Explaining the Behaviour of Hybrid Systems with PDDL+ Planning (Extended Abstract)

Diego Aineto[1], Eva Onaindia[2], Miquel Ramirez[3], Enrico Scala[1] and Ivan Serina[1]

[1]Università degli Studi di Brescia
[2]Universitat Politècnica de València
[3]University of Melbourne

Keywords
Hybrid system, hybrid automata, model checking, automated planning, explanation

1. The Hybrid System Explanation Problem

A *hybrid system* (HS) is a dynamical system that exhibits a discrete and continuous behaviour, and captures the control of continuously evolving physical activities typical in automated manufacturing, chemical engineering and robotics systems. A *hybrid trajectory* is a particular execution of the HS that interleaves discrete and continuous behaviour. We aim to solve the problem of finding a hybrid trajectory that matches a sequence of observations of an HS. This problem, which we name the HS Explanation (HSE hereinafter) problem [1], is related to the Decoding [2] and Plan Recognition problems [3] studied in purely symbolic settings and enables the extension of such theories to hybrid settings [4, 5, 6, 7]. Figure 1 illustrates the HSE problem in a motion domain in planar space where observations are circular regions parameterized by a radius $r \in \{5, 10, 20, 40\}$. The black line represents the true trajectory while colored lines are explanations that match the observations for different values of r.

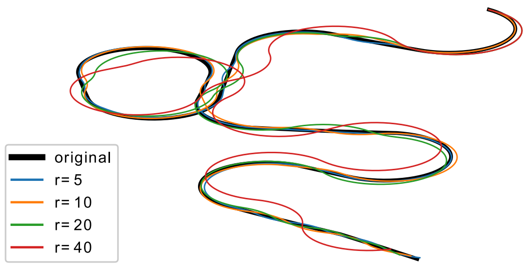

Figure 1: Original trajectory (black) and explanations (colored).

Our proposal is to leverage the formalism of *hybrid automata* (HA) [8] to formulate the

10th Italian Workshop on Planning and Scheduling (IPS-2022)
diego.ainetogarcia@unibs.it (D. Aineto); onaindia@upv.es (E. Onaindia); miquel.ramirez@unimelb.edu.au (M. Ramirez); enrico.scala@unibs.it (E. Scala); ivan.serina@unibs.it (I. Serina)

	Thermostat				Platoon			Flight		
Size	HC	dR	HSE_P	HSE_P$_e$	dR	HSE_P	HSE_P$_e$	dR	HSE_P	HSE_P$_e$
2	10 (0.1)	10 (0.2)	10 (0.5)	10 (0.5)	0	10 (1.1)	10 (0.6)	10 (0.3)	10 (0.6)	10 (0.5)
4	10 (0.5)	10 (0.5)	10 (0.6)	10 (0.6)	0	10 (1.0)	10 (0.6)	1 (14)	10 (2.8)	10 (0.7)
6	10 (0.7)	8 (44)	10 (0.6)	10 (0.6)	0	10 (0.7)	10 (0.6)	0	10 (1.3)	10 (0.8)
8	10 (1.3)	3 (15)	10 (0.6)	10 (0.6)	0	10 (0.8)	10 (0.6)	0	10 (1.6)	10 (0.8)
10	10 (1.6)	0	10 (0.7)	10 (0.6)	0	10 (4.3)	10 (0.7)	0	10 (17)	10 (0.8)
20	10 (11)	0	10 (1.4)	10 (0.7)	0	10 (3.9)	10 (0.9)	0	7 (74)	10 (1.3)
30	10 (60)	0	10 (3.1)	10 (1.0)	0	8 (67)	10 (1.1)	0	1 (127)	10 (1.6)
40	10 (118)	0	10 (5.2)	10 (1.1)	0	1 (191)	10 (1.4)	0	0	10 (1.9)
50	5 (261)	0	10 (8.2)	10 (1.3)	0	0	10 (1.5)	0	0	10 (2.3)
60	0	0	10 (14)	10 (1.6)	0	0	10 (1.9)	0	0	10 (3.5)
70	0	0	10 (20)	10 (1.8)	0	0	10 (2.2)	0	0	10 (4.3)
80	0	0	10 (27)	10 (1.9)	0	0	10 (2.4)	0	0	10 (5.8)
90	0	0	10 (38)	10 (2.2)	0	0	10 (2.8)	0	0	10 (9.4)
100	0	0	10 (49)	10 (2.5)	0	0	10 (3.1)	0	0	10 (14)

Table 1
Coverage and run-time results. Each entry reports the number of solved problems, and the median time in seconds for each system. dR stands for dReach, HC for HyComp.

problem and to use the *planning* technology to solve it. We formulate the HSE problem through two HA: one automaton models the behaviour of the HS, and the other one tracks the run of the HS and checks when the produced trajectory matches the observations. The composition of both HA effectively restricts the trajectories of the HS to those that are consistent with the observations, that is, to the *language of explanations*. We then formalize the HSE problem as an optimization problem that looks over the language of explanations to find the most suitable one. In principle, this problem can be solved as a model checking (MC) problem but MC tools struggle to find concrete trajectories, assume limited knowledge of the systems dynamics, or only handle simple dynamics. To overcome this limitation, we further provide a formal mapping from HA to PDDL+ [9], which enables the use of off-the-shelf automated planners and powerful AI Planning heuristics that have in the past proven very effective for finding trajectories [10].

2. Empirical Analysis

We evaluate our approach on three domains with different types of dynamics: Thermostat (piece-wise constant), Platoon (linear) and Flight (nonlinear) and with problem sizes ranging from 2 to 100 observations. As baselines we use dReach [11] and HyComp[1] [12], two MC tools that are able to generate counter-examples (explanations in our setting). We used the ENHSP planner [13, 14, 15] for the 2 configurations of our planning approach: a basic translation (HSE_P) and an optimisation (HSE_P$_e$) supported by [1, Theorem 3.7]. Table 1 reports the number of problems solved (out of 10) and the median time over successful runs (shown in parentheses) using a time budget of 300s per problem. We observe that HyComp and dReach scale up poorly and were only able to solve the smaller instances. HSE_P shows better performance than the MC tools, but is still unable to solve the full suite of problems as computation times quickly ramp up for the linear and nonlinear domains. HSE_P$_e$, however, solves all the problems while keeping run-time low, hinting that it would be able to scale up to larger instances.

[1]No results for Platoon and Flight are shown for HyComp as it only supports the piece-wise constant dynamics.

References

[1] D. Aineto, E. Onaindia, M. Ramirez, E. Scala, I. Serina, Explaining the behaviour of hybrid systems with pddl+ planning, in: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, 2022, pp. 4567–4573.

[2] D. Aineto, S. Jimenez, E. Onaindia, Observation decoding with sensor models: Recognition tasks via classical planning, in: Proceedings of the International Conference on Automated Planning and Scheduling, volume 30, 2020, pp. 11–19.

[3] M. Ramírez, H. Geffner, Plan Recognition as Planning, in: International Joint conference on Artifical Intelligence, (IJCAI-09), AAAI Press, 2009, pp. 1778–1783.

[4] D. Aineto, S. Jiménez, E. Onaindia, Generalized temporal inference via planning, in: Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning, volume 18, 2021, pp. 22–31.

[5] D. Aineto, S. Jiménez, E. Onaindia, M. Ramírez, Model Recognition as Planning, in: International Conference on Automated Planning and Scheduling, (ICAPS-19), 2019, pp. 13–21.

[6] D. Aineto, S. Jiménez, E. Onaindia, Learning action models with minimal observability, Artificial Intelligence Journal 275 (2019) 104–137.

[7] D. Aineto, S. Jiménez, E. Onaindia, Learning STRIPS action models with classical planning, in: International Conference on Automated Planning and Scheduling, (ICAPS-18), 2018, pp. 399–407.

[8] T. A. Henzinger, The Theory of Hybrid Automata, in: Verification of Digital and Hybrid Systems, Springer Berlin Heidelberg, 2000, pp. 265–292.

[9] M. Fox, D. Long, Modelling mixed discrete-continuous domains for planning, J. Artif. Intell. Res. 27 (2006) 235–297.

[10] M. Wehrle, M. Helmert, The Causal Graph Revisited for Directed Model Checking, in: SAS, 2009.

[11] S. Kong, S. Gao, W. Chen, E. M. Clarke, dReach: δ-Reachability Analysis for Hybrid Systems, in: TACAS, 2015.

[12] A. Cimatti, A. Griggio, S. Mover, S. Tonetta, HyComp: An SMT-Based Model Checker for Hybrid Systems, in: TACAS, 2015.

[13] E. Scala, P. Haslum, S. Thiébaux, M. Ramirez, Subgoaling techniques for satisficing and optimal numeric planning, Journal of Artificial Intelligence Research 68 (2020) 691–752.

[14] E. Scala, A. Saetti, I. Serina, A. E. Gerevini, Search-guidance mechanisms for numeric planning through subgoaling relaxation, in: ICAPS, 2020.

[15] M. Ramírez, E. Scala, P. Haslum, S. Thiébaux, Numerical integration and dynamic discretization in heuristic search planning over hybrid domains, CoRR abs/1703.04232 (2017).