What, how, and when? A hybrid system approach to multi-region search and rescue

Patrick V. Haggerty[a], Ben A. Strasser[a], Anthony J. Bloch[a], and Eva M. Kuntz[a]

[a]General Dynamics Mission Systems, Bloomington MN

ABSTRACT

Multi agent hybrid dynamical systems are a natural model for collaborative missions in which several steps and behaviors are required to achieve the goal of the mission. Missions are tasks featuring interacting subtasks, such as the decision of where to search, how to search, and when to transition from a search behavior to a rescue behavior. Control in hybrid systems is poorly understood. Theoretical results on state reachability rely on restrictive assumptions which hinder formal verification and optimization of such systems. Further difficulties arise if there are no a priori ordering or termination conditions on the intermediate steps and behaviors. We present a flexible framework to enable decentralized multi agent hybrid control and demonstrate its efficacy in a class of multi-region search and rescue scenarios. We also demonstrate the importance of dynamic target modeling at both levels of the hybrid state, i.e. estimating which region targets are in, how search behavior affects this estimate, and how the targets move between and within regions.

Keywords: hybrid dynamical systems, multi-agent autonomy, multi-agent control, autonomous search and track, target tracking, predictive models

1. INTRODUCTION

A challenge for control of autonomous unmanned vehicles (UxVs) for commercial and military applications is reconciling the difference in languages of humans and UxVs. A human specifies high-level discrete tasking in a mission specification (e.g. "move to this region", "search here", "prevent movement toward this point"), while UxVs require low-level inputs, e.g., algorithmically-determined continuous velocities or accelerations. Tasking and control inputs operate at different time scales; a single high-level task requires completion of a complex sequence of low-level control sequences.

This discrepancy makes hybrid dynamical systems a natural model for many complex activities. Informally, a hybrid dynamical system is a dynamical system with both discrete states (e.g. a finite state machine) and continuous states (e.g. dynamics governed by differential equations). A discrete element is called a mode, and each mode determines how the continuous states evolve. We give a formal definition in Section 2 . Just as with traditional dynamical systems, it is easy to incorporate control inputs. Hybrid control inputs involve changing discrete state (executive control), changing velocity/acceleration (traditional physical control), or both.

We use hybrid control systems to model complex missions by associating the discrete modes to high level mission tasks. Each mode specifies a control algorithm designed to accomplish the underlying goal of that mode. A mission progresses as the hybrid dynamical system evolves; agents use control algorithms to navigate the operational area and complete tasks, switching to new modes/tasks as they complete current ones. One problem we give special attention to is that of open-ended tasks, e.g. searching a predefined region. Unlike tasks with clear completion criteria (like reaching a position in a given amount of time), deciding when to move on from an open-ended task is a nontrivial matter.

Designing an optimal control policy for a single task is challenging and the addition of mode switching adds additional complexity which makes analysis of hybrid systems extremely difficult. Notably, the reachability problem, which asks if a dynamical system will reach or avoid certain success or failure states, is undecidable even for linear hybrid systems (essentially the simplest examples of a hybrid dynamical system).[1] Addressing reachability requires formal verification involving accounting for every contingency and time horizon, no matter

Send correspondence to patrick.haggerty@gd-ms.com and benjamin.strasser@gd-ms.com

how operationally unlikely that case may be. Such a negative theoretical result may lead one to doubt the utility of hybrid models. In a previous paper,[2] the first two authors considered partial verification through temporal logic specifications and used these to construct policies for robots playing a class of hybrid defense/pursuit games. In the current paper we consider a class of multi-agent hybrid control systems with imperfect information, non-deterministic elements, and multiple open-ended search goals which we call the Multi Region Search (MRS) problem. We show how information-theoretic control and estimation aid UxVs through such problems.

Such problems can be viewed as active information gathering problems. The multi-robot multi-target active information acquisition was formulated for Simultaneous Location and Mapping (SLAM),[3] but the proposed techniques focus on linearization and optimal control of Gaussian distributions. Exploration and active mapping techniques are specific to log-odds occupancy[4] or multi-class distributions.[5] Recently, research has addressed safe[6] and resilient[7] planning as well as learning-based policies.[8,9] Reinforcement learning solutions generally don't converge to equilibrium, and hence lack the theoretical guarantees provided by the game theoretic approach. Introducing theoretical guarantees on reinforcement learning methods is an active area of research.[10] The problem of defining a search termination criterion like we seek in the multi-region scenarios appears novel in the problem space.

Our scenarios are derived from a search and rescue scenario. This scenario involves using a team of unmanned aerial vehicles (UAVs) to search a known region (such as a National Park) and quickly identify hikers in need of aid. We will closely list our assumptions later, but our prior knowledge of the area allows us to identify smaller subregions (Regions of Interest, ROI) of the park where the hikers can be (based on terrain), to assume we know the number of hikers in need of aid (total number from entrance/exit monitoring but not how many within each subregion), and to model how the hikers can move within each subregion (based on existing trails and conditions). The existence of such subregions can drastically reduce the amount of area that the UAVs need to search, but this only results in a time save if the UAVs know when they have sufficiently searched a region and should move on (and then where to search next). Much of this paper assumes that hikers do not leave the subregion that they start in, though we discuss how to relax these conditions.

We model the UAVs in these scenarios as hybrid control agents. High-level executive control is about choosing which ROI to search for hikers (or whether to stay near an identified hiker in order to provide aid or help direct a better-suited platform to the hiker), while the physical control is about where to move and point sensors within an ROI. In terms of a finite state machine, the UAVs have search modes for each ROI as well as a "track" mode. Our goal is twofold: to determine good search policies (mostly at the physical control level, but overall search pattern for the ROIs is important, too) and to introduce estimation tools enabling the UAVs to evaluate the progress of their search, both for executive control decisions and reporting to a human operator. For simplicity, we will assume that the number of hikers relative to the number of UAVs is small, so that naïve tracking algorithms will suffice. Higher target densities will require more sophisticated multi-target tracking solutions[11] to keep track of many hikers.

This document proceeds as follows. Section 2 introduces the formalism of hybrid dynamical systems and our state estimation techniques. Section 3 details how our state estimation techniques drive search behavior, both in the searching of a particular region, how to stop searching a region, and choosing the next region to search. In Section 4 we outline the details of the MRS scenario. We outline the results in Section 5 and discuss conclusions and future research in Section 6.

2. NOTATION AND BACKGROUND

We describe our MRS scenarios using the formalism of hybrid dynamical systems. A *hybrid dynamical system* is a tuple $(Q, X, f, Init, Inv, E, G, R)$ where $Q = \{q_1, q_1, \dots\}$ is a finite set of discrete states, $f : Q \times X \to \mathbb{R}^n$ defines a vector field controlling physical dynamics in each discrete state, $Init \subseteq Q \times X$ denotes the initial states, $Inv : Q \to \mathcal{P}(X)$ denotes the invariant sets for each discrete state, $E \subseteq Q \times Q$ denotes possible transitions between discrete states, $G : E \to \mathcal{P}(X)$ associates discrete state transitions to physical "guard sets", which specify physical conditions which must be met in order for a specific transition to occur, and $R : E \times X \to \mathcal{P}(X)$ is a reset map denoting physical state changes which occur concurrently with discrete state changes. In many cases, the guard sets and reset maps are trivial. We introduce a hybrid control structure in this framework, resulting in a traditional affine control system at each discrete state while simultaneously allowing agents to have some control over the discrete states of the system.

In our scenarios, if the overall region is divided into n ROIs $R0, R1, R2, \ldots Rn$, then the searching UAVs will have $n+2$ modes, one Search mode per ROI plus one Track mode and one Done mode (essentially a return to base behavior). All possible transitions are allowed and can be made at any time (trivial guard sets). Our simulations are with single-integrator physics, so UAVs may instantaneously change velocity (within a maximum speed).

The hikers (targets) are modeled much more simply. They are assigned a random starting point within some ROI and by randomly picking a point in the ROI and moving toward that waypoint. Note this makes the targets neutral with respect to the UAVs goal of finding all targets; hikers neither avoid nor encourage detection. An example of an in progress scenario is shown in Figure 1.

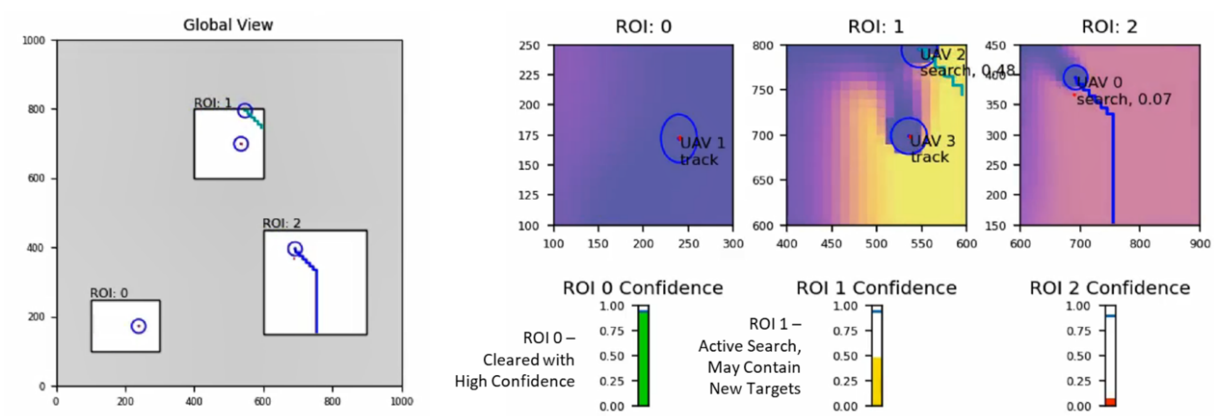

Figure 1. An in-progress multi-region search scenario with 3 ROIs (left). UAVs execute an efficient, "Potential Target Distribution"-based search (right, see Section 3).

We introduce the notions of global and local potential target distributions (GPTD, LPTD) to represent how many targets are in each ROI and where targets are likely to be detected within an ROI. There is one GPTD, which is the estimated target distribution between the ROIs, and one LPTD per ROI, which are the expected number of detections at locations in the ROI. We discretize each ROI and represent each LPTD as a matrix which can be visualized as a heatmap. The sum of each LPTD matches the corresponding entry of GPTD for consistency. When talking generally, or the distinction does not matter, we often just say potential target distribution (PTD).

The GPTD and LPTD are updated based on UAV movement and assumed target motion. First, as the UAVs move within an ROI, they receive sensor readings on whether or not there is a target in their sensor field of view (and if so, a noisy reading on the (x,y) position of the target). We assume no false negative readings, so as UAVs search locations with no targets, the corresponding matrix entries of the LPTD are set to zero. Second, the quantity of potential targets (heat) removed by this process is redistributed at the GPTD level, proportional to the GPTD estimate. This accounts for the fact that if we didn't see a target at a particular location in a particular ROI, that may be because it is either elsewhere in the same ROI or in a different ROI. If we do not assume that targets remain in their ROI, this is where we can model this. One model would be to assume the targets switch ROIs according to a Markov process. We would then periodically (matching the timescale of the Markov process) multiply the Markov transition matrix with the GPTD. Once we have updated the GPTD, we renormalize the LPTDs to match the current GPTD.

Finally, we update the LPTDs to model the movement of targets within an ROI. In the simplest case, we assume uniform random (directionless) movement (note this assumption differs slightly from how targets actually move) and periodically convolve our PTD estimate with the 3×3 kernel

$$K = \begin{bmatrix} 0.07716008 & 0.13747204 & 0.07716008 \\ 0.13747204 & 0.14147151 & 0.13747204 \\ 0.07716008 & 0.13747204 & 0.07716008 \end{bmatrix}$$

which is a local model of this assumed movement. See Figure 2 for a visual representation of this process. The time between consecutive convolutions is the time it would take for one target in the center of a local 3×3 grid to reach the center of an orthogonally-adjacent square. With more complex target movement (including adversarial targets), we replace K with another (likely location-dependent) heat kernel that better models the target movement. Note that the convolution process naturally "drops" some heat off the edges of the ROI boundary. As we assume targets do not leave an ROI, we redistribute the lost heat uniformly throughout the LPTD estimate.

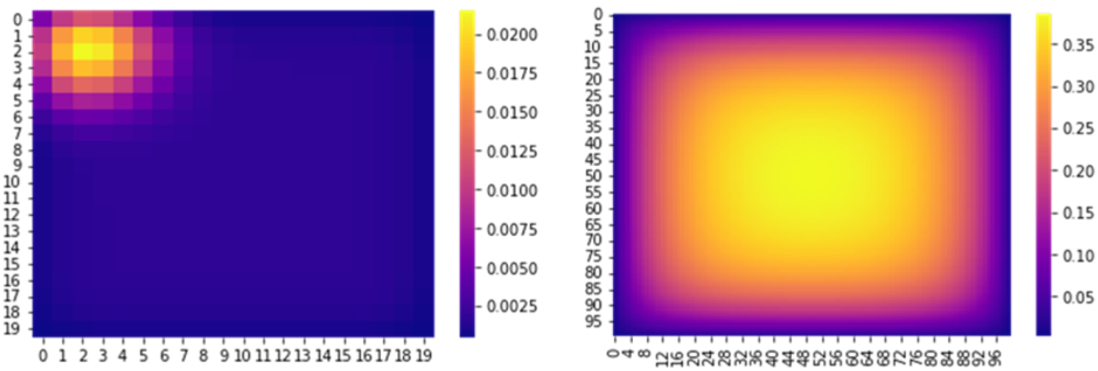

Figure 2. The LPTD and kernel K models uncertainty in moving target location. The left shows a LPTD matrix that started as a single hotspot and has been convolved with K five times. The right shows the same PTD matrix after 100 convolutions.

The GPTD and LPTDs are powerful tools. Explicitly, they simultaneously keep track of the number of targets likely in each regions and likely locations of any such target. Implicitly, it tracks freshness of information. As time passes, the convolution "spreads" heat throughout the ROI; locations the UAVs have searched will eventually become uncertain again, perhaps enough so to warrant a revisit.

3. SEARCHING ALGORITHMS

3.1 Searching with PTD

From the definition of LPTD estimate and its update, we can encode the goal of searching a region as reducing the sum of entries (total) heat to 0 in an efficient manner. It is unlikely that the total heat is actually ever reduced to 0, but it is still the goal. If we define the confidence of a region to be

$$C = \max\{0, 1 - \sum_{i,j} LPTD_{i,j}\},$$

then C is an estimate of the probability that there are zero untracked targets in the ROI. We can use $C > \alpha$ as a condition for stopping the search. In our scenarios we used $\alpha = 0.9$ and $\alpha = 0.95$.

There are many ways to accomplish this heat reduction goal. One is to use a greedy search. In this search method, UAVs set a small radius around their current position and maneuver directly to that point. There is a simple deconfliction process; UAVs communicate their planned waypoints to avoid multiple UAVs searching the same location. This is a computationally lightweight algorithm, leverages the prior on potential target locations, but is not particularly efficient or collaborative.

Another approach is what we call a *nonmyopic* search, which has a longer planning horizon. When a UAV needs a plan, it chooses the farthest away global maximum of the PTD (the "nonmyopic" part) and then plans to visit intermediate local maxima without increasing the distance travelled too much. This provides an approximate solution to the optimization problem of maximizing heat reduction per distance travelled. Of course, traditional planning algorithms like A* may be folded into this algorithm. The UAVs plan asynchronously (they replan when they reach their final waypoint) and communicate their plans to their allies, so later iterations of the planning algorithm can avoid visiting the same or nearby locations, increasing efficiency of the search. See Figure 3 for an example.

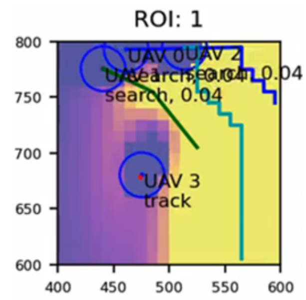

Figure 3. In the nonmyopic search, UAVs plan trajectories (colored lines emanating from the UAVs) and communicate these trajectories for consideration during others' planning cycles.

3.2 Searching Multiple Regions

However a region is searched, be it our greedy or nonmyopic searches or another search pattern that uses the PTD, once the UAVs reach the threshold for considering a region sufficiently searched, they must decide which ROI to search next.

There are a number of strategies, the two basic ones are simultaneous search or sequential search, that is, the UAVs in a region may all decide to go to the same ROI (either the closest ROI or the one with the most potential targets) or split up and go to different ROIs (proportional to the potential target ROI distribution). These are illustrated in Figure 4.

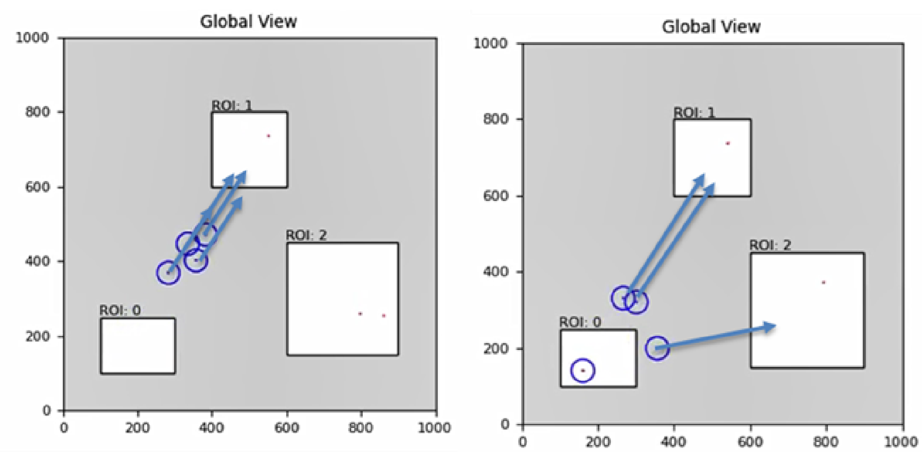

Figure 4. Sequential (left) and Simultaneous (right) search strategies.

Which strategy is most appropriate is a complicated question and may depend on the geometry of the ROIs within the region and objective functions which are important to an operator. For example, while confidence may rise more slowly using a simultaneous search, we start collecting information about all ROIs sooner than if we had left some for later. Work on strategy selection is still ongoing.

4. SCENARIO OVERVIEW

4.1 Base Scenario

We demonstrate the efficacy of our approach with an example scenario. In this first scenario, the global region is a 1KM x 1 KM square region with three ROIs. We employ four UAVs to search for three targets. We assume the UAVs maneuver at 3 m/s while the targets maneuver at 2 m/s. The assumptions that UAVs outnumber targets and that searchers move faster than targets let us use a naïve tracking behavior, as once a UAV finds a target, the target can't run away. If either of these assumptions are not guaranteed, we need a more sophisticated tracking behavior, such as one based a Joint Multitarget Probability Distribution.[11]

The prior on the target distribution among the ROIs was [0.3,0.4,0.3], though in reality the targets are randomly distributed. The UAVs begin in the lower left corner of the global region, near ROI 0. In the sequential search, the UAVs will search as follows:

1. ROI 0 because it is closest to the UAV start point and on the way to either ROI 1 or 2, then
2. ROI 1 because there are more potential targets in ROI 1 than ROI 2, then finally
3. ROI 2

In the simultaneous search, the UAVs search as follows:

1. ROI 0 because it is closest to the UAV start point and on the way to either ROI 1 and 2, then
2. Split up to search ROI 1 and 2. The assignment is based on number of UAVs progressing to this stage of the search (i.e., how many are not staying behind with targets) and the prior target distribution between ROI 1 and 2.

4.2 Variant Scenario

For more complexity, we introduce the concept of discrete region features which drive the target distribution between ROIs and behavior within ROIs. Our region features include

- time of day (night/day),
- weather (cloudy/clear skies),
- terrain (hilly/flat), and
- landmarks (water source/look-out tower/none).

Targets distribute between ROIs based on the global set of features and move differently within a given ROI based on the features of each region. To obtain a prior for our GPTD and kernels for our LPTDs, we implement an artificially-intelligent virtual agent (AIVA) as a collection of neural networks that we train to predict distribution and movement probabilities of targets. AIVA learns how targets are distributed based on region features and learns how they move in a region based on the region features and the location(s) of the UAV(s) in that specific region. The UAVs are equipped with AIVA to improve the accuracy of the GPTD and LPTD matrix over time and to remove the need for a priori knowledge of target distribution throughout the regions.

5. EXPERIMENTAL RESULTS

We briefly review experimental results. We ran the base scenario with the UAVs performing sequential search. We measured the time it took to find all 3 targets. Instead of the total simulation time, we measure the following. For each UAV, we count the number of timesteps that the UAV is physically located in the ROI corresponding to their search mode. We record the maximum count over all agents. This essentially removes the travel time between ROIs from consideration.

To ensure that times are somewhat comparable between simulation runs, when distributing the targets, we force one target to be in ROI 2.

We experiment with four different search behaviors:

- "Zamboni" - the UAVs drive back and forth in the ROI. The PTD-based hybrid control scheme is not used.

- "Random" - the UAVs use PTD state monitoring to determine when an ROI is sufficiently searched, however, the UAVs search by choosing random waypoints in the ROI to search
- "Greedy" - the UAVs search each ROI by choosing their waypoint to be the hottest point in the PTD (limited to a fixed radius around their current position). The UAVs use the PTD to determine when an ROI is sufficiently.
- "Nonmyopic" - the UAVs use the search algorithm described in Section 3 and use the PTD to determine when an ROI is sufficiently searched.

For each of these four behaviors, we ran 15 trials and averaged the search times as defined above. The results are shown in Figure 5. All of the PTD/hybrid control search behaviors significantly outperformed the preplanned Zamboni search. The Zamboni search suffers from being inefficient and inaccurate; the UAVs were missing targets, requiring backtracking. It is interesting to note that the random search greatly outperformed the Zamboni pattern and was not much worse than the greedy search. This speaks to the importance of the PTD state monitoring; monitoring search progress in terms of potential targets even with an unintelligent search behavior enabled the UAVs to have the confidence to move on. The search was inefficient, but more thorough than Zamboni. The greedy search was only 10% faster than the random search; the random search had the side effect of spreading the UAVs throughout the region, while greedy tended to keep the UAVs bunched together at the start of the search. The nonmyopic search was most efficient of all, demonstrating the importance of in situ planning and communication.

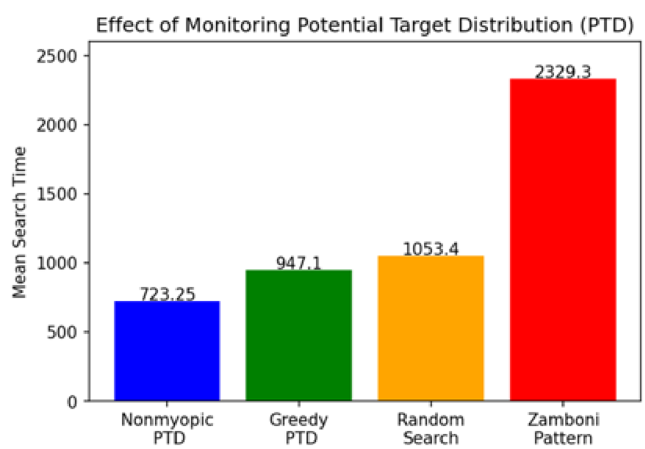

Figure 5. PTD-based hybrid control significantly increases search efficiency compared to preplanned search behaviors like Zamboni patterns.

In the variant scenario, we first measured how well AIVA predicts the initial GPTD based on the region features. AIVA achieved a mean squared error of as little 0.542 with 1500 training samples.

For target motion prediction, we use Kullback-Leibler divergence (KL divergence) to measure the different between AIVA's prediction and the ground truth model. Recall KL divergence for discrete probability distributions P and Q is defined on a sample space X is

$$D(P|Q) := \sum_{x \in X} P(x) \log\left(\frac{P(x)}{Q(x)}\right).$$

This quantity is 0 if P and Q are the same distribution and is ∞ if $Q(x) = 0 \neq P(x)$. The targets had different behaviors depending on the number of UAVs present in the region; targets would wander in a circle if there are no UAVs present, stop if there are three, and move evasively if there are one or two UAVs. Table 1 summarizes the results.

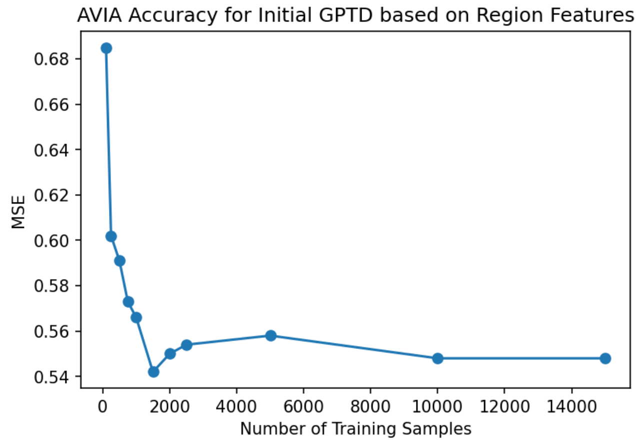

Figure 6. AIVA provides an accurate GPTD estimate with an elbow point at around 1500 training samples.

Number UAVs	Training Samples	KL
0	3,985	1.0630
1	24,267	6.5716
2	34,329	6.6480
3	20,486	4.7569

Table 1. KL divergences for the various target behaviors. AIVA struggles with the more complex evasive behaviors.

6. CONCLUDING REMARKS

We presented a flexible hybrid control system framework that leverages state estimation tools to make high confidence in situ decisions while executing open-ended tasks. We think this approach is particularly valuable as it enables dynamic, adaptive mission execution that is resilient to inaccurate/stale prior intel. In search and track scenarios, our state estimation tools provided compounded benefits as they were used to drive search patterns as well as determine when the search was over. Future research will attempt to refine the state evaluation and AIVA prediction results in more adversarial situations where targets are trying to avoid being detected, and operate in unknown/unexplored regions that may contain visual occlusions.

ACKNOWLEDGMENTS

The authors thank Raja Suresh, John Polson, Scott Blanchard and GDMS leadership for their funding, support, and counsel. We also thank John Kleider, Mike Bigrigg, and Scott Potter for sharing their technical expertise.

REFERENCES

[1] Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T., Ho, P.-H., Nicollin, X., Olivero, A., Sifakis, J., and Yovine, S., "The algorithmic analysis of hybrid systems," *Theoretical Computer Science* **138**(1), 3–34 (1995). Hybrid Systems.

[2] Haggerty, P. V. and Strasser, B. A., "Awareness, control, and trust in multi-agent hybrid dynamical systems," in [*Open Architecture/Open Business Model Net-Centric Systems and Defense Transformation 2022*], Suresh, R., ed., **12119**, 121190A, International Society for Optics and Photonics, SPIE (2022).

[3] Atanasov, N., Le Ny, J., Daniilidis, K., and Pappas, G. J., "Decentralized active information acquisition: Theory and application to multi-robot slam," in [*2015 IEEE International Conference on Robotics and Automation (ICRA)*], 4775–4782 (2015).

[4] Charrow, B., Liu, S., Kumar, V., and Michael, N., "Information-theoretic mapping using cauchy-schwarz quadratic mutual information," in [*2015 IEEE International Conference on Robotics and Automation (ICRA)*], 4791–4798 (2015).

[5] Asgharivaskasi, A. and Atanasov, N., "Active bayesian multi-class mapping from range and semantic segmentation observations," in [*2021 IEEE International Conference on Robotics and Automation (ICRA)*], 1–7 (2021).

[6] Lindemann, L., Cleaveland, M., Shim, G., and Pappas, G. J., "Safe planning in dynamic environments using conformal prediction," (2022).

[7] Schlotfeldt, B., Tzoumas, V., and Pappas, G. J., "Resilient active information acquisition with teams of robots," *IEEE Transactions on Robotics* **38**(1), 244–261 (2022).

[8] Tzes, M., Bousias, N., Chatzipantazis, E., and Pappas, G. J., "Graph neural networks for multi-robot active information acquisition," (2022).

[9] Yang, P., Liu, Y., Koga, S., Asgharivaskasi, A., and Atanasov, N., "Learning continuous control policies for information-theoretic active perception," (2022).

[10] Mahmud, S. M. N., Nivison, S. A., Bell, Z. I., and Kamalapurkar, R., "Safe model-based reinforcement learning for systems with parametric uncertainties," *Frontiers in Robotics and AI* **8** (2021).

[11] Kreucher, C. M., Kastella, K. D., and III, A. O. H., "Information-based sensor management for multitarget tracking," in [*Signal and Data Processing of Small Targets 2003*], Drummond, O. E., ed., **5204**, 480 – 489, International Society for Optics and Photonics, SPIE (2004).