Sampling-based Motion Planning with High-Level Discrete Specifications

Erion Plaku

***Abstract*— Motion planning has generally focused on computing a collision-free trajectory to a goal region. Enhancing the ability of robots in manipulation, automation, medicine, and other areas, however, often requires richer task specifications. Toward this goal, we study the problem of computing a collision-free trajectory that satisfies task specifications given by Finite Automata, STRIPS, Linear Temporal Logic, and other logic models. We propose to combine sampling-based motion planning with discrete planning. The search for a solution trajectory is conducted simultaneously over the continuous space of motions and the discrete space of the task specification. In this search, discrete plans guide motion planning as it extends a tree consisting of collision-free trajectories, while information gathered from motion planning is used to further improve the discrete plans. As a result of this interplay, the approach is able to selectively sample and explore those continuous regions and discrete plans that allow it to significantly advance the search for a trajectory that satisfies the task specification.**

I. INTRODUCTION

Motion planning has generally focused on the problem of computing collision-free trajectories to a goal region. Enhancing the ability of robots to act on their own requires, however, richer task specifications. This is true in applications ranging from manipulation and automation to robotic-assisted surgery and safety validation in hybrid systems. Such applications pose significant challenges as they require planning at multiple levels of discrete and continuous abstractions since these applications involve, on the one hand, an abstraction into discrete, logical steps, each of which may require, on the other hand, substantial continuous motion planning to carry out. Progress in this direction requires addressing the combined discrete and motion-planning problem*:

> Given a discrete specification Δ grounded in the physical world, plan the continuous motions so that the resulting robot trajectory is dynamically feasible, avoids collisions, and satisfies Δ.

Toward this goal, we propose to treat the combined problem of discrete and motion planning as a search problem over both the discrete space of actions and the continuous space of motions [10], [11]. Conceptually, the approach consists of a continuous and a discrete planning layer which work in tandem to effectively compute a trajectory that satisfies Δ.

II. DISCRETE SPECIFICATIONS

Drawing from AI, the approach allows for high-level discrete specifications to be given as Finite Automata, Linear Temporal Logic (LTL), and Planning Domain Definition Languages (e.g., STRIPS). These specifications are built in terms of propositions, predicates, Boolean and temporal connectives, and action schemas.

Propositions express general statements, e.g., "robot has grasped the object," "needle is in the target area." Predicates express relations among objects, e.g., $\textsc{on}(\textit{book}, \textit{table})$, $\textsc{InContact}(\textit{needle}, \textit{tissue})$. Propositions and predicates can be combined with Boolean connectives, such as $\neg(\textit{not})$, $\wedge(\textit{and})$, $\vee(\textit{or})$, or temporal connectives such as $\mathcal{X}(\textit{next})$, $\mathcal{U}(\textit{until})$, $\mathcal{F}(\textit{eventually})$, $\mathcal{G}(\textit{always})$. For example, the specification "the robotic car, after inspecting a contaminated area A should immediately go to the decontamination station B and then eventually go to one of the base stations C or D" can be expressed as an LTL formula $\mathcal{G}(\pi_A \rightarrow (\mathcal{X}(\pi_B) \wedge \mathcal{F}(\pi_C \vee \pi_D)))$, where $\pi_i, i = A, B, C, D$ is true iff the car enters i.

We can also consider action schemas $A = (\textit{vars}, \textit{pre}, \textit{post})$, as in STRIPS, which specify discrete changes in the world and are defined in terms of object variables, a precondition that must hold before execution, and a postcondition that will hold after execution, e.g.,

```
(:action GraspObject(?R ?obj)
  :pre  (and (ROBOT ?R) (OBJECT ?obj) (NEAR ?R ?obj))
  :post (GRASPED ?R ?obj))
```

Discrete specifications could be also given by finite automata, where each automaton state corresponds to a set of predicates and each edge corresponds to a discrete action that transforms the discrete state of the world according to the automaton states that it connects. More details about discrete specifications can be found in standard AI books [8], [9].

III. SEMANTICS OVER CONTINUOUS SPACES

The semantics of propositions and predicates are defined over the continuous space $\mathcal{S}$, which consists of a collection of continuous variables that describes the world. As an example, the predicate $\textsc{on}(\textit{book}, \textit{table})$ holds iff the book is actually on the table. This interpretation of which predicates actually hold at a continuous state provides a mapping $\textit{map}_{\mathcal{S} \mapsto \mathcal{Q}}$ from the continuous space $\mathcal{S}$ to the discrete space $\mathcal{Q}$.

Similarly, trajectories over $\mathcal{S}$ give meaning to the actions in the discrete specification. A trajectory over $\mathcal{S}$ is a continuous function $\zeta : [0, T] \rightarrow \mathcal{S}$, parametrized by time. As the continuous state changes according to ζ, the discrete state, obtained by $\textit{map}_{\mathcal{S} \mapsto \mathcal{Q}}$, may also change. As a result, the trajectory ζ follows a discrete action a if $\textit{map}_{\mathcal{S} \mapsto \mathcal{Q}}(\zeta(0))$ satisfies a's precondition and $\textit{map}_{\mathcal{S} \mapsto \mathcal{Q}}(T)$ satisfies a's postcondition. In this way, a continuous trajectory ζ satisfies a discrete specification if it maps to a sequence of discrete actions $[a_i]_{i=1}^n$ that transforms the world from its initial

Department of Electrical Engineering and Computer Science, Catholic University of America, Washington DC 20064. Email: plaku@cua.edu

*This problem has been the subject of much research in recent years [1]–[7], see also various IROS and ICRA workshops on bridging the gap between high-level tasks and motion planning discrete state to a discrete state that satisfies a given formula ϕ_{goal}. The problem statement is then to compute a continuous trajectory ζ that satisfies a given discrete specification.

IV. SIMULTANEOUSLY SEARCHING THE DISCRETE AND THE CONTINUOUS SPACE

Drawing from the success of sampling-based motion planning, the approach uses a tree $\mathcal{T}$ as the basis for conducting the search in the continuous space $\mathcal{S}$. Initially, $\mathcal{T}$ contains only its root, i.e., the initial state $s_{init} \in \mathcal{S}$. As the search progresses, $\mathcal{T}$ is extended by adding new branches, where a new branch is created by selecting a vertex $s \in \mathcal{T}$ and then generating a trajectory $\zeta : [0, T] \to \mathcal{S}$ that starts at s.

The computational efficiency of the tree search depends on the ability of the motion planner to effectively extend $\mathcal{T}$ along those directions that lead to the computation of a trajectory that satisfies the discrete specification. Discrete planning is then used to find such directions. In particular, the discrete planner groups the tree vertices according to $map_{\mathcal{S}\mapsto\mathcal{Q}}$, i.e., $\Gamma_q = \{s \in \mathcal{T} : map_{\mathcal{S}\mapsto\mathcal{Q}}(s) = q\}$, and adds an edge from Γ_{q_i} to Γ_{q_j} if there is a tree branch from a state $s_i \in \Gamma_{q_i}$ to a state $s_j \in \Gamma_{q_j}$. This provides a mapping Γ of $\mathcal{T}$ from the continuous space to the discrete space.

Consider one such group $\Gamma_q \in \Gamma$. Discrete planning is used to determine a sequence of discrete states that transforms the world from q to q_{goal}. Let $q = q_1, q_2, \ldots, q_n = q_{goal}$ be one such sequence, referred to as a discrete plan. A solution trajectory ζ can then be constructed from the discrete plan by extending $\mathcal{T}$ so that it reaches $\Gamma_{q_1}, \ldots, \Gamma_{q_n}$ in succession. In this way, the discrete plan $[q_i]_{i=1}^n$ provides a feasible direction along which the motion planner can attempt to extend $\mathcal{T}$ during the search for a solution trajectory ζ.

Note that there is a computational cost associated with each attempt of the motion planner to extend $\mathcal{T}$ according to the given discrete plan $[q_i]_{i=1}^n$. Moreover, this computational cost varies due to the system dynamics and interactions with the environment. In some cases, it may not even be possible to extend $\mathcal{T}$ along the direction indicated by the discrete plan. Therefore, since the computational cost of the discrete plans are not known a priori, an important aspect of this work relates to the development of effective cost estimations $\textsc{cost} : [q_i]_{i=1}^n \to \mathbb{R}^{>0}$ associated with each discrete plan $[q_i]_{i=1}^n$. We can measure the progress made in extending $\mathcal{T}$ to reach $\Gamma_{q_1} \ldots, \Gamma_{q_n}$, the coverage of each $\Gamma(q_i)$ by states in $\mathcal{T}$, the time spent extending $\mathcal{T}$, and other quantities that provide valuable information in estimating the cost of a discrete plan.

During discrete planning, the cost estimates are used to select more frequently less costly plans. This provides the greedy component to the search by guiding it along directions that are estimated to quickly lead to a solution trajectory. To balance greedy with methodical search, more costly plans are not completely ignored, but are selected less frequently.

In this way, discrete planning and motion planning work in tandem. On one hand, discrete planning guides motion planning during the search for a solution trajectory. On the other hand, information gathered during motion planning is used to update the cost estimates of the discrete plans. As a result of this interplay, the overall approach is able to make proper use of the computational time by selectively sampling and exploring those continuous regions and discrete plans that allow the approach to significantly advance the search for a collision-free and dynamically-feasible trajectory that satisfies the overall discrete specification.

V. FUTURE WORK AND OPEN PROBLEMS

An objective for future work is to further improve the interplay between discrete and motion planning. We are also working on adapting the approach to real robotic platforms. This will allow us to tackle increasingly complex problems arising in robot manipulation, medicine, and automation.

Another challenge is to develop efficient and provably correct algorithms that offer formal guarantees about completeness and overall performance when solving the combined discrete and motion-planning problem. Note that probabilistic completeness, which guarantees a solution will be found with probability approaching one, is as strong a guarantee as we can except given that motion planning is undecidable when combined with discrete logic [12]. To achieve probabilistic completeness, we need to extend reachability arguments to show that the approach systematically searches the space of discrete actions and continuous motions.

ACKNOWLEDGMENT

This work is supported in part by the School of Engineering's Burn's Fellowship.

REFERENCES

[1] R. C. Arkin, "Integrating behavioral, perceptual and world knowledge in reactive navigation," *Robotics and Autonomous Systems*, vol. 6, pp. 105–122, 1990.

[2] D. W. Payton, J. K. Rosenblatt, and D. M. Keirsey, "Plan guided reaction," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 20, no. 6, pp. 1370–1372, 1990.

[3] A. Saffiotti, K. Konolige, and E. H. Ruspini, "A multivalued logic approach to integrating planning and control," *Artificial Intelligence*, vol. 76, no. 1-2, pp. 481–526, 1995.

[4] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. J. Pappas, "Symbolic planning and control of robot motion," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 61–71, 2007.

[5] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic mobile robots," *Automatica*, vol. 45, no. 2, pp. 343–352, 2007.

[6] S. Cambon, R. Alami, and F. Gravot, "A hybrid approach to intricate motion, manipulation and task planning," *Int. J. Robot. Res.*, no. 1, pp. 104–126, 2009.

[7] K. Hauser and J.-C. Latombe, "Multi-modal motion planning for non-expansive spaces," in *Int. Work. Algo. Found. Robot.*, Guanajuato, Mexico, 2008.

[8] S. Russell and P. Norvig, *Artifical Intelligence: A Modern Approach*, 2nd ed. Prentice Hall, 2002.

[9] M. Ghallab, D. S. Nau, and P. Traverso, *Automated Planning: theory and practice*. Morgan Kaufmann, 2004.

[10] E. Plaku and G. D. Hager, "Sampling-based motion and symbolic action planning with geometric and differential constraints," in *IEEE International Conference on Robotics and Automation*, Anchorage, AK, 2010, pp. 5002–5008.

[11] E. Plaku, L. E. Kavraki, and M. Y. Vardi, "Falsification of LTL safety properties in hybrid systems," in *Intl. Conf. on Tools and Algorithms for the Construction and Analysis of Systems*, ser. Lecture Notes in Comp. Sci., vol. 5505, York, UK, 2009, pp. 368–382.

[12] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, "The algorithmic analysis of hybrid systems," *Theor. Comp. Sci.*, vol. 138, no. 1, pp. 3–34, 1995.