control [18,19], but its solution for a general probability distribution or a general geometry of the region is, to a large extent, still an open question. Effective approaches for the related persistent monitoring problem based on estimation [20], linear programming [21] or parametric optimization [22] have been also been proposed. OCPs with uncertainties have also been addressed by certainty equivalent event-triggered [23], minimax [24] and sampling-based [25] optimization schemes. While methods for Partially Observable Markov Decision Processes (POMDP's) can also be applied, e.g., [26,27], they typically become computationally infeasible for larger problem instances. Due to the aforementioned aspects, the problem at hand has exponential complexity in the number of objects and for any chosen time and space discretization. In this context, employing a discrete abstraction of the underlying continuous dynamics is often only possible by introducing a hierarchical decomposition [28], or additional assumptions that simplify the implementation of automatically synthesized hybrid controllers [29]. Alternatively, one may resort to event-based receding horizon approaches that have been shown to outperform other optimization methods under the presence of uncertainty, e.g., for the elevator dispatching problem [30], multi-agent reward collection problems [31] or planning with temporal logic constraints [32].

Since the locations of the objects are the only source of uncertainty in the considered problem, the ultimate goal is a tractable and scalable, albeit suboptimal, solution that avoids time discretization and requires re-computation only upon a detection of an object. Analytical solutions have been derived for the OCP of a robot, which has to find, collect and move a finite number of objects back to a depot in minimum time, but is allowed to move only along a line [33]. Since a direct generalization of this result for higher dimensional position spaces was not possible, an event-driven receding horizon approach was proposed in [34], where the robot moves along a fixed exploratory trajectory. In this paper, we address the OCP by an event-driven receding horizon approach that allows for adjusting the shape of the exploratory trajectory online, which is particularly useful under the presence of a priori unknown obstacles. Introducing a finite parameterization of the motion of the robot enables the use of Infinitesimal Perturbation Analysis (IPA) [35] for solving the worst and probabilistic case OCPs by an iterative optimization scheme only upon detecting previously undiscovered objects (or obstacles). Two additional heuristic event-driven schemes are introduced for comparison. The first one is based on exploring the environment until all objects are discovered, followed by an optimal pick-up and drop-off of all objects. The second one is based on enforcing a pick-up of an object upon its detection, followed by an optimal exploration and drop-off of all currently carried objects until all remaining ones are discovered, thus resembling the policy introduced in [36].

The remainder of the paper is organized as follows: in Section 2, we present the problem formulation. Section 3 starts with a brief discussion on the performance index and introduces a lower bound for the cost-to-go, followed by the proposed event-driven (Section 4) and the heuristic approaches (Section 5). The methods are then compared in a numerical example (Section 6), followed by the conclusions in Section 7.

Notation. *For a set S, $|S|$ and 2^S denote its cardinality and the set of all of its subsets (power set), respectively. For $r \in \mathbb{R}$, respectively, $r \in \mathbb{R}^n$, $|r|$ and $\|r\|$ denote the absolute value and the Euclidean norm. $\mathbf{I}_n$ is an identity matrix with dimension n. $\mathbf{0}_{m,n}$ represents an $m \times n$ matrix with zero entries. For a vector of zeros or ones with length m, we write $\mathbf{0}_m$ or $\mathbf{1}_m$, respectively. $\mathbb{R}, \mathbb{R}_{\geq 0}, \mathbb{R}_{>0}$ denote the sets of reals, non-negative reals and positive reals, respectively. We use the derivatives $\dot{x}(t) = \frac{dx(t)}{dt}$, $c'(s,\theta) = \frac{\partial c(s,\theta)}{\partial s}$ and the gradient $\nabla_\theta c(s,\theta) = \left[\frac{\partial c(s,\theta)}{\partial \theta_1}, \ldots, \frac{\partial c(s,\theta)}{\partial \theta_n}\right]^T$.*

2. Problem formulation

Consider a mobile robot that has to find, collect and move a finite set of objects $O = \{o_1, \ldots, o_L\}$ located in a limited position space $\mathcal{Y}_g = [-y_{\max}, y_{\max}] \times [-y_{\max}, y_{\max}] \subset \mathbb{R}^2$ back to a designated known spot (depot) y_d in minimum time. Every object $o_l, l \in \{1, \ldots, L\}$ is uniquely characterized by its position $p^{(l)} \in \mathcal{Y}_g$, which is a priori unknown to the robot, and its mass $m^{(l)} \in \mathbb{R}_{\geq 0}$. The number of objects and their masses are assumed to be known. For simplicity, the absence of obstacles in $\mathcal{Y}_g$ and a single depot located at $y_d = \mathbf{0}_2$ are assumed in the main part. Handling obstacles is revisited in the end of Section 4.

The overall system is modeled by a hybrid automaton with continuous inputs [37], i.e., a 9-tuple $\mathcal{H} = \{Q, X, F, U, E, \text{Inv}, G, R, \text{Init}\}$, where:

- Q is the finite set of discrete states;
- $X \subseteq \mathbb{R}^n$ is the continuous state set;
- $U \subseteq \mathbb{R}^m$ is the continuous input set;