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

Vladislav Nenchev[a,*], Christos G. Cassandras[b,1], 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*

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.

Keywords: Optimal control, hybrid systems, motion control

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,

*Corresponding author
Email addresses: nenchev@control.tu-berlin.de (Vladislav Nenchev), cgc@bu.edu (Christos G. Cassandras), raisch@control.tu-berlin.de (Jörg Raisch)

[1] C. G. Cassandras was supported in part by NSF under grants CNS-1239021, ECCS-1509084, and IIP-1430145, by AFOSR under grant FA9550-15-1-0471, and by ONR under grant N00014-09-1-1051.

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 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 un-

der 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 Sec. 2, we present the problem formulation. Sec. 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 (Sec. 4) and the heuristic approaches (Sec. 5). The methods are then compared in a numerical example (Sec. 6), followed by the conclusions in Sec. 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 Sec. 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;
- $F = \{f_q\}_{q \in Q}$ is the finite set of vector fields that describe the evolution of the continuous state such that

$$\dot{x}(t) = f_q(x(t), u(t)), q \in Q, x(t) \in X, u(t) \in U;$$

- $E \subseteq Q \times Q$ is the set of discrete state transitions;
- $\text{Inv} : Q \to 2^X$ is the invariant map that describes the feasible continuous state domain associated with the discrete states;
- $G : E \to 2^X$ is the guard map that captures the continuous state domain, where transitions may occur;
- $R : E \times X \to 2^X$ is the reset map that describes changes of the continuous state upon a discrete state transition;
- $\text{Init} \subset Q \times X$ is the initial state set.

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 $Q \subseteq 2^O \times 2^O \times 2^O$. The current mass of the robot is $m_q(t) = m_\emptyset + \sum_{l, o_l \in q_1} m^{(l)}$, where $m_\emptyset$ is the nominal mass of the robot.

Consider a class of mobile robots capable of moving in any direction by directly controlling their constrained driving force (e.g. differential drive robots). For differentially flat dynamics, the robot state can be expressed by $x(t) = [y^T(t), v^T(t)]^T \in X$, where $v(t) \in \mathbb{R}^2$ is the current velocity of the robot, can be assumed to evolve according to a finite collection of vector fields $F = \{f_q\}_{q \in Q}$ with

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

driven by the piecewise continuous control signal $u : [0, t_f] \to U := \{\phi \in \mathbb{R}^2 : \|\phi\| \leq 1\}$, where t_f is the free final time for the assignment. The overall continuous state $(x(t), \mathcal{Y}(t))$ consists of the robot state $x(t)$ and the region $\mathcal{Y}(t) \subseteq \mathcal{Y}_g$ that has not been explored at time t.

As Q is finite, the set of discrete state transitions (or events) $E \subseteq Q \times Q$ is also finite. Let E be partitioned into $\Delta \cup \Pi \cup \Psi$, where for $q = (q_1, q_2, q_3), q' = (q'_1, q'_2, q'_3) \in Q$,

$$\Delta = \{(q, q') : q'_1 = q_1, q'_2 = q_2, q'_3 = q_3 \cup \{o_l\}, o_l \notin q_3\}$$

is the set of detection events,

$$\Pi = \{(q, q') : q'_1 \setminus q_1 = \{o_l\}, o_l \in q_3, q'_2 = q_2, q'_3 = q_3\},$$

is the set of pick-up events, and

$$\Psi = \{(q, q') : q_1 \neq \emptyset, q'_1 = \emptyset, q'_2 = q_2 \cup q_1, q'_3 = q_3\}$$

corresponds to the set of drop-off events.

Assume that the robot is equipped with an omni-directional sensor with a footprint of size $r \ll y_{\max}$ around its current position $y(t) \in \mathbb{R}^2$, hence covering the area

$$\mathcal{O}(y(t)) = \{y_p \in \mathbb{R}^2 : \|y(t) - y_p\| \leq r\}. \tag{2}$$

Thus, detection events occur when the distance between the current robot position and the position of an object that has not been detected so far becomes r. Pick-up events occur when the robot reaches the position of an object that has not been collected so far. Drop-off events occur when the robot reaches the depot and carries objects. In addition, for both pick-up and drop-off events, zero velocity is required. To illustrate these relations, consider a scenario with two objects and its corresponding discrete dynamics shown in Fig. 1. The conditions on q and x upon the occurrence of detection, pick-up and drop-off events are captured by the invariant $\text{Inv} : Q \to 2^X$, i.e.,

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

and the guard map $G : E \to 2^X$, i.e. with $e = (q, q')$,

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

For example, upon a detection of a new object as per (2), when the robot is in the discrete state q, the first case of the Inv requires that a transition must

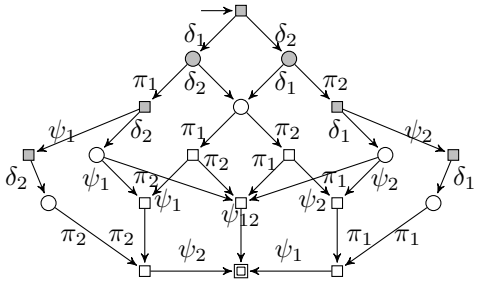


Figure 1: Discrete dynamics of $\mathcal{H}$ for $O{=}\{o_1, o_2\}$ with $\delta_l \in \Delta$, $\pi_l \in \Pi$, $\psi_l \in \Psi$, $l \in \{1, 2\}$, $\psi_{12} \in \Psi$. Exploration may take place at gray states. The robot has zero velocity at states denoted by a square.

occur, and the first case of G allows a transition to a discrete state q', where the discovered object is included in the detected objects set q'_3. The reset map $R : E \times X \to 2^X$ is trivial since no jumps of the continuous variables occur upon a discrete state switching. Note that the above conditions do not depend on $\mathcal{Y}(t)$, and hence, Inv, G and R map into 2^X. As the robot is assumed to start at the depot with zero velocity, and no objects have been detected, picked up or dropped off before that, the initial state set is

$$\text{Init} = \{(q(0), (x(0), \mathcal{Y}(0)))\} = \{((\emptyset, \emptyset, \emptyset), (\mathbf{0}_4, \mathcal{Y}_g \setminus \mathcal{O}(\mathbf{0}_2))\}.$$

As both the robot and the objects are represented by points in $\mathcal{Y}_g$, we assume that no collisions can occur. A practical setup that satisfies this assumption is, e.g., a quadrotor that has to explore a two-dimensional space on the ground from above.

Solving the addressed problem involves L detection events, L pick-up events and up to L drop-off events, as it can be advantageous to collect several objects on the way and drop them off simultaneously at the depot. Hence, for the total number N of events, $2L < N \leq 3L$ holds. The time of the occurrence of event n, $1 \leq n \leq N$ is denoted by t_n, t_0 is the initial time, $t_f = t_N$ the final time, and $t_0 \leq t_1 \leq \ldots \leq t_N$. The N time intervals $\tau_n := [t_{n-1}, t_n]$, $n = 1, \ldots, N$ form the time axis from the initial to the final time with $\tau := (\tau_1, \ldots, \tau_N)$. The input is an ordered set of functions $u = (u^1, \ldots, u^N)$, where $u^n : \tau_n \to U$ are absolutely continuous functions for $n \in \{1, \ldots, N\}$. Thus, if $\zeta = (\tau, q, \xi)_u$ is an execution of the hybrid automaton $\mathcal{H}$ for an input signal u, i.e. $(\tau, q, \xi)_u \vDash \mathcal{H}$, $q = (q^1, \ldots, q^N)$ is a discrete state trajectory with $q^n : \tau_n \to Q, q^n = \text{const.}$ $\xi = (\xi^1, \ldots, \xi^N)$ is the continuous state trajectory with $\xi^n = (x^n, \mathcal{Y}^n)$, where $x^n : \tau_n \to X$ are absolutely continuous functions, and $\mathcal{Y}^n : \tau_n \to 2^{\mathcal{Y}_g}$ non-increasing functions, i.e., $\mathcal{Y}^n(t') \subseteq \mathcal{Y}^n(t)$ for $t \leq t'$. Let $\mathcal{Z}$ denote the set of all feasible executions of the hybrid automaton $\mathcal{H}$. The cost of an execution is the

total task time

$$t_f = \sum_{n=1}^{N} (t_n - t_{n-1}) = t_N - t_0. \tag{3}$$

Further, let the set of states that can be reached upon completing the task be denoted by $Fin = \{(q_f = (\emptyset, O, O), (\mathbf{0}_4, \mathcal{Y}_g \setminus \cup_{\tilde{t}\in[0,t_f]} \mathcal{O}(y(\tilde{t}))))\}$.

One way to account for the uncertainty in the addressed OCP is to minimize, the largest cost at time t that may occur for a possible configuration of all objects that have not been discovered so far. Alternatively, the positions of the objects that have not been detected so far can be assumed to be independent identically distributed random variables with probability density functions

$$\mathcal{P}(p^{(l)}) = \begin{cases} \frac{1}{\kappa(t)}, & \text{if } p^{(l)} \in \mathcal{Y}(t), \\ 0, & \text{if } p^{(l)} \in \mathcal{Y}_g \setminus \mathcal{Y}(t), \end{cases} \tag{4}$$

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

Problem 1. *At state $(q(t), \xi(t))$, find the input signal $u|_{[t,t_f]}$ for $\mathcal{H}$, such that for $p = \{p^{(l)} : o_l \notin q_3(t)\}$*

$$A) \min_{u|_{[t,t_f]}} \max_{p} \ (t_f - t), \ \textit{s.t.} \ (q(t_f), (x(t_f), \mathcal{Y}(t_f))) \in Fin;$$

$$B) \min_{u|_{[t,t_f]}} \ E\{t_f - t\}, \ \textit{s.t.} \ (q(t_f), (x(t_f), \mathcal{Y}(t_f))) \in Fin.$$

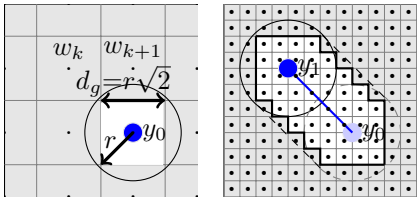
Note that Problem A is always deterministic, while Problem B is probabilistic until the last detection of an object.

Before turning to the solution, we introduce a finite approximation of the explored space and restrict the motion of the robot to a family of curves, whose shape is determined by a finite parameter vector. Thereby, we can derive approximate optimality conditions for the motion of the robot along the curve. This, on the other hand, allows for an event-driven scheme by an iterative gradient-based optimization over the parameters of the curve only upon detection events.

3. Preliminaries

To obtain a finite conservative approximation for $\mathcal{Y}(t)$, introduce a finite cover of $\mathcal{Y}_g$ by cells $\omega_k, k \in \{1 \dots, K\}$ defined by a set of grid points $W = \{w_1, \dots, w_K\}$, equally spaced by $d_g \leq r\sqrt{2}$, such that $\omega_k = \{y \in \mathcal{Y}_g : \|y - w_k\|_\infty \leq d_g/2\}$. An example is shown in Fig. 2. Let $\mathcal{W}(t)$ denote the set of grid points, whose associated cells have not been completely covered by the robot's sensing range (2) until time t, i.e. $\mathcal{W}(t) = \{w_i \in W : \omega_i \not\subseteq \cup_{\tilde{t}\in[0,t]} \mathcal{O}(y(\tilde{t}))\}$. Thus, $\mathcal{Y}(t)$ is over-approximated by $\tilde{\mathcal{Y}}(t) = \{\cup_i \omega_i : w_i \in \mathcal{W}(t)\}$.

Consider a continuous curve $c(s, \theta)$ in $\mathcal{Y}_g$, where $s(t), t \in [0, t_f]$ denotes the time-dependent position along the curve c, such that $s(0)=0$ and $s(t_f) = 1$,



(a) (b)

Figure 2: A robot with sensing radius r over the coarsest allowed grid (a). A snapshot of the robot that has moved from $y_0 = \mathbf{0}_2$ to $y_1 \neq \mathbf{0}_2$ with $d_g < r\sqrt{2}$ (b). The area covered along the path is under-approximated over the grid.

and $\theta \in \mathbb{R}^a$ is a parameter vector that determines its shape. Let $\tilde{s}(t)$ be the monotonically non-decreasing curve length function of c over $t \in [0, t_f]$. With $\alpha \in \mathbb{R}_{>0}$ denoting the arc-length of c, let $s(t) = \frac{\tilde{s}(t)}{\alpha}$ denote the normalized arc-length variable. Thus, the robot's position can be described by the parametric equation

$$y(t) = c(s(t), \theta) = \begin{bmatrix} c_1(s(t), \theta) \\ c_2(s(t), \theta) \end{bmatrix} \in \mathcal{Y}_g. \tag{5}$$

To apply gradient-based optimization, we further require that c is twice continuously differentiable with respect to s and θ. The parametric functions employed in this work are Fourier series (see Appendix A for a particular example) that exhibit rich expressiveness in terms of motion behaviors and allow for an efficient solution of the optimization problem. Other types of parametric functions or more complex robot dynamics may also be used, as outlined in Remark 1.

4. Event-driven approach

Consider a finite approximation of the explored space and restrict the motion of the robot to a parameterized curve, as described in the previous section. With that, the optimal motion of the robot along the curve can be obtained for a fixed shape of the curve. This OCP is then embedded into a simultaneous OCP, which provides a solution that satisfies all remaining constraints for solving Problem 1. The simultaneous OCP is solved by an augmented Lagrangian method that allows for replacing the constrained optimization problem by a series of unconstrained optimization problems. Employing Infinitesimal Perturbation Analysis [35], we obtain the derivative of the augmented cost and solve the unconstrained OCPs by gradient-based methods. The simultaneous problem is solved repeatedly until reaching a (local) minimum, which is attained upon satisfying an iteration threshold condition. We first turn to the the embedded problem.

4.1. Embedded problem

Let the first and second derivatives of (5) w.r.t. s be $c'(s, \theta) = \partial c / \partial s$ and $c''(s, \theta) = \partial^2 c / \partial s^2$, respectively. Further, let $\dot{s} = ds/dt$ and $\ddot{s} = d^2 s / dt^2$

denote the time derivatives. For the velocity and the acceleration along (5), we respectively obtain

$$\begin{aligned}\dot{c}(s,\theta) =& \frac{dc(s(t),\theta)}{dt} = c'(s,\theta)\dot{s},\\ \ddot{c}(s,\theta) =& \frac{d^2c(s(t),\theta)}{dt^2} = c''(s,\theta)\dot{s}^2 + c'(s,\theta)\ddot{s},\end{aligned}$$

and the robot's dynamics (1) are restated as

$$m_{q(t)}(c_i''(s,\theta)\dot{s}^2 + c_i'(s,\theta)\ddot{s}) = u_i, i \in \{1,2\}. \tag{6}$$

With the employed arc-length parameterization, the robot traverses the curve with $\|c'(s,\theta))\| = \alpha$, where α is the arc-length of c. Substituting the relation $u = u_c(t)[\cos(\varphi), \sin(\varphi)]^T$ in (6) and since $\|\ddot{c}(s,\theta)\| = \|u\|/m_{q(t)} = u_c/m_{q(t)}$ holds along the curve, the dynamics of the robot (1) are equivalently restated by (5) and the state $\bar{x} = [\bar{x}_1, \bar{x}_2]^T = [s, \dot{s}]^T$ with dynamics

$$\begin{aligned}\dot{\bar{x}}(t) =& \bar{f}_q(\bar{x}(t), u_c(t), \theta)\\ =& \begin{bmatrix}\bar{x}_2 & \sqrt{\frac{u_c^2(t)}{\alpha^2 m_{q(t)}^2} - \frac{\left(c_1''c_2' - c_1'c_2''\right)^2\bar{x}_2^4}{\alpha^4}} - \frac{(c_1''c_1' + c_2''c_2')\bar{x}_2^2}{\alpha^2}\end{bmatrix}^T.\end{aligned} \tag{7}$$

To simplify the analysis in the following, the necessary optimality conditions for $u_c(t)$ are derived for

$$\dot{\bar{x}}(t) \approx \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\bar{x}(t) + \frac{1}{\alpha m_{q(t)}}\begin{bmatrix}0\\ 1\end{bmatrix}u_c(t), \tag{8}$$

which represents an approximation of (7) along smooth curves. Note that, for lines, $c_1'' = c_2'' = 0$ implies $c_1''c_2' - c_1'c_2'' = 0$ and $c_1''c_1' + c_2'c_2'' = 0$, and (8) describes the dynamics of the robot exactly. Since the sensor footprint (2) is typically much smaller than $\mathcal{Y}_g$, for evaluating the cost-to-go we neglect the sensing range of the robot and assume that prior to their discovery all objects are located on (5), i.e., $\forall l, \exists s_l \in [0,1]$ with $p^{(l)} = c(s_l, \theta)$. In what follows, we further assume that the simultaneous OCP (presented in the following section) provides an optimal parameter θ^*, such that the robot moving along (5) with θ^* plans to cover the remaining space $\tilde{\mathcal{Y}}(t)$ as long as there are objects to be detected, and passes through object locations that have been discovered previously but have not been picked-up yet. We start the analysis assuming that there is only one undetected object, i.e. $O = \{o\}$, with mass m located at an a priori unknown spot $s_o \in [0,1]$.

4.1.1. Optimal control for one object

The robot with dynamics (8) starts at $\bar{x}(0) = [0,0]^T$, $q(0) = (\emptyset, \emptyset, \emptyset)$ and $m_{q(0)} = m_\emptyset$. The optimal control $u_c|_{[0,t_f]}$ solving Problem 1 is divided into three parts: $u_c|_{[0,t_1)}$ denoting the control until detection, $u_c|_{[t_1,t_2)}$ the control until pick-up and $u_c|_{[t_2,t_f)}$ the control until drop-off. After the object is detected

at time t_1, when the robot moves with velocity $\bar{x}_2(t_1) \geq 0$, it can be reached at time t_2 with $\bar{x}_2(t_2) = 0$ by employing a time-optimal bang-bang control switching at time $\tilde{t}_1$ [2], i.e.,

$$u_c(t) = \begin{cases} -1, & t \in [t_1, \tilde{t}_1), \\ 1, & t \in [\tilde{t}_1, t_2). \end{cases}$$

Solving (8) with the above control and $m_{q(0)} = m_\emptyset$, and applying the boundary conditions for the object's pick-up $\bar{x}_1(t_1) = \bar{x}_1(t_2), \bar{x}_2(t_2) = 0$, yields the optimal cost

$$(t_2 - t_1) = \underbrace{(1 + \sqrt{2})}_{c_g} \alpha m_\emptyset \bar{x}_2(t_1). \tag{9}$$

Since the robot stops moving at t_2, steering it back to the depot in a time-optimal manner is given by bang-bang control. Since the cost of this segment is independent of $\bar{x}(t)$ for $t \in [0, t_2)$, it can be neglected for finding the optimal control $u_c^*|_{[0,t_1)}$. Thus, only the time until pick-up has to be considered, i.e. $t_2 = t_1 + c_g \alpha m_\emptyset \bar{x}_2(t_1)$ obtained from (9) and we will skip the time interval subscript in the following.

In the worst case (Problem 1.A), the object is located the furthest away from the initial point, i.e., at $s_o = 1$. To allow for a pick-up, the corresponding position of the robot $\bar{x}_1(t_2) = 1$ should be reached with zero velocity, i.e. $\tilde{x}_2(t_2) = 0$. Thus, the time-optimal worst-case control for the first segment is given by

$$u_{c,w}^*(\bar{x}) = \begin{cases} 1, & \text{if } \bar{x}_1 \in [0, 0.5), \\ -1, & \text{if } \bar{x}_1 \in [0.5, 1]. \end{cases} \tag{10}$$

In the probabilistic case (Problem 1.B), the object's location s_o is uniformly distributed over $[0, 1]$. To compute the corresponding optimal control $u_{c,p}^*$, introduce an additional state for the time $\tilde{x}_3 = t$, leading to an extended system state $\tilde{x} = [\bar{x}^T \ t]^T$ with dynamics (8) and $\dot{\tilde{x}}_3 = 1$. The expected time for picking up the object is given by

$$\begin{aligned} E\{t_2\} =& E\{\tilde{x}_3(t_1) + c_g \alpha m_\emptyset \tilde{x}_2(t_1)\} \\ =& \int_0^1 (\tilde{x}_3(t_1) + c_g m_\emptyset \alpha \tilde{x}_2(t_1)) ds_o. \end{aligned}$$

From the assumed sensor paradigm $s_o = x_1(t_1)$, and substituting the relation $d\tilde{x}_1(t_1) = \tilde{x}_2(t_1) dt_1$, we obtain

$$E\{t_2\} = \int_0^{\tilde{t}} (\tilde{x}_2 \tilde{x}_3 + c_g \alpha m_\emptyset \tilde{x}_2^2) dt_1,$$

where the lower integral limit is obtained from the initial condition $\tilde{x}(0) = [0, 0, 0]^T$, and the upper integral limit from $\tilde{x}_1(\tilde{t}) = 1$. The corresponding control

Hamiltonian is

$$H(\tilde{x},\lambda,u_{c,p}) = (\tilde{x}_2\tilde{x}_3+c_g\alpha m_\emptyset\tilde{x}_2^2)+\lambda^T(t)\begin{bmatrix} \tilde{x}_2 & \frac{u_{c,p}}{m_\emptyset\alpha} & 1 \end{bmatrix}^T, \tag{11}$$

with absolutely continuous costate dynamics

$$\dot{\lambda}(t) = -\frac{\partial H}{\partial \tilde{x}} = -\begin{bmatrix} 0 & 2c_g\alpha m_\emptyset\tilde{x}_2+\tilde{x}_3+\lambda_1 & \tilde{x}_2 \end{bmatrix}^T. \tag{12}$$

The boundary conditions at the free final time $\tilde{t}$ are given by $\tilde{x}_1(\tilde{t}) = 1$, $\tilde{x}_2(\tilde{t}) = 0$ and $\lambda_3(\tilde{t}) = 0$. Thus, the probabilistic OCP has been transformed into a nonlinear constrained OCP. Applying Pontryagin's Minimum Principle, there exists an optimal state $\tilde{x}^*$, a control $u^*_{c,p}$, and a nontrivial costate λ^* trajectory, such that $\forall t \in [0,\tilde{t})$,

$$H(\tilde{x}^*,\lambda^*,u^*_{c,p}) \leq H(\tilde{x}^*(t),\lambda(t)^*,u_{c,p}(t)), \tag{13}$$

leading to the following results.

Theorem 1. *For the optimal control until detection at time $\tilde{t}$, for $t \in [0,\tilde{t})$, $u^*_{c,p}(t) \in \left\{-1,-\frac{1}{2c_g},1\right\}$ holds.*

Theorem 2. *The optimal control until detection is given by*

$$u^*_{c,p}(\tilde{x}) = \begin{cases} 1, & \text{if } \tilde{x}_1 \in [0,\frac{1}{3+2\sqrt{2}}), \\ -\frac{1}{2+2\sqrt{2}}, & \text{if } \tilde{x}_1 \in [\frac{1}{3+2\sqrt{2}},1]. \end{cases} \tag{14}$$

The proofs can be found in Appendix B.

4.1.2. Control for multiple objects

For multi-object setups, we propose an approach to obtain the control analogously to the single-object case. While the worst-case optimal control is built from a finite sequence of bang-bang control segments, a computationally tractable scheme for the probabilistic control is derived as follows. Recall the robot moving in the position space shown in Fig. 2 (b) and consider a scenario with two objects. Like in the single-object case, the robot starts moving in the discrete state $q = (\emptyset,\emptyset,\emptyset)$. The possible discrete event strings until the robot comes to a halt for the first time consist of detecting an object followed by its immediate pick-up, i.e. $\delta_i\pi_i$, or detecting both objects and stopping at one of the two objects' positions, i.e. $\delta_i\delta_j\pi_k$, $i,k,j \in \{1,2\}, i \neq j$ (see Fig. 1). For both cases, we employ the control (14) for the time interval up to the detection of the first object, although for more than one object this policy may not be optimal. Now assume that object o_1 has just been detected. This triggers a re-planning resulting in a new policy that takes into account the acquired information since the last re-planning. Fig. 3(a) and 3(b) show two possible outcomes of the overall optimization scheme that provide complete exploration

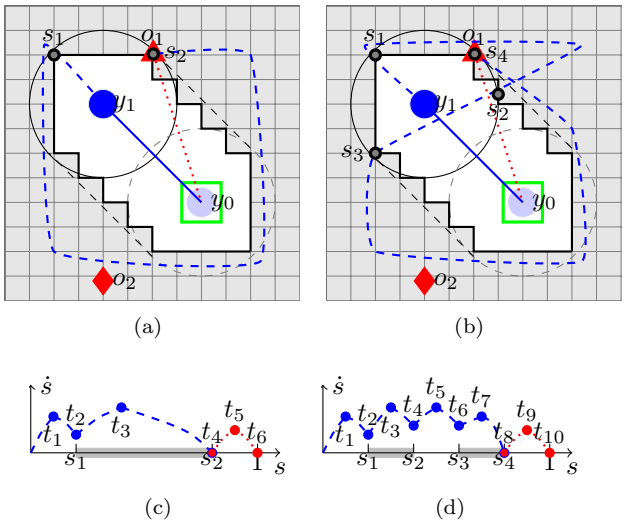


Figure 3: An exemplary scenario. The robot has moved from $y_0 = \mathbf{0}_2$ to $y_1 \neq \mathbf{0}_2$, where it has detected object o_1 denoted by ▲. Object o_2, denoted by ⧫, has not been discovered so far. Two possible motion plans for covering the remaining unexplored space are shown with dashed lines in (a) and (b), respectively. Both are followed by a pick up of o_1 and bringing o_1 back to the depot, depicted by a dotted line, which reflects that the robot is moving with a different mass in that segment. The generalized state trajectory $(s, \dot{s})|_{[0,t_f]}$ shown in (c) corresponds to the plan shown in a), while (d) represents the generalized state trajectory of the plan in (b). The gray marking in s denotes parts of the robot's path that lie in not yet explored areas, where a detection event may occur. In (a), s_1 stands for entering an unexplored region and s_2 – for the end of the overall exploration. In (b), s_1 denotes entering an unexplored region, s_2 – entering an already explored region, s_3 – entering an unexplored region, and s_4 – the end of the overall exploration. Note that the end of the overall exploration also denotes the worst-case position of o_2 with respect to the current plan.

of $\tilde{\mathcal{Y}}(t)$, allow for picking up o_1 and end at the depot. The trajectories $\bar{x}^*$ resulting from the probabilistic control are shown in Fig. 3(c) and 3(d), respectively. The particular control can be derived using the corresponding boundary and continuity conditions, as shown in Appendix C. The analysis for these two cases can be easily generalized to obtain the approximate probabilistic control for an arbitrary finite number of unexplored segments in the interval $[0, 1]$.

Moving on to a scenario with more than two objects, the controls along the curve can be obtained by following a similar line of argumentation – both the worst-case and the probabilistic controls consist of a finite sequence of appropriate bang-bang control segments. For that, consider the set of discrete state strings starting at the current state $q = q(t)$ and ending at the final state q_f, while neglecting all objects that have not been discovered so far, i.e.

$$\begin{aligned}\Sigma_q{:=}\{\sigma{=}q^0q^1\dots q^d : (q^{i-1},q^i){\in}(\Pi\cup\Psi), q_1^{i-1}{\subseteq}\tilde{q}^{i-1},\\ q_1^i{\subseteq}\tilde{q}^i, i\in\{1,\dots,d\}, q^d=q_f, q^0=q\},\end{aligned} \tag{15}$$

where $\tilde{q}^i{=}q_3^i\backslash(q_1^i{\cup}q_2^i)$ denotes objects that have been detected, but have not been picked up or dropped at the depot yet. For a given θ and with $t = t_0$, the curve (5) yields a discrete state string $\sigma \in \Sigma_q$ that is traversed in time

$$J_1{=}\sum_{i=1}^{d}(t_i{-}t_{i-1}){=}\sum_{i=1}^{d}\int_{\bar{x}_1(t_{i-1})}^{\bar{x}_1(t_i)}\frac{1}{\bar{x}_2}d\bar{x}_1. \tag{16}$$

Since minimizing (16) over the free parameters of the proposed policy, i.e. the switching times, can be decoupled at pick-up and drop-off instants, we solve $d-1$ Two-Point Boundary Value Problems (TPBVP's)

$$\begin{aligned}\bar{x}^*|_{[t_{i-1},t_i]} = \arg\min_{\bar{x}_1|_{[t_{i-1},t_i]}} \int_{\bar{x}_1(t_{i-1})}^{\bar{x}_1(t_i)}\frac{1}{\bar{x}_2}d\bar{x}_1,\\ \text{s.t. } (7), \bar{x}_2(t_{i-1})=\bar{x}_2(t_i)=0,\end{aligned} \tag{17}$$

using the corresponding control trajectory u_c for the worst-case or the probabilistic problem. A segment $\bar{x}^*|_{[t_{i-1},t_i]}$ of $\bar{x}^* = (\bar{x}^*|_{[t_0,t_1]},\dots,\bar{x}^*|_{[t_{d-1},t_d]})$ is obtained by solving an initial value problem backward and forward in time, and choosing the minimal velocity $\bar{x}_2$ along the curve. Clearly, once all objects are detected, both versions A) and B) of Problem 1 become deterministic and can be easily solved.

Remark 1. *The TPBVP* (17) *can be extended for general convex input constraints or nonlinear robot dynamics, as shown in [38]. If the curve is not continuously differentiable, an approximation can be obtained by a finite number of spline segments.*

Now the remaining task is solving the simultaneous optimization problem.

4.2. Simultaneous optimization

The solution of the simultaneous optimization problem yields a curve (5) that can be traversed optimally according to the policy outlined above, while providing complete exploration of the remaining space $\tilde{\mathcal{Y}}(t)$, passing through all positions of objects that have been discovered but have not been picked up yet and ending at the depot. Let these constraints be captured by the mapping $h : \mathcal{Z} \times \mathbb{R}^a \to \mathbb{R}^b$, where $h(\zeta, \theta) = \mathbf{0}_b$, if the constraints are satisfied, and $h(\zeta, \theta) \neq \mathbf{0}_b$, else. With (16) and (17), we solve the embedded OCP for a fixed θ, thus, obtaining the corresponding optimal execution ζ along the particular curve. The remaining task for the simultaneous OCP is to find the parameter vector for the curve with corresponding optimal execution that satisfies the aforementioned constraints. The corresponding bi-level optimization formulation at time t reads

$$\begin{aligned}(t_f - t)^* = \min_{\theta, \zeta \in \mathcal{Z}} \; & J_1(\zeta, \theta) \\ \text{s.t. } & \zeta \in \arg\min_{\bar{\zeta} \in \mathcal{Z}} \{J_1(\bar{\zeta}, \theta)\}, \\ & h(\zeta, \theta) = \mathbf{0}_b.\end{aligned} \tag{18}$$

Leveraging ideas from [39, 32], let $J_2 : \mathcal{Z} \times \mathbb{R}^a \to \mathbb{R}^b$ be a twice continuously differentiable version of the mapping h that captures the constraints of the simultaneous OCP, such that $J_2(\zeta, \theta) = \mathbf{0}_b$, if $h(\zeta, \theta) = \mathbf{0}_b$, and $J_2(\zeta, \theta) > \mathbf{0}_b$, else. We will now introduce the individual constraints of the optimization problem.

To guarantee that all objects are detected eventually, the robot has to observe every $w_{i_k} \in \mathcal{W}(t) \subset W, i_k \in \{1, \ldots, K\}$, when it moves along the curve. For every w_{i_k}, we define a function $d_k(\bar{x}, \theta)$, such that $d_k(\bar{x}, \theta) = 0$, if w_k is within the sensing range (2) of the robot with position $y(t) = c(\bar{x}_1, \theta)$, and $d_k(\bar{x}, \theta) > 0$, otherwise. The choice of $d_k(\bar{x}, \theta)$ is not unique. In particular, we employ

$$d_k(\bar{x}, \theta) = \begin{cases} 1 - \exp\left(-(D_k(\bar{x}_1, \theta) - r)^2\right), & \text{if } D_k(\bar{x}, \theta) > r, \\ 0, & \text{else,} \end{cases} \tag{19}$$

where $D_k(\bar{x}, \theta) = \|c(\bar{x}_1, \theta) - w_k\|$. This leads to the constraint vector $\mathcal{D}_{\mathcal{W}} = [\min_{\bar{x}_1 \in [0,1]} d_1(\bar{x}, \theta), \ldots, \min_{\bar{x}_1 \in [0,1]} d_{\tilde{K}}(\bar{x}, \theta)]$, $\tilde{K} = |\mathcal{W}(t)|$, which is required to be equal to $\mathbf{0}_{1,\tilde{K}}$, as every $w_{i_k} \in \mathcal{W}(t)$ has to be seen along the curve.

Let $\tilde{q} = q_3 \setminus (q_1 \cup q_2)$ denote all objects that have been detected, but have not been picked up or dropped at the depot yet. To solve the task, all objects $o_{i_l} \in \tilde{q}, i_l \in \{1, \ldots, \tilde{L}\}, \tilde{L} = |\tilde{q}|$, have to be eventually picked up, while the robot moves along the curve. Thus, in the simultaneous OCP we have to guarantee that the curve passes through all corresponding positions $p^{(i_l)}$. This requirement is captured by

$$\tilde{d}_l(\bar{x}, \theta) \begin{cases} = 0, & \text{if } p^{(i_l)} = c(\bar{x}_1, \theta), \\ > 0, & \text{else.} \end{cases}$$

Note that the choice of $\tilde{d}_l$ is not unique. In this work, we employ $\tilde{d}_l(\bar{x},\theta) = 1-\exp\left(-\|c(\bar{x}_1,\theta) - p^{(i_l)}\|^2\right)$. This leads to the constraint vector $\mathcal{D}_{q_3\setminus(q_2\cup q_1)} = [\min_{\bar{x}_1\in[0,1]} \tilde{d}_1(\bar{x},\theta),\dots,\min_{\bar{x}_1\in[0,1]} \tilde{d}_{\tilde{L}}(\bar{x},\theta)]$ that has to be equal to $\mathbf{0}_{1,\tilde{L}}$, such that every $p^{(i_l)}$ is visited at some point along the curve. Note that the constraint that the robot performs a pick-up with zero velocity is taken care of by the embedded OCP.

Further, upon a detection and, consequently, a re-computation, the robot is required to continue its motion in a smooth manner, despite changing from a curve characterized by the previous parameter θ^- at $\bar{x}^- = \bar{x}(t)$, to a curve characterized by the current parameter θ at $\bar{x} = [0,\bar{x}_2^-]^T$. Hence, introduce a function

$$d_0^y(\bar{x}^-,\theta^-,\bar{x},\theta)\begin{cases}=0, & \text{if } c(\bar{x}_1^-,\theta^-) = c(\bar{x}_1,\theta),\\ >0, & \text{else.}\end{cases}$$

for the position and, analogously, a function $d_0^v(\bar{x}^-,\theta^-,\bar{x},\theta)$ for the velocity. Note that both d_0^y and d_0^v are not unique. In particular, we employ

$$\begin{aligned} d_0^y(\bar{x}^-,\theta^-,\bar{x},\theta) =& 1-\exp\left(-\|c(\bar{x}_1^-,\theta^-) - c(\bar{x}_1,\theta)\|^2\right),\\ d_0^v(\bar{x}^-,\theta^-,\bar{x},\theta) =& 1-\exp\left(-\|c'(\bar{x}_1^-,\theta^-) - c'(\bar{x}_1^-,\theta)\|^2\right).\end{aligned}$$

In addition, to guarantee feasibility of the ODE, the discriminant in (7) has to be greater than or equal to 0, which can be captured by an expression similar to (19). Finally, the robot is required to return to the depot in order to drop all objects. Since the requirement that the drop-off is performed with zero velocity is taken care of by the embedded OCP, in the simultaneous OCP we have to guarantee that the curve ends at the depot. Thus, introduce the function $d(\theta)$, such that $d(\theta) = 0$, if $c(1,\theta) = y_d$, and $d(1,\theta) > 0$, otherwise. In particular, we use $d(\theta) = 1-\exp\left(-\|c(1,\theta)-y_d\|^2\right)$. Note that the choice of $d(\theta)$ is not unique. This leads to the constraint vector

$$\mathcal{D}_{0,f} = \begin{cases}[d_0^y, d_0^v, d], & \text{if } \bar{x}_2^2 \neq 0,\\ [d_0^y, d], & \text{else,}\end{cases}$$

which is required to be zero. Once all objects are discovered, exploration is no longer necessary and $\mathcal{D}_{\mathcal{W}}$ are neglected. Thus, the overall constraints are

$$J_2(\zeta,\theta) = \begin{cases}[\mathcal{D}_{\mathcal{W}}, \mathcal{D}_{q_3\setminus(q_2\cup q_1)}, \mathcal{D}_{0,f}]^T, & \text{if } q_3(t)\neq O,\\ [\mathcal{D}_{q_3\setminus(q_2\cup q_1)}, \mathcal{D}_{0,f}]^T, & \text{else.}\end{cases} \tag{20}$$

The constrained optimization problem (18) is approximately solved with (20) by the augmented Lagrangian method [40], yielding the iterative unconstrained optimization problem

$$(\theta_z^*,\zeta_z^*) = \arg\min_{\theta,\zeta} \hat{J} = J_1 + \frac{\mu_z}{2} J_2^T J_2 + \bar{\lambda}_z^T J_2, \tag{21}$$

where $\mu_z \in \mathbb{R}_{>0}$ is an optimization tuning variable that increases with each iteration and $\bar{\lambda}_z \in \mathbb{R}^b$ is an estimate of the Lagrangian multiplier, updated by

$$\bar{\lambda}_{z+1} = \bar{\lambda}_z + \mu_z J_2(\zeta_z^*, \theta_z^*) \text{ for every iteration } z = 0, 1, 2, \ldots \tag{22}$$

Since (20) is twice continuously differentiable w.r.t. s and θ, the unconstrained optimization problem (21) can be solved by gradient-based optimization. Taking into account the dynamics of the hybrid automaton $\mathcal{H}$ and substituting $\alpha = \|c'(\bar{x}_1, \theta)\|$ for the employed arc-length parameterization, the gradient of (21) (omitting function arguments) is given by

$$\begin{aligned} \nabla_\theta \hat{J} &= \sum_{n=1}^{N} \nabla_\theta \int_{t_{n-1}}^{t_n} dt + \sum_{\beta=1}^{b} (\mu_z J_{2,\beta} + \bar{\lambda}_{z,\beta}) \nabla_\theta J_{2,\beta} \\ &= \sum_{n=1}^{N} \nabla_\theta \int_{t_{n-1}}^{t_n} \frac{\|c'\|}{\alpha} dt + \sum_{\beta=1}^{b} (\mu_z J_{2,\beta} + \bar{\lambda}_{z,\beta}) \nabla_\theta J_{2,\beta}. \end{aligned} \tag{23}$$

Observing that $\bar{x}_1$ depends on θ through (7), we employ Infinitesimal Perturbation Analysis (IPA) [35] to obtain the gradient $\nabla_\theta \int_{t_{n-1}}^{t_n} \frac{\|c'\|}{\alpha} dt$ (and, thus, $\nabla_\theta \hat{J}$). Over an interval $\tau_n = [t_{n-1}, t_n)$, $n = 1, \ldots, N$, the evolution of $\bar{x}$ is described by the vector field $\bar{f}_{n-1}(t, \bar{x}, \theta)$ (with a slight abuse of notation). For $t \in [t_{n-1}, t_n)$,

$$\frac{d}{dt} \nabla_\theta \bar{x}(t, \theta) = \frac{\partial \bar{f}_{n-1}(t, \bar{x}, \theta)}{\partial \bar{x}} \nabla_\theta \bar{x}(t, \theta) + \nabla_\theta \bar{f}_{n-1}(t, \bar{x}, \theta) \tag{24}$$

holds with the boundary condition

$$\begin{aligned} \nabla_\theta \bar{x}(t_{n-1}^+, \theta) = \nabla_\theta \bar{x}(t_{n-1}^-, \theta) + \\ [\bar{f}_{n-2}(t_{n-1}^-, \bar{x}, \theta) - \bar{f}_{n-1}(t_{n-1}^+, \bar{x}, \theta)] \nabla_\theta t_{n-1}(\theta). \end{aligned} \tag{25}$$

Thus, we obtain

$$\nabla_\theta \bar{x}(t, \theta) = \nabla_\theta \bar{x}(t_{n-1}^+, \theta) + \int_{t_{n-1}}^{t} \frac{d}{dt} \nabla_\theta \bar{x}(t, \theta) dt. \tag{26}$$

Since $\bar{f}_{1,n-2}(t_{n-1}^-, \bar{x}, \theta) = \bar{f}_{1,n-1}(t_{n-1}^+, \bar{x}, \theta)$, from (25) we assert $\nabla_\theta \bar{x}_1(t_{n-1}^+, \theta) = \nabla_\theta \bar{x}_1(t_{n-1}^-, \theta)$. Using (24) with $\frac{\partial \bar{f}_{1,n-1}(t, \bar{x}, \theta)}{\partial \bar{x}_1} = 0$ and $\nabla_\theta \bar{f}_{1,n-1}(t, \bar{x}, \theta) = 0$ and (26), the gradient (23) (omitting function arguments) is obtained by

$$\nabla_\theta \hat{J} = \sum_{i=1}^{d} \int_{t_{i-1}}^{t_i} \nabla_\theta \frac{\|c'(\bar{x}_1, \theta)\|}{\alpha} dt + \sum_{\beta=1}^{b} (\mu_z J_{2,\beta} + \bar{\lambda}_{z,\beta}) \nabla_\theta J_{2,\beta}. \tag{27}$$

The partial derivatives of (19) w.r.t. s or θ (omitting function arguments) are obtained by

$$\nabla_{(.)} d_k = \begin{cases} 2(D_k - r) \exp\left(-(D_k - r)^2\right) \frac{\partial D_k}{\partial (.)}, & \text{if } D_k > r, \\ 0, & \text{else,} \end{cases} \tag{28}$$

where $\frac{\partial D_k}{\partial(.)}=\frac{1}{D_k}\left((c_1-w_{1,k})\nabla_{(.)}c_1+(c_2-w_{2,k})\nabla_{(.)}c_2\right)$. The corresponding derivatives of $\tilde{d}_l$ and d are obtained analogously. The required partial derivatives for Fourier series are shown in Appendix A. Then, to compute (27), we obtain the optimal solutions of the internal OCPs of the constraints $\mathcal{D}_{\mathcal{W}}$ by a gradient-based algorithm, i.e.,

$$s_{z+1} = s_z - \eta_z \left.\frac{\partial d_k(s_z,\theta)}{\partial s}\right|_{s\in[0,1]}, \tag{29}$$

where $\{\eta_z\}, z = 0,1,2\ldots$ is an appropriate step size sequence, $\left.\frac{\partial d_k}{\partial s}\right|_{s\in[0,1]}$ is the gradient projected onto the feasible interval $s \in [0,1]$ and the algorithm terminates when $\left|\left.\frac{\partial d_k}{\partial s}\right|_{s\in[0,1]}\right| < \epsilon$ (for a given threshold ϵ). Analogously, we solve the internal OCPs of the constraints $\mathcal{D}_{q_3\setminus(q_1\cup q_2)}$, yielding the value of J_2. Then, $\nabla_\theta J_2$ is computed by (28) for the (local) optima acquired by solving (29) for the corresponding constraints.

As the considered setup is static, the gradient obtained through IPA is a trivially unbiased estimate of the gradient of the cost for Problem B. Thus, the optimal parameter vector θ^* is obtained with a gradient-based algorithm, i.e.,

$$\theta_{z+1} = \theta_z - \eta_z \left.\nabla_\theta \hat{J}(\bar{x}^*,\theta_z)\right|_{\mathcal{Y}_g} \tag{30}$$

where $\{\eta_z\}, z = 0,1,2,\ldots$ is a properly selected step-size sequence and the gradient is projected onto the feasible position space $\mathcal{Y}_g$. The algorithm terminates when $|\nabla_\theta \hat{J}(\bar{x}^*,\theta_z)|_{\mathcal{Y}_g}| < \epsilon$ for a pre-specified threshold ϵ. Note that the OCPs (29) are non-convex, the solution of (21) acquired by (30) will, in general, be only locally optimal.

The overall event-driven solution is summarized in Alg. 1. Upon a detection of an object, as long as the current solution violates the constraints, (17) and (30) are solved iteratively, where μ_z increases with each iteration, thus increasing the importance of J_2 over J_1 in the optimization.

Remark 2. *Under the presence of an obstacle obs with a priori unknown size and location, the obstacle is approximated by all w_k of the discretization of $\mathcal{Y}_g$ that have been covered by the sensor until time t and belong to the obstacle region, i.e. all $w_k \in \cup_{\tilde{t}\in[0,t]}\mathcal{O}(y(\tilde{t})) \wedge w_k \in obs$. Then, J_2 is augmented by a term that is only active in the surroundings $d_s = d_g/\sqrt{2}$ of the corresponding w_k, $k \in \{1,\ldots,\tilde{K}\}$, and is continuously differentiable w.r.t. s or θ, e.g.,*

$$J_o = \sum_{k=1}^{\tilde{K}} \max\{0, 1-(D_k/d_s)^2)\}^2. \tag{31}$$

With the partial derivative

$$\nabla_{(.)}J_o = -\frac{4}{d_s^2}\sum_{k=1}^{\tilde{K}}\left[\frac{\partial D_k}{\partial(.)}\max\{0,1-\frac{D_k^2}{d_s^2}\}\right],$$

Algorithm 1 Event-driven receding horizon control

Input: Robot dynamics described by the hybrid automaton $\mathcal{H}$ with Init $= (q(t), (\bar{x}(t), \mathcal{W}(t)))$; curve $y(t) = c(\bar{x}_1, \theta)$ with initial parameter vector θ; optimization parameters $0 < \epsilon \ll 1$, $\nu > 1$

Output: The optimal control $u|^*_{[t,t_f]}$

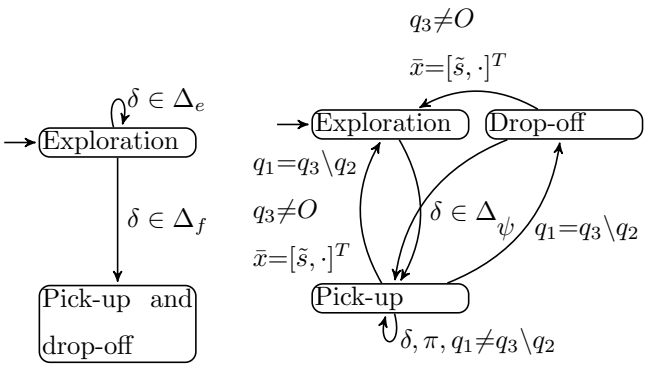
1: Set $\mu_1 = 1$ and $\bar{\lambda}_1 = \mathbf{0}_b$.
2: **while** $J_2(\bar{x}^*, \theta) > \epsilon \mathbf{1}_b$ **do**
3: **repeat**
4: Compute $u^*(\bar{x}^*)$ through (17) and J_2 and $\nabla_\theta J_2$ through (29) for θ for the worst-case or the probabilistic case.
5: Compute $\hat{J}$ and $\nabla_\theta \hat{J}|_{\mathcal{Y}_g}$ with μ_z, and update θ through (30) and $\bar{\lambda}_z$ through (22).
6: **until** $|\nabla_\theta \hat{J}|_{\mathcal{Y}_g}| < \epsilon$
7: Set $\mu_{z+1} = \nu \mu_z$.
8: **end while**
9: **return** $u|^*_{[t,t_f]} = u^*(\bar{x}^*)$

the (local) minimum of J_o^* is acquired by solving an OCP of the form (29)*, and the corresponding $\nabla_\theta J_o$ is used to solve* (30)*. Note that $J_o = 0$ and $\nabla_\theta J_o = \mathbf{0}_a$, if the curve does not intersect the obstacle. Alg. 1 remains unchanged. In general, the non-convexity of the optimization space increases with a growing number of obstacles, which may deteriorate the quality of the local minima obtained by Alg. 1. In general, in environments with high obstacle densities, one should choose parameterizations with a higher degree of freedom (i.e. larger Γ_1 and Γ_2 if using Fourier series) to allow for effective obstacle avoidance.*

Under the assumption that a solution of the simultaneous problem exists and, since the approximation of the equality constraints is sufficiently smooth and all involved functions are continuously differentiable w.r.t. the optimization variables in the approximate formulation, Alg. 1 converges to a locally optimal solution [41, 42]. In the numerical example presented in (Section 6), the algorithm typically terminated within a small number of iterations also for numerous variations of the setup. The explicit rate of convergence will be the subject of future work. The interested reader is referred to [41] for an in-depth discussion on the properties of the employed optimization method. Further, to obtain a better local solution, a Stochastic Comparison Algorithm (SCA) [43] was employed, i.e. multiple optimizations with different (quasi-)random initial conditions were run in parallel and the solution that yielded the lowest cost was chosen.

5. Alternative heuristic approaches

This section introduces two heuristic event-driven policies for solving Problem 1. For both approaches, let the exploratory motion of the robot be parameterized by a curve (5), where θ is fixed and assume that the robot with sensor footprint (2) covers $\mathcal{Y}_g$ completely for $s \in [0, 1]$, i.e. $\cup_{s\in[0,1]} \mathcal{O}(c(s, \theta)) \supseteq \mathcal{Y}_g$.



(a) Heuristic 1. (b) Heuristic 2.

Figure 4: Heuristic policies.

5.1. Complete exploration followed by pick-up and drop-off

A simple approach for Problem 1 is to explore the environment until all objects are discovered, followed by an optimal pick-up and drop-off of all objects. The corresponding bimodal policy can be captured by the automaton depicted in Fig. 4(a). The set Δ_e contains detection events before the last object is detected, i.e. $\Delta_e = \{(q, q') \in \Delta : q'_3 \neq O\}$, and $\Delta_f = \Delta \setminus \Delta_e$ are all other detection events. In exploration mode, the robot is simply following the curve that provides a complete cover of $\mathcal{Y}_g$, until the occurrence of a detection event $\delta \in \Delta_f$ at time $t' \in (0, t_f]$, which implies that the last object was found and triggers a transition to pick-up and drop-off mode. Since all objects have been detected prior to t', the remaining cost denotes the minimum time for a sequence of pick-ups and drop-offs necessary for completing the overall task. Consider the set of all corresponding strings Σ_q from $q = q(t')$ to q_f as defined in (15). Minimizing the cost J_σ of $\sigma \in \Sigma_q$ can be decoupled in terms of the input $u|_{[t',t_f]}$ at every pick-up and drop-off time instant $t_{i-1}, t_i \in [t', t_f]$ due to the zero velocity requirement, i.e.,

$$\begin{aligned} J^*_\sigma(q(t'), \xi(t')) &= \min_{u|_{[t',t_f]}} J_\sigma(q(t'), \xi(t'), u|_{[t',t_f]}) \\ &= \sum_{i=1}^{d} \min_{u|_{[t_{i-1},t_i]}} J(q_{i-1}, \xi(t_{i-1}), u|_{[t_{i-1},t_i]}) \end{aligned}$$

with $t_0 = t'$ and $t_d = t_f$. Assuming the absence of obstacles in $\mathcal{Y}_g$, the time-optimal motion of the robot with dynamics (1) from the hybrid state $(q_{i-1}, (x(t_{i-1}) = [y(t_{i-1})^T, \mathbf{0}_2^T]^T, \mathcal{Y}(t_{i-1})))$ to $(q_i, (x(t_i) = [y(t_i)^T, \mathbf{0}_2^T]^T, \mathcal{Y}(t_i)))$ with $y(t_i), y(t_{i-1}) \in p \cup \{\mathbf{0}_2\}$ is on straight lines. Thus, using an affine transformation, (1) can be reduced to a double integrator in one dimensional space. The OCP for the reduced model corresponds to the classical linear time-OCP [2] solved by a piecewise constant control that takes values in the set $\{\pm 1\}$ and

yields the optimal cost $t_i - t_{i-1} = 2\sqrt{m_{q_{i-1}}\|y(t_i)-y(t_{i-1})\|}$. The controller can be transformed back to (1) by using the inverse affine transformation (details can be found in [10]). Thus, the optimal cost for $\sigma \in \Sigma_q$ is

$$J^*_\sigma(q(t'), \xi(t')) = \sum_{i=1}^{d} 2\sqrt{m_{q_{i-1}}\|y(t_i)-y(t_{i-1})\|},$$

and the optimal cost in pick-up and drop-off mode is

$$J^*(q(t'), \xi(t')) = \min_{\sigma \in \Sigma_q} J^*_\sigma(q(t'), \xi(t')).$$

Its solution can be acquired by a standard search algorithm in the corresponding finite weighted graph.

5.2. Immediate pick-up followed by exploration and drop-off

Another option for solving Problem 1 is to pick-up objects immediately upon their detection, followed by planning an optimal exploration of the remaining not yet covered space and eventually dropping-off all currently carried objects, until all objects are discovered. The policy, captured by the automaton in Fig. 4(b), has been introduced in [36] and will only be briefly summarized here. The robot starts in exploration mode, i.e. following the curve until an object detection δ occurs at the spot $\tilde{s} \in [0, 1]$ along the curve, thus triggering a transition to pick-up mode. In pick-up mode, the robot moves to the location of the detected object in minimum time, which corresponds to a TPBVP solved by standard methods. If the robot is in pick-up mode and another detection occurs, it remains in this mode until all detected objects that have not been picked up and dropped at the depot yet, i.e. $q_3 \setminus (q_2 \cup q_1)$ are picked up, i.e. until $q_1 = q_3 \setminus q_2$. Then, the robot may either return to the depot to drop-off all currently carried objects, or resume exploration, by returning to the spot $\tilde{s}$, where it has left the curve to perform a pick-up, and continues following the curve for $s \in [\tilde{s}, 1]$ to guarantee complete coverage. After the last detection, the robot performs a pick-up and returns to the depot for drop-off.

Remark 3. *If the same initial parameterization* (5) *is employed for all approaches, the two heuristic policies are possible solution outcomes of the event-driven scheme proposed in Section 4. Thus, their evaluation will be performed in terms of a different, widely common path for exploration – the Archimedean spiral – in the next section. Under the presence of obstacles, when the robot moves along a fixed exploratory curve, guaranteeing complete coverage of $\mathcal{Y}_g$ and obstacle avoidance simultaneously is a non-trivial requirement, which will not be addressed in this paper.*

6. Numerical example

The methods were implemented in MATLAB, using *ode*45 to obtain the motion trajectories of the robot along the curve. All computations were performed on an Intel® Core™ i7 2.20 GHz processor with 8 GB RAM.

	event-driven		heuristic 1	heuristic 2
	worst-case	probabilistic		
t_f [s]	32.3	30.1	40.4	38.5
t_{comp} [s]	6.1	5.3	3.1	4.4

Table 1: Average final time t_f and average (re-)computation time t_{comp} over 100 random placements of the three objects.

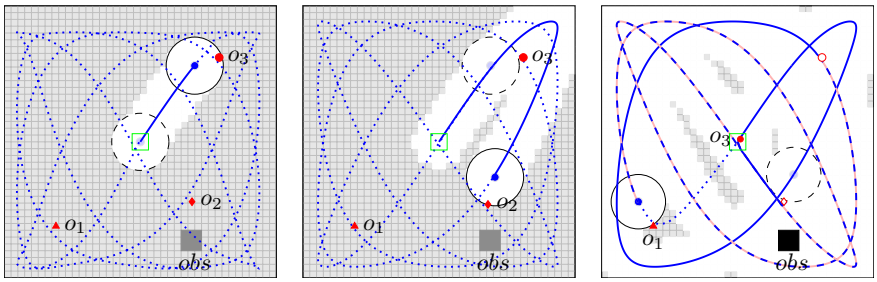
Consider the bounded position space $\mathcal{Y}_g = [-5, 5] \times [-5, 5]\,\text{m}$ and its regular discretization (as described in Section 3) with a grid constant of 0.25 m. The object set consists of three objects, i.e. $O = \{o_1, o_2, o_3\}$ with positions initially unknown to the robot. The robot with sensor footprint of size $r = 1\,\text{m}$ and nominal mass $m_\emptyset = 2\,\text{kg}$ starts at the depot $y_d = \mathbf{0}_2$ at rest. A static obstacle with a priori unknown location and size to the robot, is described by

$$\text{obs}= \left\{ y \in \mathbb{R}^2 \colon \begin{bmatrix} \mathbf{I}_2 \\ -\mathbf{I}_2 \end{bmatrix} y \leq \begin{bmatrix} \mathbf{I}_2 \\ -\mathbf{I}_2 \end{bmatrix} \begin{bmatrix} 1.875 \\ -3.625 \end{bmatrix} + \begin{bmatrix} \mathbf{I}_2 \\ \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} 0.375 \\ 0.375 \end{bmatrix} \right\}.$$

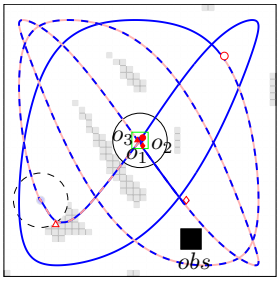
Note that the obstacle is neglected in the heuristic event-driven approaches. Fourier series of order $\Gamma_1 = \Gamma_2 = 3$ were used for the event-driven approach. Optimization was performed with parameters $\nu = 2, \epsilon = 10^{-3}$, and a maximal integration time of 40 s for *ode*45. The parameter vector θ, which characterizes the curve before the first detection, is obtained by running Alg. 1 for 100 random initializations and selecting the best (local) optimum.

For a scenario, where the masses and actual object's positions are $p^{(1)} = [-3.1, -3.1]^T$, $m^{(1)} = 1\,\text{kg}$, $p^{(2)} = [1.9, -1.9]^T$, $m^{(2)} = 2\,\text{kg}$ and $p^{(3)} = [3, 3]^T$, $m^{(3)} = 2\,\text{kg}$, Fig. 5 shows snapshots of the robot's motion at object detection instants and the final time, obtained by applying the presented methods. Executed paths with nominal dynamics are denoted by solid, executed paths in other dynamical modes by thicker dashed, and planned trajectories by dotted lines. As it can be seen in the plots, the robot successfully avoids the obstacle – snapshots at obstacle detection instants were omitted for brevity. Fig. 6 depicts the outcome of the heuristic methods for the same scenario.

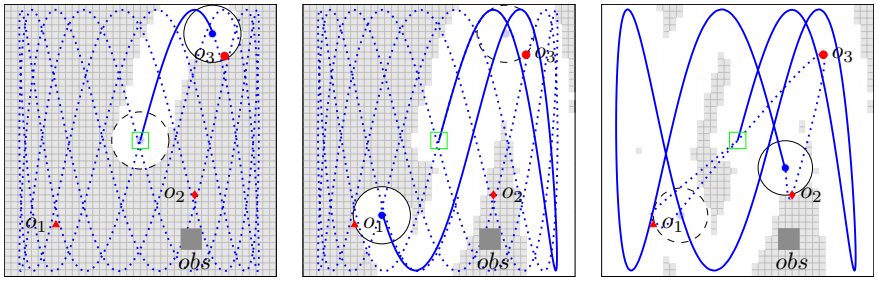
For 100 random placements of the three objects in $\mathcal{Y}_g$, the corresponding average final time t_f and computation costs t_{comp} are summarized in Table 6. In general, the worst-case solution resulted in a more "cautious" policy including intermediate pick-ups and drop-offs. In contrast, the probabilistic evaluation typically lead to a "threshold-based" policy, where previously detected objects are collected in one sweep after longer exploration phases. On average, the probabilistic solution performed better than the worst-case event-driven solution. Further, the heuristic approaches performed worse than the event-driven ones in terms of average final cost, but required a lower computational effort. Note that the convergence speed and the quality of the result of the event-driven approaches strongly depend on the chosen initial conditions and the step size selection method of the gradient optimization procedures.



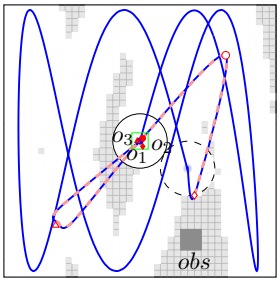
(a) $t^{(3)} = 3.8\,\mathrm{s}$ (b) $t^{(2)} = 9\,\mathrm{s}$ (c) $t^{(1)} = 30.2\,\mathrm{s}$



(d) $t_f = 36.2\,\mathrm{s}$

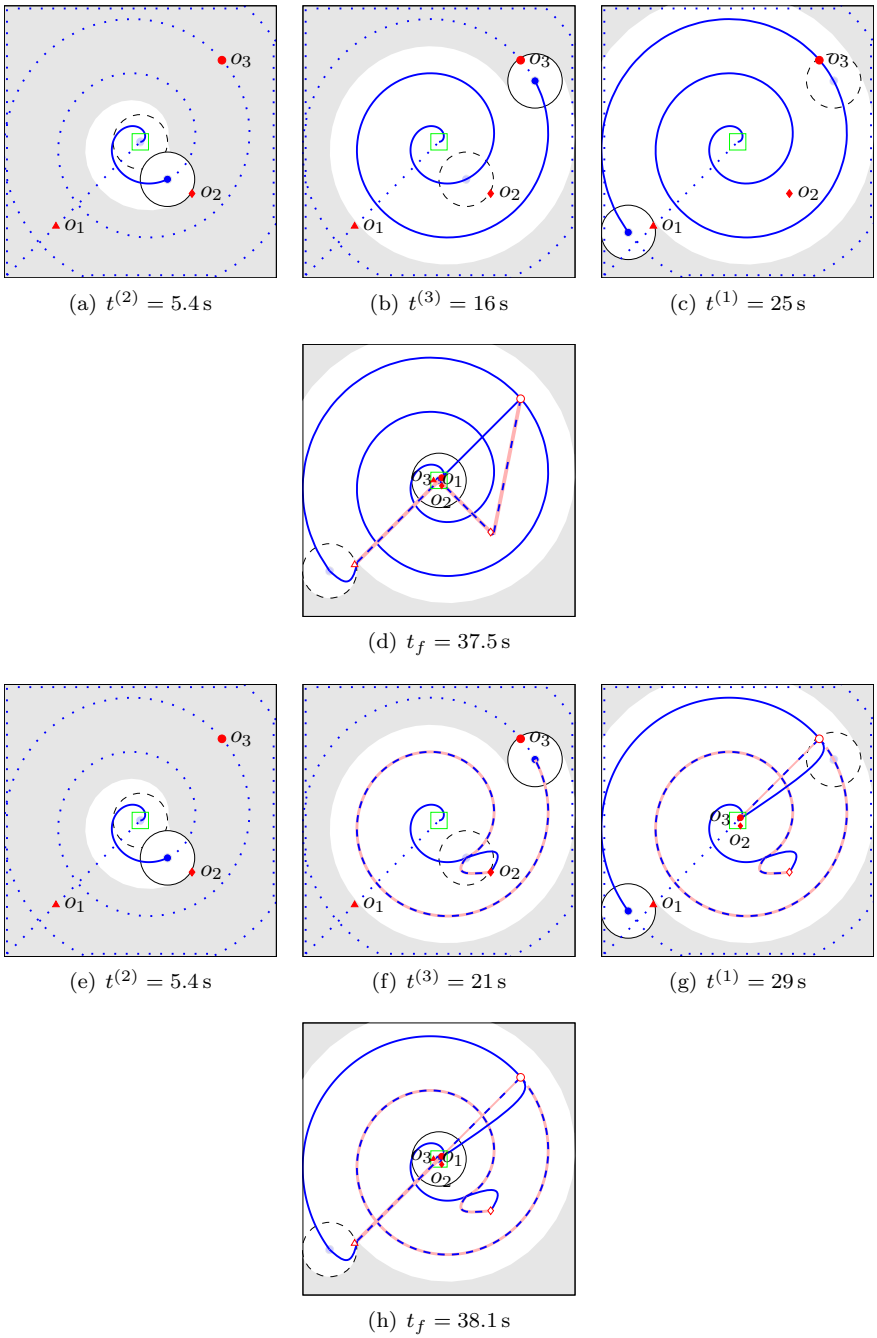


(e) $t^{(2)} = 4.4\,\mathrm{s}$ (f) $t^{(3)} = 14\,\mathrm{s}$ (g) $t^{(1)} = 25.4\,\mathrm{s}$



(h) $t_f = 32.2\,\mathrm{s}$

Figure 5: Snapshots of the robot's motion at detection instants $t^{(l)}$ and the final time t_f obtained with the event-driven worst-case (a-d) and probabilistic (e-h) methods, respectively.



(a) $t^{(2)} = 5.4\,\mathrm{s}$ (b) $t^{(3)} = 16\,\mathrm{s}$ (c) $t^{(1)} = 25\,\mathrm{s}$

(d) $t_f = 37.5\,\mathrm{s}$

(e) $t^{(2)} = 5.4\,\mathrm{s}$ (f) $t^{(3)} = 21\,\mathrm{s}$ (g) $t^{(1)} = 29\,\mathrm{s}$

(h) $t_f = 38.1\,\mathrm{s}$

Figure 6: Snapshots of the robot's motion at detection instants $t^{(l)}$ and the final time t_f obtained with the heuristic policy 1 in (a-d) and 2 in (e-h), respectively.

7. 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. An event-driven approach based on motion parameterization and gradient-based optimization has been proposed, where the cost was evaluated for the worst- and a probabilistic case, assuming uniform distribution of the objects over the restricted position space. The methods were compared to two heuristic approaches in a numerical example, reflecting the advantages of the event-driven method – a lower execution time and the ability to handle obstacles – over the heuristic approaches at the price of a moderately higher computational effort. Future work will address a multi-robot setup, other probability distributions and alternative optimization techniques, e.g. using the Alternating Direction Method of Multipliers (ADMM) [44], and how different cost functions can be considered in a similar manner.

Appendix A: Fourier series trajectories

In the event-driven approach, the position of the robot is parameterized by Fourier series of respective order Γ_1 and Γ_2, i.e.,

$$y(t) = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} a_0^1 + \sum_{\gamma=1}^{\Gamma_1} a_\gamma^1 \sin\left(4\pi^2 \gamma f_1 s(t) + \phi_\gamma^1\right) \\ a_0^2 + \sum_{\gamma=1}^{\Gamma_2} a_\gamma^2 \sin\left(4\pi^2 \gamma f_2 s(t) + \phi_\gamma^2\right) \end{bmatrix},$$

where f_1 and f_2 are base frequencies, a_0^1 and a_0^2 are zero frequency components, a_γ^1 and a_γ^2 are amplitudes for the sinusoid functions with frequency γf_1 and γf_2, and ϕ_γ^1 and ϕ_γ^2 are phase differences with respect to the $(\gamma+1)$-th term of y_1 or y_2. Since only the ratio of f_1 and f_2 (and not their absolute values) determines the shape of the trajectories, f_1 is treated as a free parameter, while f_2 is kept constant. With $A_1 = \left[a_0^1, \ldots, a_{\Gamma_1}^1\right]$, $A_2 = \left[a_0^2, \ldots, a_{\Gamma_2}^2\right]$, $\Phi_1 = \left[\phi_1^1, \ldots, \phi_{\Gamma_1}^1\right]$ and $\Phi_2 = \left[\phi_1^2, \ldots, \phi_{\Gamma_2}^2\right]$, the overall parameter vector is $\theta{=}[f_1, A_1, A_2, \Phi_1, \Phi_2]^T$. The derivative of the curve w.r.t. θ with $i \in \{1, 2\}$ is

$$\nabla_\theta y_i = \nabla_\theta c_i(s, \theta) = \begin{bmatrix} \frac{\partial y_i}{\partial f_1} & \frac{\partial y_i}{\partial A_1} & \frac{\partial y_i}{\partial A_2} & \frac{\partial y_i}{\partial \Phi_1} & \frac{\partial y_i}{\partial \Phi_2} \end{bmatrix}^T$$

with $\gamma = 1, \ldots, \Gamma_i$ and

$$\frac{\partial y_i}{\partial f_1} = \begin{cases} 4\pi^2 s(t) \sum_{\gamma=1}^{\Gamma_1} a_\gamma^1 \cos\left(4\pi^2 \gamma f_1 s(t) + \phi_\gamma^1\right), & i{=}1, \\ 0, & i{=}2, \end{cases}$$

$$\frac{\partial y_i}{\partial a_0^j} = \begin{cases} 1, & i{=}j, \\ 0, & i{\neq}j, \end{cases}, \quad \frac{\partial y_i}{\partial a_\gamma^j} = \begin{cases} \sin\left(4\pi^2 \gamma f_i s(t) + \phi_\gamma^i\right), & i{=}j, \\ 0, & i{\neq}j, \end{cases}$$

$$\frac{\partial y_i}{\partial \phi_\gamma^j} = \begin{cases} a_\gamma^i \cos\left(4\pi^2 \gamma f_i s(t) + \phi_\gamma^i\right), & i{=}j, \\ 0, & i{\neq}j, \end{cases}$$

With $i \in \{1,2\}$, for the derivative $\nabla_\theta \hat{J}$, we also need

$$c_i' = \sum_{\gamma=1}^{\Gamma_i} 4\pi^2\gamma f_i a_\gamma^i \cos\left(4\pi^2\gamma f_i s(t)+\phi_\gamma^i\right),$$
$$c_i'' = -\sum_{\gamma=1}^{\Gamma_i} 16\pi^4\gamma^2 f_i^2 a_\gamma^i \sin\left(4\pi^2\gamma f_i s(t)+\phi_\gamma^i\right).$$

$$\frac{\partial c_i'}{\partial f_1} = \begin{cases} \sum_{\gamma=1}^{\Gamma_i}(4\pi^2\gamma a_\gamma^i \cos\left(4\pi^2\gamma f_i s(t)+\phi_\gamma^i\right) \\ -16\pi^4\gamma^2 a_\gamma^i \sin\left(4\pi^2\gamma f_i s(t)+\phi_\gamma^i\right)), & i=1, \\ 0, & i=2, \end{cases}$$
$$\frac{\partial c_i''}{\partial f_1} = \begin{cases} \sum_{\gamma=1}^{\Gamma_i}(-32\pi^4\gamma^2 a_\gamma^i \sin\left(4\pi^2\gamma f_i s(t)+\phi_\gamma^i\right) \\ -64\pi^6\gamma^3 a_\gamma^i s(t) \cos\left(4\pi^2\gamma f_i s(t)+\phi_\gamma^i\right)), & i=1, \\ 0, & i=2, \end{cases}$$
$$\frac{\partial c_i'}{\partial a_0^j} = 0, \qquad \frac{\partial c_i''}{\partial a_0^j} = 0,$$
$$\frac{\partial c_i'}{\partial a_\gamma^j} = \begin{cases} \sum_{\gamma=1}^{\Gamma_i} 4\pi^2\gamma f_i \cos\left(4\pi^2\gamma f_i s(t)+\phi_\gamma^i\right), & i=j, \\ 0, & i\neq j, \end{cases}$$
$$\frac{\partial c_i''}{\partial a_\gamma^j} = \begin{cases} -\sum_{\gamma=1}^{\Gamma_i} 16\pi^4\gamma^2 f_i^2 \sin\left(4\pi^2\gamma f_i s(t)+\phi_\gamma^i\right), & i=j, \\ 0, & i\neq j, \end{cases}$$

$$\frac{\partial c_i'}{\partial \phi_\gamma^j} = \begin{cases} -\sum_{\gamma=1}^{\Gamma_i} 4\pi^2\gamma f_i a_\gamma^i \sin\left(4\pi^2\gamma f_i s(t)+\phi_\gamma^i\right), & i=j, \\ 0, & i\neq j, \end{cases}$$
$$\frac{\partial c_i''}{\partial \phi_\gamma^j} = \begin{cases} -\sum_{\gamma=1}^{\Gamma_i} 16\pi^4\gamma^2 f_i^2 a_\gamma^i \cos\left(4\pi^2\gamma f_i s(t)+\phi_\gamma^i\right), & i=j, \\ 0, & i\neq j. \end{cases}$$

When the robot moves along a curve with parameter vector θ_- and detects an object at $s_- = s(t)$, for the initial parameter vector of the simultaneous OCP, for $i \in \{1,2\}$, we set

$$\forall\gamma, \phi_\gamma^i = 4\pi^2\gamma f_i s_- + \phi_{\gamma-}^i,$$
$$a_0^i = c_i(s_-,\theta_-) - \sum_{\gamma=1}^{\Gamma_i} a_{\gamma-}^i \sin\left(\phi_\gamma^i\right),$$

such that only a small number of constraints of the simultaneous OCP is violated initially and Alg. 1 can converge fast.

Appendix B: Proofs

Proof. (Theorem 1) Since λ_2 is the only costate that depends on u_1 in (11), (13) implies that $\lambda_2^*(t)u_{c,p}^*(t) \leq \lambda_2^*(t)u_{c,p}$ must hold. For $\lambda_2(t) \neq 0, t \in [0,\tilde{t})$,

we obtain $u_{c,p}(t) = -\mathrm{sgn}(\lambda_2(t))$, i.e. $u^*_{c,p}(t) \in \{\pm 1\}$. As the input is bounded, $\lambda_2(t) = 0$ can hold at isolated times, without violating (13). A singular arc may exist since $\lambda_2(t) = 0$ can also hold over an interval. The necessary condition for this case is obtained from the generalized first-order Legendre-Clebsch condition [45]

$$\frac{d^2}{dt^2}\frac{\partial \tilde{H}}{\partial u_{c,p}} = \ddot{\lambda}_2(t) = 2c_g\alpha m_\emptyset \dddot{\tilde{x}}_2 + \ddot{\tilde{x}}_3 + \dot{\lambda}_1 = 0,$$
$$-\frac{\partial}{\partial u_{c,p}}\left[\frac{d^2}{dt^2}\frac{\partial \tilde{H}}{\partial u_{c,p}}\right] \neq 0,$$

yielding $-1/(2c_g)$ as an additional possible input value. □

Proof. (Theorem 2) From (11) and $\tilde{x}_1 \geq 0$, $\lambda_2(0) < 0$ holds and the optimal control starts with a fragment $u^*_{c,p}(t) = 1$ for $t \in [0, \bar{t}_1)$. Assuming that $\lambda_2(0) = a \in \mathbb{R}_{<0}$ for the interval $[0, \bar{t}_1)$, the adjoint variable is $\lambda_2(t) = a - \frac{2c+1}{2}t^{(2)} - \lambda_1 t$ by integration of (12). If $\lambda_2(\bar{t}_1) = \dot{\lambda}_2(\bar{t}_1) = 0$, $\lambda_1 = -\sqrt{2(1+2c)a}$, the set of possible control sequences is $\{(1), (1,-1), (1,-1/2c_g), (1,-1/2c_g,-1)\}$, or in generalized form

$$u_{c,p}(t) = \begin{cases} 1, & t \in [0, \bar{t}_1), \\ -\frac{1}{2c_g}, & t \in [\bar{t}_1, \bar{t}_2), \\ -1, & t \in [\bar{t}_2, \tilde{t}). \end{cases} \tag{32}$$

Integrating (1) with $m_q = m_\emptyset$ and (32), using the switching and final conditions $\tilde{x}_1(\tilde{t}) = 1, \tilde{x}_2(\tilde{t}) = 0, \tilde{x}_2(\bar{t}_2) = v_c$ and analyzing the expressions for $\bar{t}_2 = \bar{t}_1$ and $\bar{t}_2 = \tilde{t}$, we obtain

$$\sqrt{\frac{2(m_\emptyset\alpha + c_g\alpha^2 {m_\emptyset}^2 {v_c}^2)}{2c_g+1}} \leq \bar{t}_1 \leq \sqrt{m_\emptyset\alpha + \frac{m_\emptyset^2\alpha^2 v_c^2}{2}},$$

for $0 \leq v_c \leq \sqrt{\frac{2}{m_\emptyset\alpha}}$. Substituting the solutions of the ODE, the cost $E\{\ell\}$ becomes a function of $\bar{t}_1$ and v_c. For $\frac{\partial E\{\ell\}}{\partial \bar{t}_1} = 0$, $\bar{t}_1^* \in \left\{\sqrt{\frac{2\alpha m_\emptyset}{2c_g+1}}, \sqrt{m_\emptyset\alpha + \frac{\alpha^2 m_\emptyset^2 v_c^2}{2}}\right\}$ holds. Analyzing $\frac{\partial^2 E\{\ell\}}{\partial \bar{t}_1^2}$ in the feasible interval, we can verify that the first value corresponds to the minimum. Substituting $\bar{t}_1^*$ in $E\{\ell\}$ and evaluating $\frac{\partial E\{\ell\}}{\partial v_c}$, for the optimal velocity $v_c^* \in \left\{0, \sqrt{\frac{2}{m_\emptyset\alpha}}\right\}$ holds. Analyzing $\frac{\partial^2 E\{\ell\}}{\partial v_c^2}$ yields $v_c^* = 0$ and $\bar{t}_2 = 2c_g\sqrt{\frac{2m_\emptyset\alpha}{2c_g+1}}$, leading to $\bar{t}_2 = \tilde{t}$. Solving the ODE with (32) yields the switching point $1/(1+2c_g)$ and the corresponding controller. □

Appendix C: Optimal control for two objects

First, consider the planned path shown in Fig. 3(a) with corresponding generalized trajectory shown in Fig. 3(c). Since o_2 is discovered at latest when the

robot reaches s_2, $\bar{x}_2(t_4) = 0$. Thus, optimizing the trajectory for $t \in [t_4, t_6]$ can be decoupled, and the optimal control is given by

$$u_c^*(\bar{x}) = \begin{cases} 1, & \text{if } \bar{x}_1 \in [s_2, \frac{1+s_2}{2}), \\ -1, & \text{if } \bar{x}_1 \in [\frac{1+s_2}{2}, 1). \end{cases}$$

To obtain the optimal switching times of the remaining trajectory, we solve the ODE (8) with

$$u_{c,p}(t) = \begin{cases} 1, & \text{if } t \in [0, t_1), \\ -1, & \text{if } t \in [t_1, t_2), \\ 1, & \text{if } t \in [t_2, t_3), \\ -\frac{1}{2c_g}, & \text{if } t \in [t_3, t_4). \end{cases}$$

With the boundary conditions $\bar{x}_1(0) = \bar{x}_2(0) = 0$, and $\bar{x}_1(t_2) = s_1$, we obtain

$$-\frac{(t_2 - t_1)^2}{2m} + \frac{t_1}{m}(t_2 - t_1) + \frac{t_1^2}{2m} = s_1,$$

where $m = m_q \alpha$, and since $t_2 \geq t_1$,

$$t_1 = \begin{cases} t_2, & \text{if } t_2 \leq \sqrt{2ms_1}, \\ t_2 - \frac{\sqrt{2}}{2}\sqrt{t_2^2 - 2ms_1}, & \text{else.} \end{cases}$$

Using the boundary conditions $\bar{x}_1(t_4) = s_2, \bar{x}_2(t_4) = 0$, we obtain

$$-\frac{(t_4-t_3)^2}{4c_g m} + \frac{t_3+2t_1-2t_2}{m}(t_4-t_3) + \frac{t_3-3t_2+4t_1}{2m}(t_3-t_2) + s_1 = s_2,$$
$$t_4 = (2c_g + 1)t_3 - 4c_g(t_2 - t_1),$$

yielding

$$t_3 = 2(t_2 - t_1) + \sqrt{\frac{{t_2}^2 - 4t_1 t_2 + 4{t_1}^2 + 2m(s_2 - s_1)}{2c_g + 1}}.$$

Since $t_3 \in \mathbb{R}_+$ and $t_3 \geq t_2$, together with the derived expression for t_1, we obtain

$$t_3 = \begin{cases} \sqrt{\frac{t_2^2 + 2m(s_2 - s_1)}{2c_g + 1}}, & \text{if } t_1 = t_2, \\ t_2, & \text{else.} \end{cases}$$

From the continuity of the variables, there exists a time when $t_1 = t_2 = t_3 = \sqrt{\frac{m}{c_g}(s_2 - s_{sw})}$. Thus, the switching takes place at

$$s' = \frac{1}{2c_g + 1} s_2.$$

Note that this corresponds to the result of Theorem 2, scaled for the interval $[0, s_2]$. For $t_2 > \sqrt{2ms_1}$, substituting t_1 into the expression for $t_3 = t_2$, we obtain $t_2 = t_3 = \sqrt{\frac{m}{c_g}(s_2 - s_1)}$ and

$$t_{1,s} = \sqrt{\frac{m(s_2 - s_1)}{c_g}} - \frac{\sqrt{2}}{2}\sqrt{\frac{m(s_2 - s_1) - 2c_g m s_1}{c_g}}$$

yielding a switching at

$$s'' = -\frac{2^{\frac{3}{2}}\sqrt{s_2 - s_1}\sqrt{s_2 - (2c_g + 1)s_1} - 3s_2 + (2c_g + 3)s_1}{4c_g}$$

Thus, the optimal control is given by

$$u^*_{c,p}(t, \bar{x}) = \begin{cases} 1, & \text{if } \bar{x}_1 \in [0, s_{sw,1}), \\ -1, & \text{if } \bar{x}_1 \in [s_{sw,1}, s_{sw,2}), \\ -\frac{1}{2c_g}, & \text{if } \bar{x}_1 \in [s_{sw,2}, s_2), \end{cases} \tag{33}$$

where $s_{sw,1} = \max\{s', s_1\}$ and

$$s_{sw,2} = \begin{cases} s'', & \text{if } s_1 > s', \\ s', & \text{else.} \end{cases}$$

Now consider the planned path shown in Fig. 3(b) with corresponding generalized trajectory shown in Fig. 3(d). Analogously to the above analysis, it can be easily shown that the optimal control is

$$u^*_{c,p}(\bar{x}) = \begin{cases} 1, & \text{if } \bar{x}_1 \in [0, s_{sw,1}), \\ -1, & \text{if } \bar{x}_1 \in [s_{sw,1}, s_{sw,2}), \\ -\frac{1}{2c_g}, & \text{if } \bar{x}_1 \in [s_{sw,2}, s_{sw,3}), \\ 1, & \text{if } \bar{x}_1 \in [s_{sw,3}, s_{sw,4}), \\ -1, & \text{if } \bar{x}_1 \in [s_{sw,4}, s_{sw,5}), \\ -\frac{1}{2c_g}, & \text{if } \bar{x}_1 \in [s_{sw,5}, s_2), \end{cases}$$

where with $\bar{x}_2(t_4) = \bar{x}_2(t_6)$ we obtain the appropriate switching spots $s_{sw,1}$ to $s_{sw,5}$. Thus, for n unexplored intervals in $[0, 1]$, it can be shown that the optimal control consists of a string of n controls of the form (33) with appropriate switching conditions.

References

[1] S. Thrun, W. Burgard, D. Fox., Probabilistic Robotics, MIT Press, 2008.

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

[3] F. Clarke, Discontinuous feedback and nonlinear systems, in: Proc. of IFAC Symp. on Nonlinear Control Systems (NOLCOS), 2010.

[4] D. Feng, B. H. Krogh, Acceleration-constrained time-optimal control in n dimensions, IEEE Trans. on Automatic Control 31 (10) (1986) 955–958.

[5] I. Mitchell, A. Bayen, C. Tomlin, A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games, IEEE Trans. on Automatic Control 50 (7) (2005) 947–957.

[6] P. Grieder, M. Kvasnica, M. Baotić, M. Morari, Stabilizing low complexity feedback control of constrained piecewise affine systems, Automatica 41 (10) (2005) 1683 – 1694.

[7] D. Applegate, R. Bixby, V. Chvátal, W. Cook, The Traveling Salesman Problem: A Computational Study: A Computational Study, Princeton Series in Applied Mathematics, Princeton University Press, 2011.

[8] P. Toth, D. Vigo, Vehicle routing : problems, methods, and applications, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2015.

[9] C. Seatzu, D. Corona, A. Giua, A. Bemporad, Optimal control of continuous-time switched affine systems, IEEE Trans. on Automatic Control 51 (2006) 726–741.

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

[11] B. Passenberg, 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.

[12] C. G. Cassandras, J. Lygeros (Eds.), Stochastic hybrid systems, CRC Press, 2010.

[13] R. Bellman, An optimal search problem, SIAM Review 5.

[14] L. Stone, Theory of Optimal Search, Mathematics in Science and Engineering, Elsevier Science, 1976.

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

[16] S. Alpern, S. Gal, The Theory of Search Games and Rendezvous, Kluwer's Int. Ser. in Oper. Research & Management Science, 2003.

[17] M. Kao, J. Reif, S. Tate, Searching in an unknown environment: an optimal randomized algorithm for the cow-path problem, Information and Computation 131 (1) (1996) 63–80.

[18] J. Cortes, S. Martinez, T. Karatas, F. Bullo, Coverage control for mobile sensing networks, Robotics and Automation, IEEE Trans. on 20 (2) (2004) 243–255.

[19] M. Zhong, C. G. Cassandras, Distributed coverage control and data collection with mobile sensor networks, IEEE Trans. on Automatic Control 56 (10) (2011) 2445–2455.

[20] J. Le Ny, G. Pappas, On trajectory optimization for active sensing in gaussian process models, in: Proc. of 48th IEEE Conf. on Decision and Control (CDC) held jointly with 28th Chinese Control Conference (CCC), 2009, pp. 6286–6292.

[21] S. Smith, M. Schwager, D. Rus, Persistent robotic tasks: Monitoring and sweeping in changing environments, Robotics, IEEE Trans. on 28 (2) (2012) 410–426.

[22] C. Cassandras, X. Lin, X. Ding, An optimal control approach to the multi-agent persistent monitoring problem, IEEE Trans. on Automatic Control 58 (4) (2013) 947–961.

[23] A. Molin, S. Hirche, On the optimality of certainty equivalence for event-triggered control systems, IEEE Trans. on Automatic Control 58 (2013) 470–474.

[24] H. Axelsson, M. Boccadoro, M. Egerstedt, P. Valigi, Y. Wardi, Optimal mode-switching for hybrid systems with varying initial states, Nonlinear Analysis: Hybrid Systems 2 (3) (2008) 765–772.

[25] M. Rickert, A. Sieverling, O. Brock, Balancing exploration and exploitation in sampling-based motion planning, IEEE Trans. on Robotics 30 (6) (2014) 1305–1317.

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

[27] C. Amato, G. Konidaris, A. Anders, G. Cruz, J. P. How, L. P. Kaelbling, Policy search for multi-robot coordination under uncertainty, in: Proc. of Robotics: Science and Systems Conf. (RSS-15), 2015.

[28] M. Lahijanian, M. R. Maly, D. Fried, L. E. Kavraki, H. Kress-Gazit, M. Y. Vardi, Iterative temporal planning in uncertain environments with partial satisfaction guarantees, IEEE Trans. on Robotics 32 (3) (2016) 583–599.

[29] V. Raman, N. Piterman, C. Finucane, H. Kress-Gazit, Timing semantics for abstraction and execution of synthesized high-level robot control, IEEE Trans. on Robotics 31 (3) (2015) 591–604.

[30] K. S. Wesselowski, C. G. Cassandras, The elevator dispatching problem: Hybrid system modeling and receding horizon control, in: Proc. of 2nd IFAC Conf. on Analysis and Design of Hybrid Systems, 2006, pp. 136–141.

[31] Y. Khazaeni, C. G. Cassandras, Event-driven cooperative receding horizon control for multi-agent systems in uncertain environments, in: IEEE Trans. on Control of Network Systems, Vol. 5, 2018, pp. 409–422.

[32] V. Nenchev, C. Belta, Receding horizon robot control in uncertain environments with temporal logic constraints, in: 15th European Control Conf. (ECC'16), 2016, pp. 2614–2619.

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

[34] V. Nenchev, C. G. Cassandras, Optimal exploration and control for a robotic pick-up and delivery problem in two dimensions, in: Proc. of the 54th Conf. on Decision and Control (CDC'15), 2015, pp. 258–263.

[35] C. G. Cassandras, Y. Wardi, C. G. Panayiotou, C. Yao, Perturbation analysis and optimization of stochastic hybrid systems, European Journal of Control 16 (6) (2010) 642 – 661.

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

[37] T. Henzinger, The Theory of Hybrid Automata, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000, pp. 265–292.

[38] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, M. Diehl, Time-optimal path tracking for robots: A convex optimization approach, Automatic Control, IEEE Trans. on 54 (10) (2009) 2318–2327.

[39] E. A. Gol, M. Lazar, C. Belta, Language-guided controller synthesis for linear systems, IEEE Trans. on Automatic Control 59 (5) (2014) 1163–1176.

[40] D. P. Bertsekas, Constrained optimization and Lagrange multiplier methods, Athena Scientific, 1996.

[41] J. Nocedal, S. Wright, Numerical optimization, 2nd Edition, Springer-Verlag, 2006.

[42] M. Xu, J. Ye, A smoothing augmented lagrangian method for solving simple bilevel programs, Computational Optimization and Applications 59 (2014) 353–377.

[43] G. Bao, C. G. Cassandras, Stochastic comparison algorithm for continuous optimization with estimation, Journal of optimization theory and applications 91 (3) (1996) 585–615.

[44] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends in Machine Learning 3 (1) (2011) 1–122.

[45] H. M. Choset, Principles of Robot Motion: Theory, Algorithms, and Implementation., The MIT Press, 2005.