2015 European Control Conference (ECC)
July 15-17, 2015. Linz, Austria

Optimal motion planning with temporal logic and switching constraints

Vladislav Nenchev, Calin Belta, and Jörg Raisch

Abstract— **This paper proposes a method for automatic generation of time-optimal robot motion trajectories for the task of collecting and moving a finite number of objects to particular spots in space, while maintaining predefined temporal logic constraints. The continuous robot dynamics change upon an object pick-up or drop-off. The temporal constraints are expressed as syntactically co-safe Linear Temporal Logic (scLTL) formulas over the set of object and drop-off sites. We propose an approach based on constructing a discrete abstraction of the hybrid system modeling the robot in the form of a finite weighted transition system. Then, by employing tools from automata-based model checking, we obtain an automaton containing only paths that satisfy the specification. The shortest path in this automaton is found by graph search and corresponds directly to the time-optimal hybrid trajectory. The method is applied to a case study with a mobile ground robot and a case study involving a quadrotor moving in an environment with obstacles, thus reflecting its computational advantage over a direct optimization approach.**

I. Introduction

Motion and path planning are two of the most prominent problems in robotics. In particular, for the class of point-to-point navigation problems with obstacle avoidance, many efficient solutions have been proposed. They range from discretization approaches, which utilize graph search algorithms [1], continuous approaches based on, e.g., potential fields [2], and sampling-based methods such as rapidly-exploring random trees [3]. Recently, there has been an increasing interest in more advanced assignments that involve solving multiple tasks or visiting several locations in a complex environment. This problem is addressed by assuming an a-priori given decomposition into high-level planning and low-level control, thus focusing mainly on the purely discrete path planning task. High level specifications can be often captured effectively by temporal logic formulas that allow for employing model checking and automata game techniques to obtain a solution [4], [5], [6]. Finding the optimal path for a discrete transition system with a temporal logic specification has been studied, e.g. in [7]. The closely related Vehicle Routing Problem (VRP) [8], where the goal is to plan optimal routes for vehicles that have to service customer requests located at different spatial sites with temporal constraints, has also received extensive attention from several research communities. The VRP represents, in general, an instant of the well known NP-hard Traveling Salesperson Problem (TSP), for which many effective heuristics providing satisfactory solutions to moderately sized problems have been proposed [9]. VRPs combined with temporal logic specifications have been studied in [10], where the solution is acquired by solving a Mixed-Integer Linear Program (MILP) that incorporates the specification as additional optimization constraints, and in [11] by an automata-based approach. The hybrid Optimal Control Problem (OCP) of motion planning with continuous dynamics and a discrete specification has been addressed in, e.g., [12], [13]. An LTL specification can also be encoded directly into a single mixed-integer problem to obtain the optimal control for mixed logical dynamical [14] or differentially flat systems [15]. Hybrid optimal exploration and control problems with discrete high-level specifications [16], [17], [18] have also been investigated.

We present a method that generates the optimal control of a robot with switching double integrator dynamics for a pick-up and delivery task while maintaining a-priori given temporal logic and physical capacity constraints. More specifically, we want to find the minimum time trajectory satisfying a specification given as an scLTL formula, which is motivated by problems arising in logistics, vehicle routing and rescue assignments. By employing results from classical optimal control, we obtain a tailored finite representation of the robot's possible optimal motion in the environment in the form of a weighted transition system. As an scLTL formula can be represented by a Deterministic Finite Automaton (DFA), we obtain a finite product automaton that contains only satisfying runs of the system. Then, the optimal solution is found by discrete search over the product automaton.

Even though the abstraction relies on the assumption of double integrator dynamics, its applicability is by no means restricted to such systems only. There is a substantial number of vehicles that can be described by double integrators upon applying feedback linearization [19]. Further, the time-optimal solution for a double integrator is well known to provide a lower bound for the time to move along a straight path segment for many classes of systems and can thus be efficiently used to obtain an initial trajectory guess for more complex (nonlinear) dynamics [20]. The proposed approach allows parallelized computation, thus reducing the computation time as opposed to solving a single large optimization problem. Our solution guarantees optimality, correctness, and incorporates the robot's physical capacity constraint.

The remaining paper is organized as follows. In Section II,

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

C. Belta is with the Department of Mechanical Engineering and the Division of Systems Engineering, Boston University, MA, USA. cbelta@bu.edu

J. Raisch is with the Control Systems Group, Technische Universität Berlin and the Systems and Control Theory Group, Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany. raisch@control.tu-berlin.de

V. Nenchev was supported in part by the Fulbright Program and the German Academic Exchange Service (DAAD). C. Belta was partially supported by the NSF under grant CNS-1035588 and the ONR under grants N00014-10-10952 and N00014-09-1051.

TABLE I

ATOMIC PROPOSITIONS FOR THE TASK.

π	Proposition
π_l	Pick up o_l and carry it
π_d	Drop off all currently carried objects

we state the addressed problem. A suitable model by a hybrid automaton and an OCP reformulation are presented in Section III, followed by our solution. Then, we provide two case studies of a mobile ground robot and a quadrotor, moving in an environment with obstacles, and a comparison of the computational effort of our approach with related methods in Section V. Finally, we briefly summarize the results and discuss possible future work in Section VI.

II. PROBLEM FORMULATION AND APPROACH

In this section we introduce the OCP for a mobile robot that has to pick-up, move and drop-off objects with temporal logic constraints in a bounded environment. For a set S, let $|S|$ and 2^S denote the cardinality of S and the set of all subsets (power set) of S, respectively. A word ω over S is a finite sequence of elements from S, e.g., $\omega = s_1 \dots s_n$, such that $s_i \in S, \forall i \in \{1, \dots, n\}$. Consider an environment limited to the compact space $Y \subset \mathbb{R}^2$, containing L movable objects, uniquely labeled by the set $O = \{o_1, \dots, o_L\}$. An object $o_l, l \in \{1, \dots, L\}$ is characterized by its mass $m_l \in \mathbb{R}_+$ and position $y_l \in Y$. Let d denote a depot (drop-off location) with $y_d \in Y$. We assume that no two objects can be located at the same spot, except for at the depot.

Remark 1. *For simplicity, the absence of obstacles in the environment is assumed for the problem formulation. However, a possible solution for this case will be outlined in a follow-up remark and a case study will be provided.*

Consider a point-like mobile robot with dynamics

$$\dot{x}(t)=f(x,u,\mu)=\begin{bmatrix} 0_{2,2} & I_2 \\ 0_{2,2} & 0_{2,2} \end{bmatrix} x(t)+\frac{1}{\mu(t)}\begin{bmatrix} 0_{2,2} \\ I_2 \end{bmatrix} u(t), \tag{1}$$

where $x = [y' \; v']' \in \mathbb{R}^4$ with y and v standing for the position and the velocity, respectively, $0_{2,2}$ denotes a zero 2×2 matrix, I_2 an identity matrix with dimension 2 and the piecewise continuous input $u : [0, T] \to \mathbb{R}^2$ is limited by

$$\|u(t)\| \leq u_{\max}, \forall t \in [0, T], \tag{2}$$

where T stands for the final time and $u_{\max} \in \mathbb{R}_+$ is the maximum norm of the input. The function $\mu : [0, T] \to [m_\emptyset, m_{\max}] \subset \mathbb{R}_+$ denotes the overall mass of the robot that changes over time due to object pick-up and drop-off and is restricted to the interval between the nominal value $m_\emptyset$ and the maximum weight capacity constraint $m_{\max}$. Let the robot start at an initial position $y_0 \in Y$ with zero velocity. Assume that the robot has to satisfy a task specification captured by an scLTL formula [21] over the set of atomic propositions $\Pi = \{\pi_1, \dots, \pi_L, \pi_d\}$ denoting statements, summarized in Table I, which can be true or false at any time. Atomic propositions can be combined in a formula by boolean operators $\neg$ (negation), $\vee$ (disjunction) and $\wedge$ (conjunction), and temporal operators: $\mathcal{X}$ (next), $\mathcal{U}$ (until) and $\mathcal{F}$ (eventually). In general, for any scLTL formula ϕ over a set Π we can obtain a DFA with input alphabet 2^Π accepting all finite words that satisfy ϕ. Intuitively, $\mathcal{X}\pi$ states that $\pi \in \Pi$ becomes true in the next position of a word; $\pi_1 \mathcal{U} \pi_2$ expresses that $\pi_1 \in \Pi$ is true until $\pi_2 \in \Pi$ becomes true in a word; and $\mathcal{F}\pi$ requires that $\pi \in \Pi$ becomes true at some position in a word. As the atomic propositions (and therefore the specification) are defined over object pick-ups and drop-offs, we assume that the robot moving in the environment generates a word over Π, extended by an additional symbol only upon a pick-up or drop-off. The addressed OCP reads as follows.

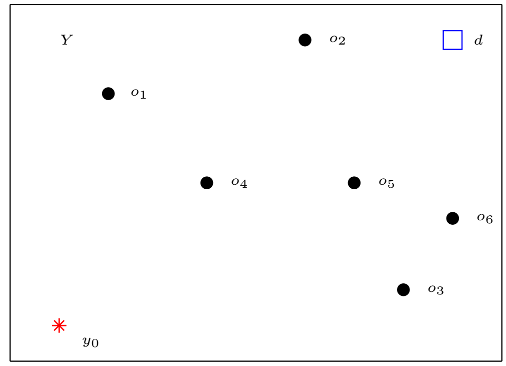

Fig. 1. Problem setup of the example. The starting point is marked by a star, objects are denoted by dots and the depot by a square.

Problem 1. *Given a robot with dynamics* (1) *with a control input limited by* (2) *deployed in an environment, and a specification as an scLTL formula ϕ over Π, find a time-optimal control trajectory $u|^*_{[0,T]}$ that steers the robot between its initial position, the object's sites and the depot, and satisfies ϕ and the maximum capacity constraint m_{max}.*

Example. *Let an environment $Y = [0, 5]^2$ contain the objects $O = \{o_1, \dots, o_6\}$ located at $y_1 = [1 \; 3.75]'$, $y_2 = [3 \; 4.5]'$, $y_3 = [4 \; 1]'$, $y_4 = [2 \; 2.5]'$, $y_5 = [3.5 \; 2.5]'$, $y_6 = [4.5 \; 2]'$, respectively, with masses $m_l = 1\,kg, \forall l \in \{1, \dots, 6\}$ and, a depot d at $y_d = [4.5 \; 4.5]'$. Consider a mobile ground robot starting at $x_0 = [0.5 \; 0.5 \; 0 \; 0]'$ with dynamics* (1), *control input limited by* (2) *with $u_{max} = 1\,N$, nominal mass $m_\emptyset = 3\,kg$ and maximum weight capacity $m_{max} = 5\,kg$. The setup is depicted in Figure 1. Assume an assignment where the robot starts neither at the depot nor at an object location and has to first pick-up object o_1, then either o_2 or o_4, then o_5 followed by o_6 or o_3 followed by o_5 and then end at d.*[1] *With $\gamma_1 = \neg\pi_1 \wedge \ldots \wedge \neg\pi_6 \wedge \neg\pi_d$, this specification is represented by the scLTL formula:*

$$\begin{aligned}\phi_1 :=& \gamma_1 \mathcal{U}(\pi_1 \wedge ((\pi_1 \vee \pi_d)\mathcal{U}((\pi_2 \vee \pi_4)\mathcal{U}((\pi_2 \vee \pi_4 \vee \pi_d) \\ & \mathcal{U}(((\pi_5 \wedge \mathcal{X}\pi_6) \vee (\pi_3 \wedge \mathcal{X}\pi_5)) \wedge \mathcal{X}\pi_d))))).\end{aligned} \tag{3}$$

Note that ϕ_1 allows intermediate visits of d, so that the objects, currently carried by the robot, can be dropped off.

[1]This specification may represent a scenario in which an assembly process at d requires specific parts located at different places in a particular order.

Our solution relies on the practical assumptions that the robot needs to stop moving whenever it picks up or drops off objects and that a pick-up and drop-off leads to an instantaneous change in the current mass $\mu(t)$. This allows for a representation of the robot moving in the environment by a hybrid automaton. By employing results from classical optimal control, we propose a tailored finite abstraction for the hybrid automaton under low-level control in the form of a weighted transition system. Automata-based model checking is performed by a parallel composition of the abstraction and the DFA corresponding to the scLTL formula. Then, the time-optimal path is found by graph search in the product automaton. The optimal control trajectory is generated by re-translating the optimal path into the hybrid automaton.

III. Hybrid model and problem reformulation

As L is finite and both object pick-ups and drop-offs are instantaneous, the dynamics (1) can be represented by a finite number of discrete states $q \in Q$ that denote the robot carrying a subset of objects $O_q \subseteq O$ with an overall mass $m_q = m_\emptyset + \sum_{l, o_l \in O_q} m_l$. Hence, $Q := \{q | m_q \leq m_{\max}\}$ and the codomain of $\mu(t)$ is the finite set $M_q := \{m_q | q \in Q\}$. Then, the overall motion of the robot can be described by a hybrid automaton, i.e., a 10-tuple $H = (S, U, F, E, \mathrm{Inv}, G, R, \Pi, \Lambda, S_0)$:

- $S = Q \times X$ – the hybrid state space of the system with Q representing the finite set of discrete states and $X \subset \mathbb{R}^4$ the continuous state space;
- $U = \{\theta \in \mathbb{R}^2 | \|\theta\| \leq u_{\max}\}$ – the continuous input space;
- $F = \{f_q\}_{q \in Q}$ – the collection of vector fields $f_q : X \times U \to X$ describing the dynamics (1) with $\mu(t) = m_q$;
- $E \subseteq Q \times Q = E_1 \cup E_2$ – the discrete state transitions, where E_1 are transitions corresponding to the pick-up of a single object, i.e.,
$$(q_i, q_j) \in E_1, \text{iff } \exists o_l \in O \setminus O_{q_i}, \text{s.t. } O_{q_j} = O_{q_i} \cup \{o_l\},$$
and E_2 contains transitions corresponding to the drop-off of all objects currently carried by the robot, i.e.,
$$(q_i, q_j) \in E_2, \text{iff } O_{q_i} \neq \emptyset \wedge O_{q_j} = \emptyset;$$
- $\mathrm{Inv} : Q \to 2^X$ – the invariant map with
$$\mathrm{Inv}(q) = \begin{cases} X \setminus \{[y_l{}' \ 0_2']', [y_d{}' \ 0_2']'\}, & \text{if } O_q \neq \emptyset, \\ X \setminus \{[y_l{}' \ 0_2']'\}, & \text{if } O_q = \emptyset; \end{cases}$$
- $G : E \to 2^X$ – the guard map with
$$G(e) = \begin{cases} \{[y_l{}' \ 0_2']'\}, & \text{if } e \in E_1, \\ \{[y_d{}' \ 0_2']'\}, & \text{else;} \end{cases}$$
- $R : E \times X \to X$ – the reset map with $R(e, x) = x, \forall (e, x) \in E \times G(e)$;
- Π – the set of atomic propositions;
- $\Lambda \subseteq E \times \Pi$ – the labeling relations giving the atomic propositions $\pi \in \Pi$ satisfied at a discrete state transition $e \in E$ with $(e, \pi) \in \Lambda$, iff $(G(e) = \{[y_l' \ 0_2']'\} \wedge \pi = \pi_l) \vee$ $(G(e) = \{[y_d' \ 0_2']'\} \wedge \pi = \pi_d)$, i.e., π_l or π_d is true at any transition corresponding to the pick-up of object o_l or the drop-off of objects at d, respectively;
- $S_0 = \{(q(0), x(0))\} = \{(q_0, [y_0{}' \ 0_2']')\}$ – the initial state set, where $O_{q_0} = \emptyset$.

Let $L_s \in \mathbb{N}$ denote the finite number of stages $[t_i, t_{i+1}), i \in \{0, \ldots, L_s - 1\}$, closed from the left and open from the right, where t_i are the time instants of discrete state switchings and $t_{L_s} = T$. Then, a sequence of piecewise continuous input trajectories $u_H = (u|_{[t_0, t_1)}, \ldots, u|_{[t_{L_s-1}, t_{L_s})})$ yields a trajectory of the hybrid system $\tau_H = (t, q, x)$, where $t = (t_0, \ldots, t_{L_s})$ is a sequence of strictly increasing initial, switching and final times; $q = (q_0, \ldots, q_{L_s - 1})$ is a sequence of discrete states; and $x|_{[0,T]} = (x|_{[t_0, t_1)}, \ldots, x|_{[t_{L_s-1}, t_{L_s})})$ is a sequence of absolutely continuous state trajectories. The cost is given by the overall time

$$J(\tau_H, u_H) := \int_0^T dt = \sum_{i=0}^{L_s - 1} (t_{i+1} - t_i). \tag{4}$$

As H is deterministic, Problem 1 can be restated as follows.

Problem 2. *Find a control trajectory $u_H|_{[0,T]}^*$ for the hybrid system H that minimizes* (4) *subject to restrictions imposed by the specification ϕ.*

IV. Solution

The solution of Problem 2 consists of a finite sequence of piecewise continuous controls enforcing the time-optimal motion of the robot from a location y_i to another location y_j, where $y_i, y_j \in \mathcal{Y}$, $\mathcal{Y} = (\cup_{l \in \{1, \ldots, L\}} \{y_l\}) \cup \{y_d\} \cup \{y_0\}$ with zero velocity. As the initial and final states for each stage are fixed, the corresponding two-point boundary value problems (TPBVP) can be decoupled. By applying these controls, the hybrid automaton evolves between certain hybrid states, which can be captured by a finite discrete abstraction.

A. Discrete abstraction

Consider first the problem of steering the robot in a time-optimal manner from a state $s_i = (q_{k_i}, [y_{r_i}' \ 0_2']')$ to another state $s_j = (q_{k_j}, [y_{r_j}' \ 0_2']')$, $s_i, s_j \in S$. Due to the assumed absence of obstacles in Y, the time-optimal motion of the robot with dynamics (1) is on straight lines. Define an affine transformation $\xi = T_{ij}(y - y_i)$, where the 1×2 matrix T_{ij} can be chosen such that $T_{ij}(y_j - y_i) = 1$. This provides a reduced motion model in the 2-dimensional state space $\tilde{x} = [\xi \ \dot{\xi}]'$ with the scalar input $\tilde{u} := (T_{ij}/m_{q_i})u$, leading to the partial OCP:

$$\min_{\tilde{u}} \ t_f^{ij},$$
$$\text{s.t. } \dot{\tilde{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \tilde{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tilde{u}(t), u_{\max}^{ij} = \frac{\|T_{ij}\|}{m_{q_i}},$$
$$\tilde{x}(0) = [0 \ 0]', \tilde{x}(t_f^{ij}) = [1 \ 0]', |\tilde{u}(t)| \leq u_{\max}^{ij}.$$

This OCP corresponds to the classical linear time-OCP [22] and its solution is a piecewise constant controller taking values in the set $\{-u_{\max}^{ij}, u_{\max}^{ij}\}$. Thus, any transition from the hybrid state s_i to the hybrid state s_j has minimum duration

$$t_f^{ij*} = 2\sqrt{(m_{q_i}/u_{\max})\|y_j - y_i\|}. \tag{5}$$

Note that the transition time is independent of the final discrete state q_j. The optimal control input u_{ij}^* can be obtained by a re-translation to the hybrid automaton.

Remark 2. *If the control input for* (1) *is constrained by* $|u_1| \leq u_{1,max}, |u_2| \leq u_{2,max}$ *(which is often the case when applying feedback linearization), the optimal transition time is given by the maximum of the individual minimum times for each spatial direction. The optimal control for a transition can be acquired approximately, e.g., as described in [23].*

Remark 3. *Under the presence of obstacles or for more complex discrete-time linear vehicle dynamics, the corresponding partial OCPs can be solved approximately by formulating MILPs for a finite polyhedral under-approximation of* (2) *and a polyhedral over-approximation of the obstacles (see [24]). For nonlinear dynamics, approximate costs can also be obtained, e.g., for quadrotors by solving nonlinear TPBVPs for each transition [25], or by neglecting the yaw as an additional degree of freedom and linearizing the dynamics around the hovering point [26]. A corresponding case study is provided in Section V.*

Based on the above elaborations, the time-optimal motion of the robot between all hybrid states $s_i, s_j \in S$, relevant for solving the task, are represented by a finite transition system (TS) $\mathcal{T} = (S^H_\mathcal{T}, \Delta^H_\mathcal{T}, \Pi, \Lambda^H_\mathcal{T}, W^H_\mathcal{T}, S^H_{0,\mathcal{T}})$, where

- $S^H_\mathcal{T} = Q \times \mathcal{Y}$ is the finite set of states;
- $\Delta^H_\mathcal{T} \subseteq S^H_\mathcal{T} \times S^H_\mathcal{T}$ – the set of transitions, where for $s_i = (q_{k_i}, y_{r_i}), s_j = (q_{k_j}, y_{r_j}) \in S^H_\mathcal{T}$, $(s_i, s_j) \in \Delta^H_\mathcal{T}$, iff $G((q_{k_i}, q_{k_j})) = \{[y'_{r_j}\ \ 0'_2]'\}$;
- Π – the finite set of atomic propositions;
- $\Lambda^H_\mathcal{T} : S^H_\mathcal{T} \to \Pi$ – the labeling function that associates a state $s_j = (q_{k_j}, y_{r_j}) \in S^H_\mathcal{T}$ with an atomic proposition $\pi_{r_j} \in \Pi$, iff $((q_{k_i}, q_{k_j}), \pi_{r_j}) \in \Lambda$;
- $W^H_\mathcal{T} : \Delta^H_\mathcal{T} \to \mathbb{R}_+$ – the weighting function, where $\forall (s_i, s_j) \in \Delta^H_\mathcal{T}, W^H_\mathcal{T}((s_i, s_j)) = t^{ij*}_f$, as given in (5);
- $S^H_{0,\mathcal{T}} = \{s_0 = (q_0, y_0)\}$ – the initial state set.

A finite run (discrete trajectory) of $\mathcal{T}$ is given by a state sequence $\tau^H_\mathcal{T} = s_{i_0} s_{i_1} \ldots s_{i_N}$, where $s_{i_0} = s_0$ and $s_{i_n}, s_{i_{n+1}} \in S^H_\mathcal{T}, (s_{i_n}, s_{i_{n+1}}) \in \Delta^H_\mathcal{T}, \forall n \in \{0, N-1\}$ with cost

$$J(\tau^H_\mathcal{T}) := \sum_{n=0}^{N-1} W^H_\mathcal{T}((s_{i_n}, s_{i_{n+1}})).$$

The finite word produced by $\tau^H_\mathcal{T}$ is $\omega(\tau^H_\mathcal{T}) = \pi_{i_0} \ldots \pi_{i_N}$, $\forall n \in \{0, \ldots, N-1\}, \Lambda^H_\mathcal{T}(s_{i_n}) = \pi_{i_n}$. We now want to find the optimal run in $\mathcal{T}$ that satisfies the given specification.

B. Optimal control synthesis

Let $\mathcal{A}_\phi = (S_\phi, \Sigma, \Delta_\phi, S_{0,\phi}, S_{m,\phi})$ be a DFA corresponding to the scLTL formula ϕ, where S_ϕ is the finite set of states, $\Sigma = 2^\Pi$ the input alphabet, $\Delta_\phi : S_\phi \times \Sigma \to S_\phi$ the transition mapping, $S_{0,\phi}$ the set of initial states and, $S_{m,\phi} \subset S_\phi$ the set of marked or accepting states. Following the idea of automata-based model checking, we construct a parallel composition of the weighted TS $\mathcal{T} = (S^H_\mathcal{T}, \Delta^H_\mathcal{T}, \Pi, \Lambda^H_\mathcal{T}, W^H_\mathcal{T}, S^H_{0,\mathcal{T}})$ and $\mathcal{A}_\phi$ corresponding to the formula ϕ, leading to a weighted finite product automaton $\mathcal{A}^\mathcal{P} = \mathcal{T} \times \mathcal{A}_\phi = (S_\mathcal{P}, \Delta_\mathcal{P}, W_\mathcal{P}, S_{\mathcal{P},0}, S_{\mathcal{P},m})$ with

- $S_\mathcal{P} \subseteq S^H_\mathcal{T} \times S_\phi$ – the finite set of reachable states;
- $\Delta_\mathcal{P} \subseteq S_\mathcal{P} \times S_\mathcal{P}$ – the set of transitions, where $((s_\mathcal{T}, s_\phi), (s'_\mathcal{T}, s'_\phi)) \in \Delta_\mathcal{P}$, iff $(s_\mathcal{T}, s'_\mathcal{T}) \in \Delta^H_\mathcal{T} \wedge (s_\phi, \{\Lambda^H_\mathcal{T}(s'_\mathcal{T})\}, s'_\phi) \in \Delta_\phi$;
- $W_\mathcal{P} : \Delta_\mathcal{P} \to \mathbb{R}_+$ – the weighting function, such that $W_\mathcal{P}(((s_\mathcal{T}, s_\phi), (s'_\mathcal{T}, s'_\phi))) = W^H_\mathcal{T}((s_\mathcal{T}, s'_\mathcal{T}))$, $\forall ((s_\mathcal{T}, s_\phi), (s'_\mathcal{T}, s'_\phi)) \in \Delta_\mathcal{P}$;
- $S_{\mathcal{P},0} = S^H_{0,\mathcal{T}} \times S_{\phi,0}$ – the initial state set;
- $S_{\mathcal{P},m} \subseteq S^H_\mathcal{T} \times S_{\phi,m}$ – the set of accepting (final) states.

An accepting run $\tau_\mathcal{P} = s_{\mathcal{P},0} \ldots s_{\mathcal{P},N} = (s_{i_0}, s_{\phi,0}) \ldots (s_{i_N}, s_{\phi,N})$, $(s_{i_0}, s_{\phi,0}) \in S_{\mathcal{P},0}, (s_{i_N}, s_{\phi,N}) \in S_{\mathcal{P},m}$ of $\mathcal{A}^\mathcal{P}$ corresponds to an accepting run of $\mathcal{A}_\phi$ over the input word $\omega(\tau_\mathcal{P}) = \{\pi_{i_0}\} \ldots \{\pi_{i_N}\}$. The following theorem reflects the role of $\mathcal{A}^\mathcal{P}$ for the OCP.

Proposition 1. *The shortest accepting run* $\tau^*_\mathcal{P}$ *of* $\mathcal{A}^\mathcal{P}$ *corresponds to a finite run* $\tau^H_\mathcal{T}$ *of* $\mathcal{T}$ *that can be retranslated to the control trajectory* $u_H|^*_{[0,T]}$ *solving Problem 1.*

The proof follows directly from the properties of the discrete abstraction $\mathcal{T}$ (states denote all relevant discrete evolutions of the system, weights denote minimum transition times) and the product automaton $\mathcal{A}^\mathcal{P}$ (capturing all paths that satisfy the specification). $\mathcal{A}^\mathcal{P}$ can be seen as a weighted finite graph, where $S_\mathcal{P}$ is the node set and $\Delta_\mathcal{P}$ is the set of edges with weights $W_\mathcal{P}$. Clearly, the cost of a path in $\mathcal{A}^\mathcal{P}$ is given by the sum of the weights of its transitions. Finding the shortest path from the initial state to a marked state can be done by employing Dijkstra's algorithm. The correctness of the proposed scheme follows directly from Proposition 1 and the properties of Dijkstra's algorithm. For convenience, we summarize the proposed solution in Algorithm 1.

Algorithm 1 Solution of Problem 1

Input: Finite set of objects $O = \{o_1, \ldots, o_L\}$ with masses m_l and locations y_l, $l \in \{1, \ldots, L\}$, a depot d at y_d; robot dynamics $\dot{x}(t) = f(x, u, \mu)$, $\|u(t)\| \leq u_{\max}$, nominal mass $m_\emptyset$, maximum capacity constraint $m_{\max}$, $\mu : [0, T] \to [m_\emptyset, m_{\max}]$ and initial position y_0; an scLTL formula ϕ over $\Pi = \{\pi_1, \ldots, \pi_L, \pi_d\}$
Output: The optimal control $u|^*_{[0,T]}$

```
procedure Initialization(Input)
    Construct hybrid automaton H = (S, U, F, E, Inv, G, R, Π, Λ, S_0)
    Construct finite abstraction of H under time-optimal low-level control by TS 𝒯
    Construct DFA 𝒜_φ corresponding to φ
end procedure
Compute product automaton 𝒜^𝒫 = 𝒯 × 𝒜_φ
run:= shortest_path(𝒜^𝒫)
control:= control_translate_to_H(run)
return control
```

C. Complexity

Let $|S^H_\mathcal{T}|$ denote the size of the finite TS $\mathcal{T}$, which grows with the number of objects $|O|$ und the number of locations $|\mathcal{Y}|$, and $|S_\phi|$ the size of $\mathcal{A}_\phi$, which is exponential in the length of ϕ, i.e., $|\phi|$. Then, $|S_\mathcal{P}| \leq |S^H_\mathcal{T}|.|S_\phi| \leq |\mathcal{Y}|.2^{|O|+|\phi|}$ holds for size of the product automaton. However, depending on the maximum capacity constraint, the masses of the

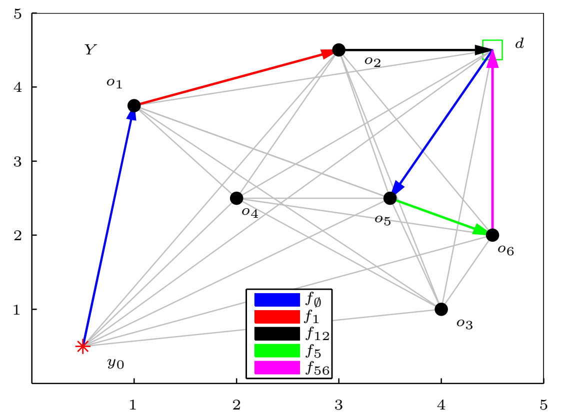

Fig. 2. Optimal solution with $J^* = 35.78\,\mathrm{s}$ for the example. The trajectory parts where the robot moves with different dynamics are denoted by different colors. Gray paths represent possible transitions in the environment.

objects and the specification, the size of the automaton will typically be much smaller than this upper bound. The time complexity of Dijkstra's algorithm is $\mathcal{O}(|S_{\mathcal{P}}| \log |S_{\mathcal{P}}|)$.

V. Implementation and Case Studies

In this section we apply the proposed method to two case studies. Algorithm 1 was implemented as follows. The DFA $\mathcal{A}_\phi$ corresponding to the scLTL specification ϕ was obtained with *scheck* [27]. The computation of the finite abstracted TS $\mathcal{T}$, the product automaton $\mathcal{A}^{\mathcal{P}} = \mathcal{T} \times \mathcal{A}_\phi$ and Dijkstra's algorithm were implemented in MATLAB. All computations were performed on an Intel® Core™ i7 2.20 GHz processor with 8 GB RAM. Obstacle avoidance and more complex vehicle dynamics are incorporated into the low-level OCPs (Remark 2 and 3) by formulating corresponding MILPs for a discrete-time approximation of the dynamics with sampling time $0.1\,\mathrm{s}$ over a finite horizon $N_{\max} \in \mathbb{N}$. All MILPs are solved by directly interfacing the solver Gurobi.

A. Mobile ground robot

Recall the example from Section II. The DFA with size $|S_\phi| = 17$ states, corresponding to the scLTL specification ϕ_1 in (3), was computed in less than 1 s. By employing Algorithm 1 we obtain the optimal path depicted in Figure 2. The robot first collects objects o_1 and o_2, then performs an intermediate drop-off, and then collects o_5 followed by o_6 and, finally, ends at the depot. This behavior was automatically produced by our approach in accordance with the specification and the capacity constraint.

B. Quadrotor with obstacles

Consider an environment $Y = [0,5]^3\,\mathrm{m}$ containing the objects $O = \{o_1, o_2, o_3\}$, where o_1 is initially located at $[1\ 3.75\ 5]'$, o_2 at $[4\ 1\ 3]'$ and o_3 at $[3\ 4.5\ 2]'$, with masses $m_i = 0.5\,\mathrm{kg}$, $\forall i \in \{1,2,3\}$, and a depot d at $y_d = [5\ 5\ 0.5]'$. Let a set of polyhedral obstacles $\mathrm{Obs} = \{\mathrm{obs}_1, \ldots, \mathrm{obs}_6\}$ be given with

$$\mathrm{obs}_j = \left\{ \xi \in \mathbb{R}^3 \,\middle|\, \begin{bmatrix} I_3 \\ -I_3 \end{bmatrix} \xi \leq \begin{bmatrix} I_3 \\ -I_3 \end{bmatrix} \kappa_j + \begin{bmatrix} I_3 \\ I_3 \end{bmatrix} \epsilon_j \right\}, \quad (6)$$

$\forall j \in \{1, \ldots, 6\}$, where $\kappa_j \in Y$ is the center and $\epsilon_j \in \mathbb{R}^3_+$ a size vector, summarized in Table II. A quadrotor has to solve

TABLE II

Obstacle parameters in the quadrotor case study.

	obs_1	obs_2	obs_3	obs_4	obs_5	obs_6
κ_j	$\begin{bmatrix}1\\3.75\\2.5\end{bmatrix}$	$\begin{bmatrix}4\\1\\1.5\end{bmatrix}$	$\begin{bmatrix}3\\4.5\\1\end{bmatrix}$	$\begin{bmatrix}5\\5\\0.25\end{bmatrix}$	$\begin{bmatrix}0.25\\3.75\\1\end{bmatrix}$	$\begin{bmatrix}1\\1.5\\0.5\end{bmatrix}$
ϵ_j	$\begin{bmatrix}0.5\\0.75\\2.5\end{bmatrix}$	$\begin{bmatrix}0.5\\1\\1.5\end{bmatrix}$	$\begin{bmatrix}1\\0.5\\1\end{bmatrix}$	$\begin{bmatrix}0.25\\0.25\\0.25\end{bmatrix}$	$\begin{bmatrix}0.25\\0.75\\1\end{bmatrix}$	$\begin{bmatrix}1\\1\\0.5\end{bmatrix}$

an optimal pick-up and delivery assignment, where object o_1 is picked up first, and then o_2 and o_3 in an arbitrary order, while allowing a drop-off after every pick-up, corresponding to the scLTL formula

$$\phi_2 := \gamma_2 \mathcal{U}(\pi_1 \wedge ((\pi_1 \vee \pi_d)\mathcal{U}((\pi_2 \wedge ((\pi_2 \vee \pi_d)\mathcal{U} (\pi_3 \wedge \mathcal{X}\pi_d))) \vee (\pi_3 \wedge ((\pi_3 \vee \pi_d)\mathcal{U}(\pi_2 \wedge \mathcal{X}\pi_d))))))$$

with $\gamma_2 = \neg\pi_1 \wedge \neg\pi_2 \wedge \neg\pi_3 \wedge \neg\pi_d$. The size of the corresponding DFA is $|S_\phi| = 17$ states, computed in less than 1 s. A practical application of this case study may be an automated package delivery and pick-up system, resembling the recently announced plans of major international online retailers and logistics companies.

A quadrotor can be represented by a nonlinear model with state $\tilde{x} = (y', v', \tilde{r}', \tilde{w}')'$, where $y \in Y$ is the position, $v \in [-5,5]^3$ m/s the velocity, $\tilde{r} \in \mathbb{R}^3$ the orientation (rotation about an axis by an angle $\|\tilde{r}\|$), $\tilde{w} \in \mathbb{R}^3$ the angular velocity [28]. In our case study, the dynamics are linearized around the hovering state of the quadrotor and, as the yaw is an additional degree of freedom, it is set to zero. Thus, we use a reduced model with state $x = (y', v', r', w')'$, where $r \in [-1,1]^2$ rad and $w \in [-5,5]^2$ rad/s, yielding $\dot{x}(t) = f_q(x,u) = Ax(t) + B_{q_i}u(t)$,

$$A = \begin{bmatrix} 0_{3,3} & I_3 & 0_{3,2} & 0_{3,2} \\ 0_{3,3} & 0_{3,3} & \begin{bmatrix} 0 & g \\ -g & 0 \\ 0 & 0 \end{bmatrix} & 0_{3,2} \\ 0_{2,3} & 0_{2,3} & 0_{2,2} & I_2 \\ 0_{2,3} & 0_{2,3} & 0_{2,2} & 0_{2,2} \end{bmatrix}, B_q = \begin{bmatrix} 0_{3,2} & 0_{3,1} \\ \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m_{q_i}} \end{bmatrix} & 0_{3,1} \\ 0_{2,2} & 0_{2,2} \\ 0_{2,2} & \frac{l_a}{\mathcal{I}_{q_i}} I_2 \end{bmatrix}$$

where $g = 9.81\,\mathrm{m/s^2}$, $l_a = 0.15\,\mathrm{m}$ is the distance from the center of the vehicle to each rotor, $m_\emptyset = 1\,\mathrm{kg}$ and m_{q_i} are the masses of the unloaded and loaded quadrotor, and $\mathcal{I}_\emptyset = 5.10^{-3}\ \mathrm{kg\,m^2}$ and $\mathcal{I}_{q_i}$ are the moments of inertia of the unloaded and loaded quadrotor, respectively. The moment of inertia changes instantaneously by adding or subtracting $2.10^{-2}\ \mathrm{kg\,m^2}$ from its current value at an object pick-up or drop-off, respectively. The control input is $u = (u_f, u_{y_1}, u_{y_2})' \in \mathbb{R}^3$, where $u_f \in [-4.55, 9.94]\,\mathrm{N}$ is the total thrust of the rotors relative to the thrust needed to keep the quadrotor in the hovering state, and $u_{y_1}, u_{y_2} \in [-3.6, 3.6]\,\mathrm{N}$ describe the relative thrust of the rotors producing roll and pitch motion, respectively. Applying the proposed method for the quadrotor starting at $x_0 = 0_{10}$, we obtain the approximately optimal solution (Figure 3), which satisfies ϕ_2 and the capacity constraint $m_{\max} = 3\,\mathrm{kg}$.

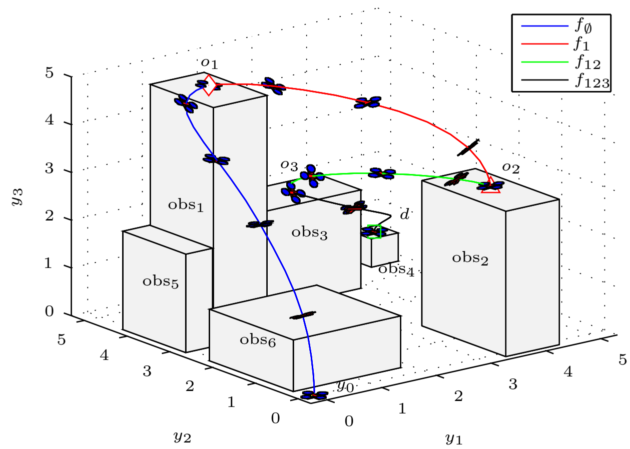

Fig. 3. Approximately optimal solution for a quadrotor with overall time $J^* = 7.5\,\mathrm{s}$ for the specification ϕ_2. The trajectory parts where the quadrotor moves with different dynamics are denoted by different colors.

TABLE III

COMPUTATION TIMES t_{COMP} (MEAN$\pm$STANDARD ERROR) OVER 50 RUNS.

Case study	two-level		direct	
	$N_{\max}$	t_{comp} [s]	$N_{\max}$	t_{comp} [s]
Ground robot	N/A	2.1	75	3 ± 1
	N/A	2.1	85	6 ± 2
Quadrotor	25	7 ± 1	80	11 ± 2
	30	8 ± 1	100	13 ± 3

C. Discussion

In related approaches, the LTL specification was directly encoded as linear constraints in a MILP that solves the complete OCP over a finite time horizon [14], [15]. In contrast, our method is based on decomposing the OCP into a finite set of continuous low-level and one discrete high-level OCP. Under the presence of obstacles, the low-level OCPs can be formulated as MILPs with a significantly lower number of discrete variables over a shorter time horizon compared to the direct optimization approach. Table III reflects the computational advantage of our two-level method over the alternative direct approach for different time horizons $N_{\max}$. Thus, our approach appears particularly promising for recomputation in dynamically changing environments.

VI. Conclusions

In this paper we presented a method for obtaining time-optimal control trajectories for mobile robots collecting a finite number of objects and moving them to particular spots in space while satisfying a finite temporal logic specification and a capacity constraint. The suggested procedure includes a discrete abstraction of the hybrid system under (approximately) time-optimal low-level control; after applying automata-based model checking techniques, the optimal high-level control synthesis was conducted by graph search. The method was applied in two case studies, involving a mobile ground robot and a quadrotor moving in an environment with obstacles, showing its computational advantage over a direct optimization approach. For future work, we plan to extend the method for the presence of uncertainty.

References

[1] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006, available at http://planning.cs.uiuc.edu/.

[2] E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential fields," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

[3] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Int. Journal of Robotics Research (IJRR)*, vol. 20, pp. 378–400, 1999.

[4] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic robots," *Automatica*, vol. 45, no. 2, pp. 343–352, 2009.

[5] E. Plaku, L. Kavraki, and M. Vardi, "Hybrid systems: From verification to falsification by combining motion planning and discrete search," *Formal Methods in Syst. Design*, vol. 34, pp. 157–182, 2009.

[6] T. Wongpiromsarn, U. Topcu, and R. M. Murray, "Receding horizon control for temporal logic specifications," in *Hybrid Systems: Computation and control (HSCC)*, 2010, pp. 101–110.

[7] S. L. Smith, J. Tumova, C. Belta, and D. Rus, "Optimal path planning for surveillance with temporal-logic constraints," *Int. Journal of Robotics Research (IJRR)*, vol. 30, no. 14, pp. 1695–1708, 2011.

[8] P. Toth and D. Vigo, *The Vehicle Routing Problem*. Philadelphia, PA, USA: SIAM, 2001.

[9] G. Laporte, "Fifty years of vehicle routing," *Transportation Science*, vol. 43, no. 4, pp. 408–416, 2009.

[10] S. Karaman and E. Frazzoli, "Vehicle routing with temporal logic specifications: Applications to multi-uav mission planning," *Journal of Robust and Nonlinear Control*, vol. 21, pp. 1372–1395, 2011.

[11] C. Vasile and C. Belta, "An automata-theoretic approach to the vehicle routing problem," in *Proc. of Robotics: Science and Systems*, 2014.

[12] A. Bhatia, L. Kavraki, and M. Vardi, "Motion planning with hybrid dynamics and temporal goals," in *IEEE CDC*, 2010.

[13] E. Gol, M. Lazar, and C. Belta, "Language-guided controller synthesis for discrete-time linear systems," in *HSCC*, 2012, pp. 95–104.

[14] S. Karaman, R. Sanfelice, and E. Frazzoli, "Optimal control of mixed logical dynamical systems with linear temporal logic specifications," in *IEEE Conf. on Decision and Control (CDC)*, 2008, pp. 2117–2122.

[15] E. M. Wolff, U. Topcu, and R. M. Murray, "Automaton-guided controller synthesis for nonlinear systems with temporal logic," in *IEEE/RSJ IROS*, 2013, pp. 4332–4339.

[16] M. Maly, M. Lahijanian, L. Kavraki, H. Kress-Gazit, and M. Vardi, "Iterative temporal motion planning for hybrid systems in partially unknown environments," in *HSCC*, 2013.

[17] 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, pp. 1236–1241.

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

[19] B. d'Andrea Novel, G. Bastin, and G. Campion, "Dynamic feedback linearization of nonholonomic wheeled mobile robots," in *IEEE Int. Conf on Robotics and Automation (ICRA)*, 1992, pp. 2527–2532.

[20] T. Kalmar-Nagy, R. D'Andrea, and P. Ganguly, "Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle," *Robotics and Autonomous Systems*, vol. 46, pp. 47–64, 2004.

[21] O. Kupferman and M. Vardi, "Model checking of safety properties," *Formal Methods in System Design*, vol. 19, no. 3, pp. 291–314, 2001.

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

[23] S. Pifko, A. Zorn, and M. West, "Geometric interpretation of adjoint equations in optimal low thrust space flight," in *Proc. of the AIAA/AAS Astrodynamics Specialist Conf.*, 2008.

[24] A. Richards and J. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *Proc. of the American Control Conf. (ACC)*, vol. 3, 2002, pp. 1936 – 1941 vol.3.

[25] M. Hehn, R. Ritz, and R. DAndrea, "Performance benchmarking of quadrotor systems using time-optimal control," *Autonomous Robots*, vol. 33, no. 1-2, pp. 69–88, 2012.

[26] D. J. Webb and J. van den Berg, "Kinodynamic rrt*: Asymptotically optimal motion planning for robots with linear dynamics," in *Proc. of IEEE Int.Conf. on Robotics and Automation (ICRA)*, 2013.

[27] T. Latvala, "Efficient model checking of safety properties," in *10th International SPIN Workshop*. Springer, 2003.

[28] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro-uav testbed," *Robotics Automation Magazine, IEEE*, vol. 17, no. 3, pp. 56–65, 2010.