Optimal exploration and control for a robotic pick-up and delivery problem in two dimensions

Vladislav Nenchev and Christos G. Cassandras

Abstract— **This paper addresses an optimal control problem for a robot that has to find and collect a finite number of objects and move them to a depot in minimum time. The objects are modeled by point masses with a priori unknown locations in a bounded two-dimensional space. The robot has forth-order dynamics that change instantaneously at any pick-up or drop-off of an object. The corresponding hybrid Optimal Control Problem (OCP) is solved by a receding horizon scheme, where the derived lower bound for the cost-to-go is evaluated for the worst- and a probabilistic case, assuming a uniform distribution of the objects. We first present a time-driven approximate solution based on time and position space discretization. Due to the high computational cost of this solution, we alternatively propose an event-driven approximate approach based on a suitable motion parameterization. The solutions are compared in a numerical example, suggesting that the latter approach offers a significant computational advantage while yielding similar qualitative results compared to the former.**

I. Introduction

We address a time Optimal Control Problem (OCP) for a vehicle with continuous dynamics that has to find, collect and move a finite number of objects back to a designated spot in space. The objects with a-priori known masses are located in a bounded two-dimensional space. The challenging aspects of this problem are (at least) threefold. First, obtaining an explicit time-optimal controller is often impossible even for linear systems without uncertainties due to the discontinuity of the value function [1]. A wide range of approximate solutions has been proposed, e.g. based on numerical continuation [2], value set approximation [3] etc. Second, collecting a finite number of objects and moving them to a particular spot with minimal overall cost represents an instance of the well-known NP-hard Traveling Salesperson Problem (TSP) [4]. Further, object pick-ups and drop-offs cause autonomous switchings of the robot's continuous dynamics, leading to an additional growth in the number of optimal solution candidates. While deterministic versions of this problem can be handled efficiently, e.g., by two-stage optimization [5] or relaxation [6], the complexity of most approaches for stochastic setups [7] scales poorly with the problem size. Third, optimal exploration of a limited space is an inherently difficult problem by itself. For a known probability distribution, minimizing the expected time for detecting a target located on a real line by a searcher that can change its motion direction instantaneously, has a bounded maximal velocity and starts at the origin, was originally addressed in [8]. Even though this problem has received considerable attention from different research communities, e.g., as a "pursuit-evasion game" in game theory [9], [10], as a "cow-path problem" in computer science [11] or as a "coverage problem" in robotics [12], its solution for a general probability distribution or a general geometry of the region is, to a large extent, still an open question. Certainty equivalent event-triggered [13], min-max [14] and sampling-based [15] optimization schemes have also been employed for OCPs with uncertainties. While methods for Partially Observable Markov Decision Processes can also be applied, e.g., [16], they may 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 time and space discretization. This motivates the use of a receding horizon approach, which has been shown to outperform other alternatives under uncertainty, e.g., for elevator dispatching [17] or multi-agent reward collection [18].

In [19], a combined optimal exploration and control scheme was proposed for a vehicle that has to find, collect and move a finite but unknown number of objects in a two-dimensional position space. The approach was based on a policy enforcing a pick-up upon an object's detection, followed by a certainty equivalent discrete optimization on a finite approximation of the motion in the environment. This heuristic restriction was omitted in [20], where the OCP was studied for the worst- and a probabilistic case assuming uniform distribution of the objects on a line interval. It was shown that one optimal solution consists of complete exploration followed by a pick-up and drop-off of all objects. Since this result could not be generalized for higher dimensions, in this paper we propose a receding horizon approach based on solving a non-convex OCP for a finite space and time discretization. Leveraging ideas from [21], where the persistent monitoring of a limited one- dimensional mission space by a team of agents has been effectively reduced to a parametric OCP, we alternatively parameterize the motion by finite curves that allow to cover the whole position space using the robot's sensing range. The non-convex worst- and probabilistic case OCPs are then replaced by a finite number of convex OCPs and integral evaluations along the curve.

Notation. *For a set S, $|S|$ and 2^S denote its cardinality and the set of all of its subsets (power set), respectively. $\mathbf{0}_{m,n}$ represents an $m \times n$ matrix with zero entries. If $m = 1$, we write $\mathbf{0}_n$. $\mathbf{I}_n$ is an identity matrix with dimension n. The derivative of $x(t)$ with respect to time is denoted by $\dot{x}(t)$.*

V. Nenchev is with the Control Systems Group, Technische Universität Berlin, Germany. nenchev@control.tu-berlin.de

C. G. Cassandras is with the Division of Systems Engineering and Center for Information and Systems Engineering, Boston University, MA, USA. cgc@bu.edu

C. G. Cassandras was supported in part by NSF under grants CNS-1239021 and IIP-1430145, by AFOSR under grant FA9550-12-1-0113, by ONR under grant N00014-09-1-1051, by ARO under grant W911NF-11-1-0227 and by the Cyprus Research Promotion Foundation under Grant New Infrastructure Project/Strategic/0308/26.

The authors would like to thank J. Raisch for insightful discussions.

II. Problem formulation

Consider a finite set of objects $O = \{o_1, \ldots, o_L\}$, where every $o_l, l \in \{1, \ldots, L\}$, is uniquely characterized by its position $p^{(l)} \in \mathcal{Y}_g$, $\mathcal{Y}_g = [-y_{\max}, y_{\max}] \times [-y_{\max}, y_{\max}] \subset \mathbb{R}^2$, and mass $m^{(l)} \in \mathbb{R}_{\geq 0}$. A robot with a sensor footprint around its current position $y(t) \in \mathbb{R}^2$, hence covering the area

$$\mathcal{O}(y(t)) = \{y_p \in \mathbb{R}^2 : \|y(t) - y_p\| \leq \rho\}, \rho \ll y_{\max}, \quad (1)$$

has to find, collect and move all objects back to the depot located at $y_d = \mathbf{0}_2$ in minimum time. The overall system is modeled by a hybrid automaton H. The discrete state at time t is $q(t) = (q_1(t), q_2(t), q_3(t))$, where $q_1(t) \subseteq O$ is the set of objects being carried by the robot, $q_2(t) \subseteq O$ the set of objects that has been dropped at the depot prior to or at time t, and $q_3(t) \subseteq O$ is the set of objects that have been detected so far. Clearly, $q(t) \in Q$ with $Q := 2^O \times 2^O \times 2^O$. The mass of the robot and its current load is $m_q(t) = m_{q_1}(t) = m_\emptyset + \sum_{o_l \in q_1} m^{(l)}$, where $m_\emptyset$ stands for the nominal mass of the robot. The continuous state $(x(t), \mathcal{Y}(t)) \in \mathcal{X}$ consists of the robot state $x(t) = [y^T(t)\ v^T(t)]^T \in X$, where $v(t) \in \mathbb{R}^2$ is the current velocity of the robot, and the region $\mathcal{Y}(t) \subseteq \mathcal{Y}_g$ that has not been explored at time t. Clearly, $\mathcal{Y}(t)$ is non-increasing, i.e., $\mathcal{Y}(t) \subseteq \mathcal{Y}(\tau)$ for $t \geq \tau$. The state $x(t)$ evolves according to a finite collection of vector fields $F = \{f_q\}_{q \in Q}$, i.e.

$$\dot{x}(t) = f_q(x, u) = \begin{bmatrix} \mathbf{0}_{2,2} & \mathbf{I}_2 \\ \mathbf{0}_{2,2} & \mathbf{0}_{2,2} \end{bmatrix} x(t) + \frac{1}{m_{q(t)}} \begin{bmatrix} \mathbf{0}_{2,2} \\ \mathbf{I}_2 \end{bmatrix} u(t), \quad (2)$$

driven by the piecewise continuous control signal $u : [0, t_f] \rightarrow U := \{\phi \in \mathbb{R}^2 | \|\phi\| \leq u_{\max}\}$, where t_f is the free final time of the overall pick-up and drop-off mission. As Q is finite, the set of discrete state transitions (or events) $E \subseteq Q \times Q$ is also finite. We partition E into $\Delta \cup \Pi \cup \Psi$, where for $q = (q_1, q_2, q_3), q' = (q'_1, q'_2, q'_3) \in Q$, $\Delta = \{(q, q') : q'_1 = q_1, q'_2 = q_2, q'_3 = q_3 \cup \{o_l\}, o_l \in O\}$ is the set of detection events, $\Pi = \{(q, q') : q'_1 \setminus q_1 = \{o_l\}, q'_2 = q_2, q'_3 = q_3, o_l \in O\}$, is the set of pick-up events, and $\Psi = \{(q, q') : q_1 \neq \emptyset, q'_1 = \emptyset, q'_2 = q_2 \cup q_1, q'_3 = q_3\}$ is the set of drop-off events. With the sensor paradigm (1), detection events occur, whenever the distance between the current robot position and the position of an object that has not been detected so far becomes ρ. Pick-up events occur, whenever the robot reaches the position of an object that has not been collected so far. Drop-off events occur, whenever the robot reaches the depot and carries objects. For both pick-up and drop-off events, zero velocity is required. This is captured by the invariant map $\text{Inv} : Q \rightarrow 2^X$ with

$$\text{Inv}(q) = \begin{cases} X \backslash \{[y^T\ v^T]^T : \|y - p^{(l)}\| = \rho\} & \text{if } \{o_l\} \notin q_3, \\ X \backslash \{[{p^{(l)}}^T\ \mathbf{0}_2^T]^T\} & \text{if } \{o_l\} \notin q_1 \cup q_2, \\ X \backslash \{\mathbf{0}_4\} & \text{if } q_1 \neq \emptyset, \end{cases}$$

the guard map $G : E \rightarrow 2^X$ with

$$G(e) = \begin{cases} \{[y^T\ v^T]^T : \|y - p^{(l)}\| = \rho\} & \text{if } e \in \Delta, q'_3 \backslash q_3 = \{o_l\}, \\ \{[{p^{(l)}}^T\ \mathbf{0}_2^T]^T\} & \text{if } e \in \Pi, q'_1 \backslash q_1 = \{o_l\}, \\ \{\mathbf{0}_4\} & \text{if } e \in \Psi, q_1 \neq \emptyset, \end{cases}$$

and the (trivial) reset map $R : E \times X \rightarrow X$ with $R(e, x) = x, \forall (e, x) \in E \times G(e)$. Note that we do not account for collisions between the objects and the robot, since both are assumed to be points in $\mathcal{Y}_g$. A practical setup that satisfies this assumption is, e.g., a quadrotor exploring a two-dimensional ground space from above. As the robot is assumed to start at the depot with zero velocity, and no objects have been picked up or dropped off before that, the initial state set is $\text{Init} = \{(q(0), (x(0), \mathcal{Y}(0)))\} = \{((\emptyset, \emptyset, \emptyset), (\mathbf{0}_4, \mathcal{Y}_g \setminus \mathcal{O}(\mathbf{0}_2))\}$.

Solving the addressed problem involves L detection, L pick-up and between 1 and L drop-off events, as it can be advantageous to collect several objects and then drop them simultaneously off at the depot. Hence, for the total number of events Λ, $2L < \Lambda \leq 3L$ holds, and the time line τ is accordingly partitioned into Λ intervals, i.e. $\tau = (\tau_1 = [t_0, t_1], \ldots, \tau_\Lambda = [t_{\Lambda-1}, t_\Lambda]), t_0 = 0, t_\Lambda = t_f$. The input is an ordered set of functions $u = (u_1, \ldots, u_\Lambda)$, where $u_\lambda : \tau_\lambda \rightarrow U$ are absolutely continuous for $\lambda \in \{1, \ldots, \Lambda\}$. Thus, if $\zeta = (\tau, q, \xi)_u$ is an execution of H for an input signal u, $q = (q^1, \ldots, q^\Lambda)$ is a discrete state trajectory with $q^\lambda : \tau_\lambda \rightarrow Q, q^\lambda = \text{const}, \forall t \in \tau_\lambda$. $\xi = (x, \mathcal{Y})$ is the continuous state trajectory with $x = (x_1, \ldots, x_\Lambda)$, $x_\lambda : \tau_\lambda \rightarrow X$ are absolutely continuous functions. $\mathcal{Y} = (\mathcal{Y}_1, \ldots, \mathcal{Y}_\Lambda)$ is a sequence of non-increasing functions $\mathcal{Y}_\lambda : \tau_\lambda \rightarrow 2^{\mathcal{Y}_g}$. The cost of ζ is

$$t_f = \sum_{\lambda=1}^{\Lambda} (t_\lambda - t_{\lambda-1}). \quad (3)$$

One way to account for the uncertainty is to minimize (3), while assuming that the locations of all objects that have not yet been discovered yield the worst-case cost. Alternatively, one can assume that the positions of the objects that have not been detected so far are independent identically distributed random variables with probability density functions

$$\mathcal{P}(p^{(l)}) = \begin{cases} \frac{1}{\kappa(t)}, & \text{if } o_l \notin q_3(t), \\ 1, & \text{if } o_l \in q_3(t) \wedge o_l \notin q_1(t) \wedge o_l \notin q_2(t), \\ 0, & \text{else}, \end{cases} \quad (4)$$

$\forall l$, where $\kappa(t)$ is the Lebesgue measure of $\mathcal{Y}(t)$. This leads to the following worst-case (A) and probabilistic (B) OCPs.

Problem. *At $(q(t), \xi(t))$, find the input signal $u|_{[t,t_f]}$ producing the execution $\zeta|_{[t,t_f]}$ of the hybrid automaton H, such that for $p = \{p^{(l)} | o_l \in O \setminus q_3(t)\}$,*

$$A) \min_{u|_{[t,t_f]}} \max_{p} \ t_f - t, \quad \text{or}$$

$$B) \min_{u|_{[t,t_f]}} \ E\{t_f - t\},$$

$$\text{s.t. } q(t_f) = (\emptyset, O, O), x(t_f) = \mathbf{0}_4.$$

Note that Problem A is deterministic, while Problem B becomes deterministic once the last object is discovered. The outline of the solution reads as follows. First, we derive a lower bound for the cost-to-go. Then, we propose two approaches for solving Problems A and B in a receding horizon manner – one based on space and time discretization and one based on motion parameterization, which allows for an event-driven implementation, i.e., the OCPs are re-solved only upon a detection.

III. Preliminary analysis

Consider an assignment with two objects, i.e., $O = \{o_1, o_2\}$, and the discrete dynamics of the corresponding

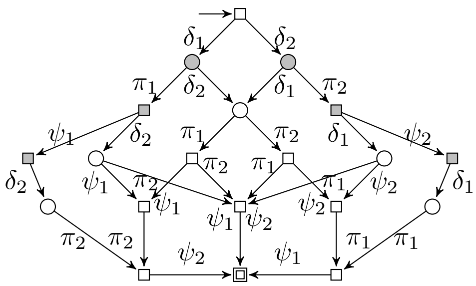

Fig. 1. Discrete dynamics for $O{=}\{o_1, o_2\}$, $\delta_l{\in}\Delta$, $\pi_l{\in}\Pi$, $\psi_l{\in}\Psi$. Gray nodes denote exploration, while square nodes correspond to zero velocity.

hybrid automaton $\mathcal{H}$ (Fig. 1). Let $t' \in [0, t_f]$ be the time when the robot has zero velocity and may thus perform a pick-up or drop-off (such states are denoted by a square in Fig. 1) and let $t \in [0, t')$. Then, the overall cost-to-go at $\xi(t)$ is the sum of the cost ℓ until the robot stops moving at t' and the remaining time for reaching Fin, i.e.,

$$J(\xi(t), u|_{[t,t_f]}, p){=}\ell(\xi(t), u|_{[t,t']}){+}J(\xi(t'), u|_{[t',t_f]}, p). \quad (5)$$

Assuming that all objects have been detected prior to t', the second term on the right hand side of (5) corresponds to the cost for the optimal sequence of pick-up and drop-off events necessary for completing the overall task. Let the set of all feasible discrete state sequences from $q(t') = q$ to the final discrete state q_f be denoted by $\Sigma_q := \{\sigma = q_0 q_1 \dots q_d : q_i = (q_i, \xi_i), q_{i+1} = (q_{i+1}, \xi_{i+1}), (q_i, q_{i+1}) \in (\Pi \cup \Psi), i \in \{0, \dots, d-1\}, q_d = (q_f, \xi(t_f)), q_0 = q\}$. Optimizing $J_\sigma(\xi(t'), u|_{[t',t_f]}, p)$ for a particular $\sigma \in \Sigma_q$ can be decoupled in terms of the input $u|_{[t',t_f]}$ at every pick-up and drop-off, i.e.,

$$J_\sigma(\xi(t'), p){=}\sum_{i=1}^{d-1} \min_{u|_{[t_i,t_{i+1}]}} J(\xi(t_i), u|_{[t_i,t_{i+1}]}, p). \quad (6)$$

Since $(q(t_i), x(t_i) = [y^T\ \mathbf{0}_2^T]^T), x(t_i) \in G(e)$ and $(q(t_{i+1}), x(t') = [y'^T\ \mathbf{0}_2^T]^T), x(t') \in G(e'), e \neq e', e, e' \in \Pi \cup \Psi$, for any term of the sum (6)

$$\min_{u|_{[t_i,t_{i+1}]}} J(\xi(t_i), u|_{[t_i,t_{i+1}]}, p) \geq 2\sqrt{\frac{m_{q(t)}}{u_{\max}}\|y - y'\|} \quad (7)$$

holds. Thus, we obtain

$$J(\xi(t'), u|_{[t',t_f]}, p) \geq J_{lb}(\xi(t'), p) = \min_{\sigma \in \Sigma_q} J_\sigma(\xi(t'), p). \quad (8)$$

Intuitively, this lower bound corresponds to bypassing the exploratory states in $\mathcal{A}_H$, such that exploration does not contribute to the cost-to-go. Clearly, (8) holds for an arbitrary number of objects and allows for the approximate solutions of Problems A and B described in the following.

IV. Space-discretized approach

Let a complete regular discretization of $\mathcal{Y}_g$ be given by a finite number of squares $d\mathcal{Y}_g^k$ with center points $y_{\mathcal{Y}_g}^k \in \mathcal{Y}_g$, such that $\cup_k d\mathcal{Y}_g^k \subseteq \mathcal{Y}_g, k \in \{1, \dots, K\}$ and $\forall y \in d\mathcal{Y}_g^k, \|y - y_{\mathcal{Y}_g}^k\| \ll \rho$. This allows to keep track of the evolution of the continuous state $\mathcal{Y}(t)$ by means of an under-approximation of the already explored space.

A. Worst-case solution

By applying the min-max inequality, (8) and since $\ell \geq 0$, $J_{lb} \geq 0$, for the (certainty equivalent) worst-case evaluation of (5), we obtain

$$\begin{aligned} t_f^{\mathrm{w}} - t &= \min_{u|_{[t,t_f]}} \max_p \ J(\xi(t), u|_{[t,t_f]}, p) \\ &\geq \max_p((\min_{u|_{[t,t']}} \ \ell(\xi(t), u|_{[t,t']})) + J_{lb}(\xi(t'), p)) \\ &\geq \min_{u|_{[t,t']}} \ \ell(\xi(t), u|_{[t,t']})) + \max_p J_{lb}(\xi(t'), p). \end{aligned} \quad (9)$$

Initially, $\xi(t) = \xi(t') = ((\emptyset, \emptyset, \emptyset), (\mathbf{0}_4, \mathcal{Y}_g \setminus \mathcal{O}(\mathbf{0}_2))$. Then, by treating the positions of the objects p as free variables over $\mathcal{Y}(t)$, i.e., $p = \{p^{(l)} \in \mathcal{Y}(t) | o_l \notin q_3(t)\}$, we solve

$$p^* = \arg\max_p \min_{\sigma \in \Sigma_q} J_\sigma(\xi(t'), p), \quad (10)$$

which can be formulated as a Mixed-Integer Program (MIP), and corresponds directly to the initial optimal control $u|_{[0,t_f^{\mathrm{w}}]}$. Once the robot starts moving, the control is acquired by solving the following OCP. Given the solution of (10), approximating (9) is formulated as a Mixed-Integer Linear Program (MILP)

$$\begin{aligned} &\min_{u|_{[t,t']}} (t' - t), \\ \text{s.t. } &\forall i, \forall q, b_{i,q} \implies x_{i+1} = f_q(x_i, u_i); \\ &\forall i, \forall q, b_{i,q} \in \mathcal{B}; x_{t'} = [y^T\ \mathbf{0}_2^T]^T, y \in p^*, \end{aligned} \quad (11)$$

over an equidistant discretization of $u|_{[t,t']}$ with $i \in [0, N_{\text{max}} - 1], N_{\text{max}} \in \mathbb{N}$, where $x_{i+1} = f_q(x_i, u_i)$ is a discrete-time version of (2) and $\mathcal{B}$ is the feasible Boolean constraint set corresponding to the discrete dynamics of H. For brevity, further implementation details are omitted and we refer the reader to [19] for a closely related optimization implementation. The OCPs are solved repetitively at every time instant until all objects are detected. The algorithm terminates in finite time, since $\mathcal{Y}(t)$ is decreasing over time.

B. Probabilistic solution

With (4), (5) and (8), the optimal cost in the probabilistic case is given by the minimum expected time

$$\begin{aligned} E\{t_f{-}t\}{=}&\min_{u|_{[t,t_f]}} E\{J(\xi(t), u|_{[t,t_f]}, p)\} \\ \geq &\min_{u|_{[t,t']}} \underbrace{\frac{1}{\kappa(t)} \iint_{\mathcal{Y}(t)} \ell dy}_{E\{\ell\}} + \min_{\sigma \in \Sigma_q} \underbrace{\frac{1}{\kappa(t)} \iint_{\mathcal{Y}(t)} J_\sigma dp}_{E\{J_\sigma\}}. \end{aligned} \quad (12)$$

The Lebesgue measure at time t is $\kappa(t) = \mathcal{A}_{\mathcal{Y}_g} / \mathcal{A}_{\mathcal{Y}(t)}$, where $\mathcal{A}_{(.)}$ denotes the area of the corresponding region. By applying Jensen's inequality to the first term in (12), we obtain the lower bound $E\{\ell\} \geq \ell(\xi(t), u|_{[t,t']}, E\{p\})$. Intuitively, this lower bound denotes the cost for a scenario where all objects that have not been detected yet are located at the point in the unexplored region with minimal distance to the depot. To compute the control $u|_{[t,t']}^*$ that minimizes the first term in (12), we again formulate a MILP of the form (11) for a discrete-time version of the robot's dynamics with the corresponding switching constraints of H. The second term in (12) is obtained by computing the expected cost for each sequence $\sigma \in \Sigma_q$ through numerical integration over $p = \{p^{(l)} \in \mathcal{Y}(t) | o_l \notin q_3(t)\}$, followed by choosing the sequence $\sigma^*\ in \Sigma_q$ that yields the minimal cost. This allows for a receding horizon scheme that minimizes the cost-to-go at each time instant until all objects are detected. The termination of the algorithm in finite time follows from the fact that $\mathcal{Y}(t)$ decreases over time.

V. Solution by motion parameterization

The approaches presented in the previous section require solving computationally expensive MIPs at each time instant online. Ultimately, we want a tractable and scalable scheme that requires re-computation only upon detecting objects. In [20] it was shown that when the robot moves on a line, the optimal policy for multiple objects is given by complete exploration followed by an optimal pick-up and drop-off of all objects. For $\mathcal{Y}_g \subset \mathbb{R}^2$, this policy can be used to obtain an approximation for the cost-to-go, i.e.,

$$J(\xi(t), u|_{[t,t_f]}, p) \approx J_{ub}(\xi(t), p) = \hat{\ell}(\xi(t)) + \hat{J}(\xi(t_e), p), \quad (13)$$

$$\hat{\ell} = \begin{cases} 0, & \text{if } q_3(t) = O, \\ t_e - t, & \text{else,} \end{cases} \quad \hat{J} = \begin{cases} 0, & \text{if } q_3(t) = \emptyset, \\ J_{lb}(\xi(t_e), p), & \text{else.} \end{cases}$$

where $t_e \in [0, t_f - t)$ is the time for complete exploration, i.e. until all objects are discovered. Let the exploratory motion of the robot be restricted to a piecewise smooth curve $C(s)$, parameterized by the normed Euclidean arc-length $s \in [0,1]$, with the following assumptions:

(i) the robot with sensor footprint (1) and position $y(t) = C(s)$ covers $\mathcal{Y}_g$ completely for $s \in [0,1]$, i.e. $\cup_{s\in[0,1]} \mathcal{O}(C(s)) \supseteq \mathcal{Y}_g$;
(ii) all objects are located on C prior to their discovery, i.e. $p^{(l)} \in C, \forall o_l \in \tilde{p}, \tilde{p} = O \setminus q_3(t)$;
(iii) intermediate pick-up('s) and/or drop-off('s) before all objects are discovered may start at $C(s)$ only if the robot eventually returns to the spot s and continues following C, i.e., $\exists (y(t) = C(s) \wedge y(t+) \notin C(s+)) \implies (\exists y(t') = y(t) = C(s)), t \leq t'$, if $q_3(t) \neq O$.

Due to Assumption (i) keeping track of the continuous state $\mathcal{Y}(t)$ is possible by only saving the current position $s(t)$ along the curve. Assumption (ii) is motivated by the fact that $\mathcal{O}$ is typically much smaller than $\mathcal{Y}_g$. Thus, the time for an immediate pick-up upon a detection can be neglected and the cost-to-go can be evaluated along the curve, instead of over $\mathcal{Y}(t)$. Since displacement is a function of time, let the trajectory's time dependence be captured by the path coordinate $s(t)$. Assume that the path starts at $C(s(0) = 0) = C_0 \in \mathbb{R}^2$ and ends at $C(s(t_e) = 1) = C_f \in \mathbb{R}^2$. Since $s(t)$ is monotonically increasing for $t \in [0, t_e]$, $\dot{s}(t) \geq 0$ everywhere and $\dot{s}(t) > 0$ almost everywhere. The first derivative of the curve with respect to s is given by $C'(s) = \partial C / \partial s$, while the second derivative with respect to s is $C''(s) = \partial^2 C / \partial s^2$. Further, $\dot{s} = ds/dt$ and $\ddot{s} = d^2 s / dt^2$. Thus, for the velocity and the acceleration along the path C, we respectively obtain

$$\dot{C}(s) = \frac{dC(s(t))}{dt} = C'(s(t))\dot{s}(t), \quad (14)$$

$$\ddot{C}(s) = \frac{d^2 C(s(t))}{dt^2} = C''(s(t))\dot{s}(t)^2 + C'(s(t))\ddot{s}(t). \quad (15)$$

Using (15), (2) is equivalently restated as

$$m_{q(t)}(C''(s(t))\dot{s}(t)^2 + C'(s(t))\ddot{s}(t)) = u(s(t)). \quad (16)$$

Define a nonlinear substitution $\alpha(s) = \ddot{s}$ and $\beta(s) = \dot{s}^2$, leading to $\beta'(s) = 2\alpha(s)$. With the change of variables and the nonlinear transformation, the curve tracking problem

$$\begin{aligned} &\min_{\alpha,\beta,u} \int_t^{t_e} dt = \int_{s(t)}^{s(t_e)} \frac{1}{\dot{s}} ds = \int_{s(t)}^{1} \frac{1}{\sqrt{\beta(s)}} ds, \\ &\text{s.t. } (16), |u(s)| \leq u_{\max}, \\ &\qquad \beta(0) = \dot{s}_0^2, \beta(1) = \dot{s}_{t_e}^2, \beta(s) \geq 0, s \in [s(t), 1], \end{aligned} \quad (17)$$

is convex, since the cost is convex in terms of $\alpha(s)$, $\beta(s)$ and $u(s)$ and the constraints are linear [22].

Now the only remaining question is: What kind of curves satisfy Assumption (i) and can be traversed fast by the robot? Looking at the space-discretized solutions (Sec. VI), Lissajous curves are a good choice for the worst-case, i.e.

$$y(t) = C_1(s(t)) = [a_1 \sin(2\pi b_1 s), a_2 \sin(2\pi b_2 s)]^T. \quad (18)$$

Complete coverage of $\mathcal{Y}_g$ by a Lissajous curve is provided by a suitable choice of the real-valued parameters a_i and b_i, $i \in \{1,2\}$, which can be obtained, e.g. by applying Infinitesimal Perturbation Analysis yielding sensitivities of the objective function that can be used in a gradient-descent method [23]. For the probabilistic case, an Archimedean spiral appears to be a reasonable choice, i.e., in polar coordinates (r, s),

$$r(t) = C_2(s(t)) = a s(t), \quad (19)$$

where the parameter a is chosen such that $\mathcal{Y}_g$ is completely covered at $r(t_e) = a$. With that, the worst- and probabilistic case OCPs can be restated as follows.

A. Worst-case solution

With (13), initially, we only have to solve (17) for the curve (18) to obtain t_e, and the robot starts tracking C_1 with the corresponding control input $u(s)$. Upon a detection at time t with $(q(t), \tilde{\xi}(t) = (x(t), s(t)))$, to obtain the worst-case value of (13), we solve

$$J_{ub}(\xi(t), p) \approx \min_{\sigma}((t_e - t) + \max_{p} J_\sigma(\xi(t_e), p)),$$

with the constraints that follow from Assumptions (ii) and (iii), i.e., $p^{(l)} = C(\tilde{s}_l), \forall p^{(l)} \in \tilde{p}, \hat{s}_l \in (s(t), 1]$ and neglecting the cost for an immediate pick-up (the control solving the corresponding deterministic OCP can be obtained by standard approaches for point-to-point motion, e.g., [2]). Then, by substituting (18) in (8) obtained with (7), the OCP corresponds to solving an OCP for each $\sigma \in \Sigma_{q(t)}$ for $\hat{s} = [\hat{s}_1, \ldots, \hat{s}_{|\tilde{p}|}]^T$ by a gradient-projection algorithm, i.e.,

$$\hat{s}^{n+1} = \hat{s}^n - \eta^n \frac{\partial J_\sigma}{\partial \hat{s}}, \quad (20)$$

where $\{\eta^n\}, n = 1, 2, \ldots$ is an appropriate step size sequence, $\hat{s}^1$ was selected randomly and the algorithm terminates when $|\partial J_\sigma / \partial \hat{s}| < \epsilon$ (for a fixed threshold ϵ), followed by choosing the sequence σ^* that yields the minimal worst-case cost.

B. Probabilistic solution

Due to Assumptions (ii) and (iii), the expected value of (13) is evaluated over the curve C_2 instead of over $\mathcal{Y}(t)$. Since the used Euclidean arc-length parameterization $s(t)$ preserves the structure of $x(t)$ [24] and the acceleration constraints of the vehicle are symmetric, its dynamics along the curve can be reduced to a double integrator. Neglecting the optimal time for a pick-up of an object with $p^{(l)} \notin C_2$, we can employ the necessary optimality conditions that have been derived in [20] for a robot moving on a line. Thus, with $\kappa_C(t)$ denoting the arc-length of $C_2(s), s \in [s(t), 1]$, the exploration cost

$$E\{t_e - t\} \approx \frac{1}{\kappa_C(t)} \int_{s(t)}^{1} \hat{\ell} ds, \tag{21}$$

is minimized by

$$|u(s)| \leq \begin{cases} u_{\max}, & s \in [0, s_{\text{sw}}), \\ \frac{u_{\max}}{2m_q(1+\sqrt{2})}, & s \in [s_{\text{sw}}, 1], \end{cases} \tag{22}$$

with $s_{\text{sw}} = 1/(1 + 2(1 + \sqrt{2}))$. By solving (17) with the additional constraint (22), we obtain the expected exploration time $E\{t_e - t\}$ (and the initial control input u_1). Upon a detection at time t with $(q(t), \tilde{\xi}(t) = (x(t), s(t)))$, to obtain the overall optimal expected cost, for each $\sigma \in \Sigma_q$ we solve

$$E\{J_\sigma\} \approx \frac{1}{\kappa_C(t)} \int \cdots \int_{\hat{s}_l \in (s(t),1], l=1,\ldots,|\tilde{p}|} J_\sigma \, d\hat{s}_1 \ldots d\hat{s}_{|\tilde{p}|}, \tag{23}$$

with $p^{(l)} = C(\tilde{s}_l), \forall p^{(l)} \in \tilde{p}, \hat{s}_l \in (s(t), 1]$, and then pick the sequence σ^* that yields the minimal expected cost. For (19), (23) can be efficiently evaluated in polar coordinates: with $p^{(l_1)} \in \tilde{p}$ and $p^{(l_2)} \in \tilde{p}$, respectively denoted by $r_{l_1} = as_1$ and $r_{l_2} = as_2$, using the law of cosines and substituting the relation $|s_1 - s_2| = \tilde{s}$, each integral of a term of the form (7) can be obtained by

$$\int_{s(t)}^{1} \int_{0}^{1} 2a \sqrt{\frac{m_{q(t)}}{u_{\max}} \sqrt{s_1^2 + |s_1 - \tilde{s}|^2 - 2s_1 |s_1 - \tilde{s}| \cos(\pi \tilde{s})}} \, d\tilde{s} ds_1.$$

Remark. *As the masses of the objects and the limits of $\mathcal{Y}_g$ are known, it may be possible to neglect some feasible $\sigma \in \Sigma_q$ for the particular setup, if their lower bound is always larger than the lower bounds of all other sequences in Σ_q.*

C. Complexity

The space-discretized approaches are computationally expensive due to their NP-hardness and the generally increasing non-convexity of $\mathcal{Y}(t)$ during mission execution. In contrast, the parameterized approaches rely on solving a finite number of OCPs by gradient projection (worst-case) or evaluating integrals numerically (probabilistic case) with significantly lower computational cost and allowing for parallelization. For $|p| = |O \setminus q_2(t)|$ denoting the number of objects that have not yet been dropped off at the depot, the number of possible discrete state sequences is $|\Sigma_q| = |p|! 2^{(|p|-1)}$. In our implementation, the cost-to-go evaluation remained feasible for up to $|p| = 4$ for the space-discretized approaches, and for significantly larger $|p|$ for the parameterized solution.

VI. Numerical Example

1) Implementation: The methods were implemented in MATLAB, using the solver IPOPT for the MIPs of the space-discretized methods and the solver SeDuMi for the parameterized solutions. All computations were performed on an Intel® Core™ i7 2.20 GHz processor with 8 GB RAM.

2) Setup: Consider the position space $\mathcal{Y}_g = [-5, 5] \times [-5, 5]$ m containing the objects $O = \{o_1, o_2, o_3\}$ with $p^{(1)} = [-3.1, -3.1]^T, m^{(1)} = 1$ kg, $p^{(2)} = [1.9, -1.9]^T, m^{(2)} = 2$ kg, $p^{(3)} = [3, 3]^T, m^{(3)} = 1$ kg. The robot has nominal mass $m_\emptyset = 2$ kg and moves according to the vector field set $F = \{f_q\}_{q \in 2^O}$. The size of the sensor footprint is $\rho = 1$ m. The parameters of (18) were chosen as $a_1 = a_2 = y_{\max} - 0.5\rho$, $b_1 = 4$ and $b_2 = 5$, and $a = 8\pi$ for (19). The parts of (19) outside of $\mathcal{Y}_g$ were obtained by projection and spline interpolation, while for the cost-to-go evaluation the borders of $\mathcal{Y}_g$ were neglected. A sampling time $t_s = 0.2$ s was chosen for the space-discretized approaches.

3) Analysis: Fig. 2 shows snapshots of the robot's motion at detection instants and the final time by applying the presented methods. The space-discretized and the parameterized methods lead to qualitatively similar solutions with small performance differences. The worst-case evaluation results in a more "cautious" solution by immediately picking up and dropping off objects at the depot. In contrast, the probabilistic evaluation typically leads to a "threshold-based" solution, where previously detected objects are collected in one sweep after longer exploration phases. It is notable that for 100 simulations with random placements of O in $\mathcal{Y}_g$, the probabilistic parameterized solution (which represents a greedy policy in terms of exploration) performed on average 7.8% better than the worst-case parameterized solution. On average, online re-computation took 24 s and 16 s for the worst-case and probabilistic space-discretized approaches (performed at every time instant), and 1 s and 0.5 s for the worst-case and probabilistic parameterized approaches (performed upon a detection), respectively, thus verifying that the latter two are suitable for real-time computation.

VII. Conclusions

A time-optimal hybrid control problem for a robot that has to find and collect a finite number of objects located in a two-dimensional space and move them to a depot has been addressed. Two approaches have been proposed for the worst- and a probabilistic case, assuming uniform distribution of the objects – one based on space and time discretization and one based on motion parameterization. The numerical case study reflected the computational advantage of the latter, while yielding similar qualitative results. The "cautious" worst-case policy results in an immediate pick-up and drop-off after a detection, while the "threshold-based" probabilistic policy consists of longer exploration phases and multiple successive pick-up's with a following drop-off. Future work will assess the performance of other parameterizations, e.g., Fourier curves, and address the presence of obstacles.

References

[1] A. E. Bryson and Y. C. Ho, *Applied optimal control : optimization, estimation, and control*. John Wiley and Sons, New York, 1975.

[2] D. Feng and B. H. Krogh, "Acceleration-constrained time-optimal control in n dimensions," *IEEE Trans. on Automatic Control*, vol. 31, no. 10, pp. 955–958, 1986.

[3] I. Mitchell, A. Bayen, and C. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.

[4] D. Applegate, R. Bixby, V. Chvátal, and W. Cook, *The Traveling Salesman Problem: A Computational Study*, ser. Princeton Ser. in Applied Mathematics. Princeton Univ. Press, 2011.

[5] V. Nenchev, C. Belta, and J. Raisch, "Optimal motion planning with temporal logic and switching constraints," in *14th European Control Conf. (ECC'15)*, 2015, pp. 1135–1140.

[6] B. Passenberg and O. Stursberg, "Graph search for optimizing the discrete location sequence in hybrid optimal control," in *3rd IFAC Conf. on Analysis and Design of Hybrid Systems*, 2009, pp. 304–309.

[7] C. G. Cassandras and J. Lygeros, Eds., *Stochastic hybrid systems*. CRC Press, 2010.

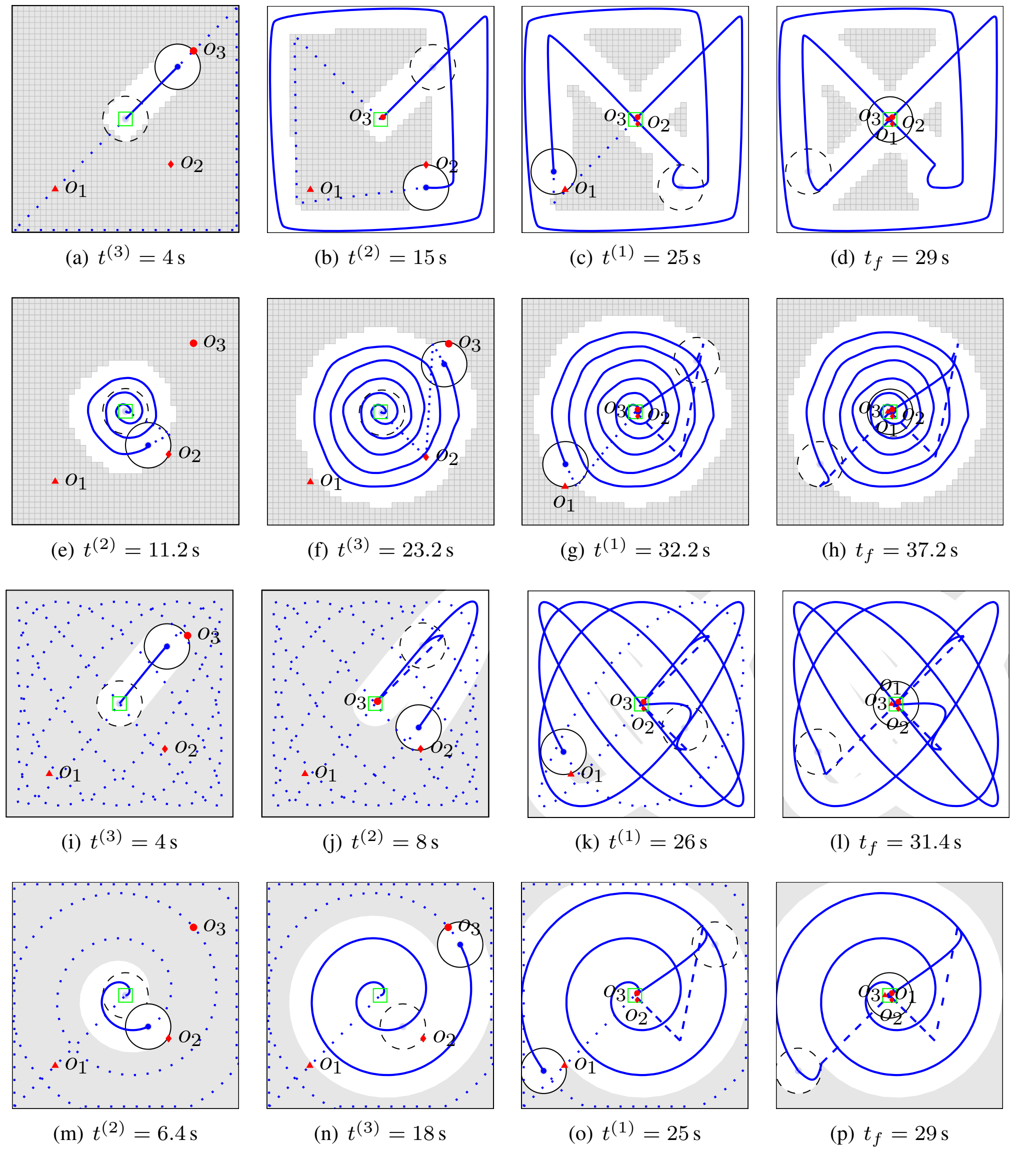

Fig. 2. Snapshots at detection instants $t^{(l)}$ and the final time t_f: space-discretized worst-case (a-d) and probabilistic (e-h); parameterized worst-case (i-l) and probabilistic (m-p). The path with nominal dynamics is denoted by solid, other dynamical modes by dashed, and planned paths by dotted curves.

[8] R. Bellman, "An optimal search problem," *SIAM Review*, vol. 5, 1963.

[9] L. J. Guibas, J.-C. Latombe, S. M. Lavalle, D. Lin, and R. Motwani, "A visibility-based pursuit-evasion problem," *International Journal of Computational Geometry and Applications*, vol. 9, pp. 471–494, 1996.

[10] S. Alpern and S. Gal, *The Theory of Search Games and Rendezvous*. Kluwer Int. Ser. in Operations Research & Management Science, 2003.

[11] M. Kao, J. Reif, and S. Tate, "Searching in an unknown environment: an optimal randomized algorithm for the cow-path problem," *Information and Computation*, vol. 131, no. 1, pp. 63–80, 1996.

[12] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[13] A. Molin and S. Hirche, "On the optimality of certainty equivalence for event-triggered control systems," *IEEE Trans. on Automatic Control*, vol. 58, pp. 470–474, 2013.

[14] H. Axelsson, M. Boccadoro, M. Egerstedt, P. Valigi, and Y. Wardi, "Optimal mode-switching for hybrid systems with varying initial states," *Nonlinear Analysis: Hybrid Systems*, vol. 2, no. 3, 2008.

[15] M. Rickert, A. Sieverling, and O. Brock, "Balancing exploration and exploitation in sampling-based motion planning," *IEEE Trans. on Robotics*, vol. 30, no. 6, pp. 1305–1317, 2014.

[16] M. Toussaint, "The bayesian search game," in *Theory and Principled Methods for Designing Metaheuristics*, ser. Natural Computing Series, Y. Borenstein and A. Moraglio, Eds. Springer, 2014, pp. 129–144.

[17] K. S. Wesselowski and C. G. Cassandras, "The elevator dispatching problem: Hybrid system modeling and receding horizon control," in *IFAC Analysis and Design of Hybrid Systems*, 2006, pp. 136–141.

[18] Y. Khazaeni and C. G. Cassandras, "A new event-driven cooperative receding horizon controller for multi-agent systems in uncertain environments," in *53rd CDC*, 2014, pp. 2770–2775.

[19] V. Nenchev and J. Raisch, "Towards time-optimal exploration and control by an autonomous robot," in *21st Mediterranean Conf. on Control and Automation (MED'13)*, 2013, p. 1236–1241.

[20] V. Nenchev and C. G. Cassandras, "Optimal exploration and control for a robotic pick-up and delivery problem," in *53rd Conf. on Decision and Control (CDC'14)*, 2014, pp. 7–12.

[21] C. Cassandras, X. Lin, and X. Ding, "An optimal control approach to the multi-agent persistent monitoring problem," *IEEE Trans. on Automatic Control*, vol. 58, no. 4, pp. 947–961, 2013.

[22] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Trans. on Automatic Control*, vol. 54, no. 10, pp. 2318–2327, 2009.

[23] X. Lin and C. G. Cassandras, "Trajectory optimization for multi-agent persistent monitoring in two-dimensional spaces," in *53rd Conf. on Decision and Control (CDC'14)*, 2014, pp. 3719–3724.

[24] A. A. Agrachev and Y. L. Sachkov, *Control theory from the geometric viewpoint*, ser. Enc. of mathematical sciences. Springer, 2004.