A Hybrid Architecture for Correct-by-Construction Hybrid Planning and Control

Robert P. Goldman Dan Bryce Michael J.S. Pelican
David J. Musliner
SIFT, LLC, Minneapolis, MN
{rpgoldman,dbryce,mpelican,musliner}@sift.net
Kyungmin Bae
Carnegie-Mellon University, Pittsburgh, PA
kquine@gmail.com

1 Abstract

This paper describes Hy-CIRCA, an architecture for verified, correct-by-construction planning and execution for hybrid systems, including nonlinear continuous dynamics. Hy-CIRCA addresses the high computational complexity of such systems by first planning at an abstract level, and then progressively refining the original plan. Hy-CIRCA integrates the dReal nonlinear SMT solver with enhanced versions of the SHOP2 HTN planner and the CIRCA Controller Synthesis Module (CSM). SHOP2 computes a high level nominal mission plan, the CIRCA CSM develops reactive controllers for the mission steps, accounting for disturbances, and dReal verifies that the plans are correct with respect to continuous dynamics. In this way, Hy-CIRCA decomposes reasoning about the plan and judiciously applies the different solvers to the problems they are best at.

2 Introduction

In this paper we describe Hy-CIRCA, an architecture for verified, correct-by-construction planning and execution for hybrid systems, including nonlinear continuous dynamics (see Figure 1). Hy-CIRCA addresses the high computational complexity of nonlinear hybrid systems by first planning at an abstract level, and then progressively refining the original plan. During this refinement process, Hy-CIRCA incorporates formal verification at increasing levels of fidelity.

Hy-CIRCA is an extension of our Playbook[1] approach for controlling multiple autonomous agents to cover hybrid discrete/continuous planning and control, with nonlinear continuous dynamics. The Playbook approach aims to make it easy for users to exert supervisory control over multiple autonomous systems by "calling a play" [9]. The Playbook approach is implemented by combining (1) a human-machine interface for commanding and monitoring the autonomous systems; (2) a hierarchical planner for translating commands into executable plans; and (3) a smart executive to manage plan execution by coordinating the control systems of the individual autonomous agents, tracking plan execution, and triggering replanning when necessary.

[1] Playbook® is a registered trademark of SIFT, LLC.

Hy-CIRCA integrates the dReal nonlinear SMT solver [4] with enhanced versions of the SHOP2 [11] planner and the CIRCA Controller Synthesis Module (CSM) [10, 6]. The planning process in Hy-CIRCA proceeds in 5 steps: (1) SHOP2 computes an approximate, nominal mission plan. While computing this plan, Hy-CIRCA also computes a hybrid automaton model of the plan, featuring more expressive continuous dynamics. (2) dReal solves this hybrid model, establishing the correctness of the plan, and refining it by computing values for its continuous parameters. This mission plan uses projection to handle resources and find paths to the goal state. However, it is still an open-loop plan. (3) To build an executable plan, Hy-CIRCA extracts specifications for closed-loop, hard real-time supervisory controllers that will achieve each step of the plan and reject disturbances from outside sources of change (e.g., adversaries or nature). (4) Based upon these specifications, CIRCA CSM plans the supervisory controllers. The CSM uses an over-approximating abstraction of the continuous dynamics. (5) Finally, dReal ensures correctness of the controllers by verifying they meet the specifications, using a higher-fidelity nonlinear hybrid model.

Hy-CIRCA has superficial similarities with established three layer architectures (TLAs) for robotics, but those typically feature deliberative planning, reactive programming for a smart executive, and then a platform control interface [5]. The TLA provides a combination of open-loop projective planning to efficiently reason about mission goals and manage resources, with closed-loop, event-driven interaction with low-level platform control. In most TLAs, considerations of continuous and hybrid dynamics are confined to the executive layer, the executive is usually manually programmed rather than automatically synthesized, and the relationship between the planning and executive layers is *ad hoc*. TLAs contrast with hierarchical schemes such as CHARON [1], that are homogeneous across abstraction layers, permitting systematic reasoning about relations between components in terms of traces and trace refinements. Hy-CIRCA shares the ability to systematically reason about relations between layers, and permits higher-level behaviors to constrain synthesis of lower-level behaviors. Hy-CIRCA is unique in handling nonlinear continuous dynamics.

As the first step in developing Hy-CIRCA, we have constructed a proof-of-concept implementation of key parts of Hy-CIRCA, and tested it on a demonstration problem involving multi-agent firefighting using uninhabited aerial vehicles (UAVs). In this paper, we describe this aerial firefighting scenario and how Hy-CIRCA meets its challenges.

Our contributions include: a method of using the SHOP2 planner to perform hybrid automaton model construction; a logical formalization of the SHOP2-generated plan as a hybrid system, for use in the dReal SMT solver; techniques for extracting controller specifications from an HTN plan; and techniques for verifying the correctness of CIRCA closed-loop controllers on hybrid systems.

3 Scenario of Use

The following motivating use case illustrates how the Hy-CIRCA architecture can be used for hybrid planning and control of multiple-platform packages of autonomous systems. In this multi-UAV firefighting scenario (see Figure 2), the fleet of UAVs includes *waterbombers* that drop water or retardant on a fire, *spotters* that localize the fires and

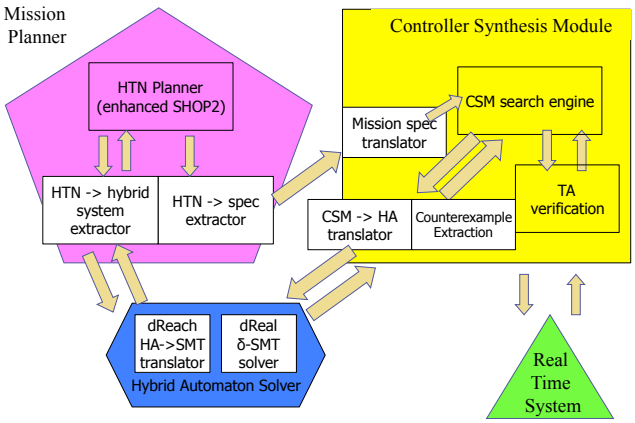

Fig. 1: Hy-CIRCA synthesizes hybrid controllers to satisfy mission specifications expressed in high-level temporal logic through two-way interactions between the Mission Planner, the Controller Synthesis Module, and the Hybrid Automaton Solver.

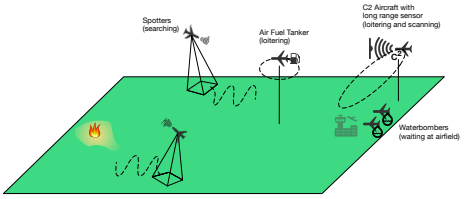

Fig. 2: Multiple-UAV firefighting scenario.

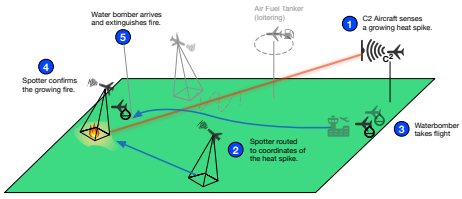

Fig. 3: Plan to extinguish a fire.

transmit targeting coordinates to the waterbombers, *C2 aircraft* that coordinate operations and have long-range sensors that detect possible fires, and *tankers* that provide in-air refueling.

The mission's objective is to extinguish a fire that has just been spotted by the long-range, low-accuracy sensors on a C2 aircraft. An airborne fire manager on the C2 aircraft uses his Playbook interface to call the *extinguish* play, tailoring the play to the situation by incorporating information about the location of the suspected fire, the assets available to the team, and their current state/location. Given the plan library and any tailored parameters from the operator, Hy-CIRCA first invokes the SHOP2 HTN planner to generate a complete set of tasks for the mission. The first step in the mission plan (see Figure 3) is for a spotter to go to the vicinity of the suspected fire to confirm its existence. After confirmation, the fire can be extinguished by dropping a load of retardant from a waterbomber. SHOP2 also plans a route for the waterbomber to reach the area of the fire to extinguish the fire. In order for the waterbomber to correctly target the fire, it must get a stream of location information from a spotter as it drops the retardant.

SHOP2 chooses a take-off time for the waterbomber that attempts to get it to the fire area shortly after the spotter has arrived. This will avoid fuel waste that would occur in a plan where the waterbomber arrives first and must loiter, waiting for the spotter. But we don't want the waterbomber to arrive too late: even though the spotter uses less fuel than the waterbomber, we still don't want to waste its fuel, or keep it from other uses. Note that this means that the problem has *required concurrency* [3]. Required concurrency is a feature of more difficult temporal planning/scheduling problems in which there are "too early," as well as "too late" constraints. Problems with required concurrency differ from simpler temporal problems whose temporal aspect can be solved by choosing the earliest feasible activity start times.

SHOP2's plan is only approximate, because of limits in its computations about real continuous quantities such as fuel (in particular, its inability to solve systems of simultaneous equations). But SHOP2 can more efficiently solve the discrete sequencing problems than dReal, and it can invoke special solvers to synthesize waypoint sequences.

Once the initial plan has been computed, Hy-CIRCA uses the higher fidelity reasoning offered by dReal to refine it. As a side-effect of computing its plan, Hy-CIRCA's version of SHOP2 builds a hybrid systems SMT problem. Hy-CIRCA uses dReal to solve this problem, where a solution is a satisfying trajectory through the high-dimensional hybrid space. By solving that problem, dReal will synthesize continuous parameters for the mission plan. In our tests, dReal solved a system of nonlinear constraints to choose the fuel and flame retardant loads for the waterbomber, and refined the mission schedule based on more accurate models of flight than those used by SHOP2.

The mission plan guides the operation of the autonomous systems, but is not sufficient for their closed-loop control. Hy-CIRCA uses an enhanced version of the CIRCA Controller Synthesis Module (CSM) to automatically synthesize controllers for the platforms (waterbombers, spotters, etc.) in the mission. These controllers will constitute the smart executives for those platforms.

The first step of the controller synthesis process is to generate controller specifications from the mission plan. These specifications include both temporal logic invariants and goals. For our current Hy-CIRCA proof of concept, we extracted the specifications for the waterbomber aircraft in the mission. These specifications include control operations used in flight, representations of known disturbances, temporal logic representations of the goals ("eventually the fire should be extinguished"), temporal logic invariants ("the waterbomber must release its load within k time units from receiving a target message from the spotter"), and signal temporal logic (STL) [8] hybrid invariants ("the vehicle must always maintain a fuel reserve of at least n gallons").

The CIRCA CSM uses these inputs to synthesize a closed-loop, real-time discrete outer-loop controller for each platform. Note that these controllers will perform coordinated actions, directed by the mission plan that has been translated into temporal logic specifications. For example, the spotter will repeatedly transmit targeting information until the waterbomber has dropped its load. Similarly, the waterbomber will wait in the vicinity of the fire until it has received the targeting information, and is constrained to drop its load before the targeting information becomes stale.

The CIRCA CSM reasons about continuous processes only as approximated by upper and lower bounds on temporal durations. For example, its reasoning about whether

it can reach the target quickly enough is based on whether it can initiate the motion soon enough, and is verified in terms of time bounds on its flight processes.

For more accurate reasoning about the control of continuous processes, Hy-CIRCA re-checks the controller using dReal. We developed a technique for translating the CIRCA controllers and STL invariants into a hybrid automaton representation that can be checked by dReach (a preprocessor translating hybrid automata for dReal) and dReal. We tested this algorithm by translating the waterbomber controller into a hybrid automaton, the STL invariants into a separate automaton, forming the product, and then checking reachability. The reachability computation was done using dReach's translation from automaton to dReal SMT formulas. *E.g.*, in one of our tests, we checked the invariant that the waterbomber would always have an adequate fuel reserve.

If Hy-CIRCA finds that an invariant cannot be verified, it will use a process of culprit extraction to translate the counterexample into information that the CIRCA CSM can use to guide backjumping and repair the synthesized controller [7]. We have developed a method for performing this culprit extraction. We tested the approach on an example where the waterbomber's controller originally generated a plan that involved monitoring a fuel level warning, which is signaled when a threshold fuel level is reached (like the fuel level warning in a car). The original synthesized controller would go into a special fuel-conserving flight mode and return to base if the low fuel warning was triggered. In some missions, where the flight radius is lower, this controller will be verified to work correctly.

However, if the flight radius is longer, then dReal will detect that a failing state can be reached if the fuel warning goes off when the aircraft is beyond some distance d from base. From this counterexample, Hy-CIRCA can extract a culprit that indicates that the controller using the special return-to-base flight mode is not safe. This will cause the CSM to backjump and choose the (more expensive) recovery action of in-air refueling. The resulting revised controller then will be verified by dReal.

4 Conclusions and Future Work

We have described our Hy-CIRCA framework for integrated planning, controller synthesis, and verification for nonlinear hybrid domains. Hy-CIRCA decomposes complex mission planning into strategy planning with SHOP2, controller synthesis with CIRCA, and verification with dReach. Some of the most challenging and novel aspects of this framework are how the tools integrate to solve the overall problem. For instance, SHOP2 and dReal interact to solve planning and scheduling with numeric resources. SHOP2 and CIRCA collaborate to synthesize low-level controllers for high-level actions that must satisfy temporal properties. CIRCA and dReach cooperate to verify the controllers with respect to the underlying nonlinear continuous change.

As we continue to develop the Hy-CIRCA framework, the important remaining issues concern how to automatically divide the overall problem among the Hy-CIRCA components and how to effectively evaluate these decisions. We recognize that the division of decision making between the tools is an important area for future research, especially given that we have recently made advances in scaling dReal's discrete decision-making capabilities [2]. We have designed algorithms for SHOP2/CSM and CSM/-

dReach integration, and tested them on cases drawn from the firefighting scenario. We are about to begin a second phase of the project to complete the integration and extend our set of experiments. We are also working to extend the capabilities of the system in a number of areas. In order to handle the full generality of the temporal specifications handled by the Hy-CIRCA CSM, we are improving the native finite trace verification of temporal logic specifications in the CSM. The current timed automaton verifier in the CSM only checks safety conditions. We will extend it to handle liveness conditions, as well. While the basics of liveness checking are well understood, our algorithms for extracting nogoods – used for backjumping – need extension in order to accommodate counterexamples to liveness goals.

Acknowledgments Thanks to the anonymous reviewers, Laura Humphrey, Sicun Gao, and Soonho Kong for comments and advice. This material is based upon work supported by the U.S. Air Force (AFRL) under Contract No. FA9550-15-C-0030. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the authors and do not reflect the views of the U.S. Air Force (AFRL).

References

1. Alur, R., Dang, T., Esposito, J., Fierro, R., Hur, Y., Ivančić, F., Kumar, V., Lee, I., Mishra, P., Pappas, G., Sokolsky, O.: Hierarchical hybrid modeling of embedded systems. In: EMSOFT, Lecture Notes in Computer Science, vol. 2211, pp. 14–?? Springer Verlag (2001)
2. Bryce, D., Gao, S., Musliner, D.J., Goldman, R.P.: SMT-based nonlinear PDDL+ planning. In: Proceedings National Conference on Artificial Intelligence. pp. 3247–3253 (2015)
3. Cushing, W., Kambhampati, S., Mausam, Weld, D.S.: When is temporal planning really temporal? In: Veloso, M.M. (ed.) Proceedings of the 20th International Joint Conference on Artificial Intelligence. pp. 1852–1859 (2007)
4. Gao, S., Kong, S., Clarke, E.M.: dReal: An SMT solver for nonlinear theories over the reals. In: CADE (2013)
5. Gat, E.: Three-layer architectures. In: Kortenkamp, D., Bonasso, R.P., Murphy, R. (eds.) Artificial Intelligence and Mobile Robots. AAAI Press/MIT Press, Cambridge, MA (1998)
6. Goldman, R.P., Musliner, D.J., Pelican, M.J.: Exploiting implicit representations in timed automaton verification for controller synthesis. In: Proceedings of the 2002 Hybrid Systems: Computation and Control Workshop (Mar 2002)
7. Goldman, R.P., Pelican, M.J.S., Musliner, D.J.: Guiding planner backjumping using verifier traces. In: Zilberstein, S., Koehler, J., Koenig, S. (eds.) Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling. pp. 279–286 (Jun 2004)
8. Maler, O., Nickovic, D.: Monitoring temporal properties of continuous signals. In: Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems, pp. 152–166. Springer (2004)
9. Miller, C.A., Goldman, R.P., Funk, H.B., Wu, P., Pate, B.: A playbook approach to variable autonomy control: Application for control of multiple, heterogeneous unmanned air vehicles. In: AHS 60th Annual Forum Proceedings. pp. 2146–2157. American Helicopter Society, Alexandria, VA (Jun 2004)
10. Musliner, D.J., Durfee, E.H., Shin, K.G.: CIRCA: a cooperative intelligent real-time control architecture. IEEE Transactions on Systems, Man and Cybernetics 23(6), 1561–1574 (1993)
11. Nau, D., Au, T.C., Ilghami, O., Kuter, U., Murdock, J.W., Wu, D., Yaman, F.: SHOP2: An HTN planning system. Journal of Artificial Intelligence Research 20, 379–404 (2003)