When to Ask for Help: Introspection in Multi-Robot Teams

Jen Jen Chung, Lauren Milliken, Geoffrey A. Hollinger and Kagan Tumer

Abstract— **This paper describes our preliminary work towards expertise-driven interactions between human operators and multi-robot teams. We consider scenarios where the mission objective changes suddenly due to some unexpected stimulus observed by the robot team. Policies trained to achieve the first objective may no longer be effectual. In such cases, external information from a mission expert, such as a human operator, can be sought to rapidly realign team behavior to achieve the new mission objective. We investigate the use of robot introspection via online expertise modeling to monitor team performance and seek human assistance when required. Preliminary results show that by framing the problem as a partially observable Markov decision process (POMDP), robots in the team are able to maintain an internal belief on their mission expertise and correctly identify when to request assistance.**

I. Introduction

One of the goals of automation is to reduce the workload on human operators such that each operator can manage a larger number of robots in the team. Currently, many successful human-robot collaborations in the field require a high human-to-robot ratio. Recent examples include the DARPA Robotics Challenge where large human teams interacted with a single robot [1], military use of robots in surveillance and exploration missions [2], as well as interplanetary exploration applications such as the Mars rovers, which are each supported by a team of human operators [3].

The respective robots in each of these examples are well-equipped and highly sophisticated in terms of both the onboard hardware and software. These tools ensure that they are capable of meeting the mission requirements as specified prior to execution. However, despite the best efforts of robot designers, unexpected changes to the internal system state or external environment are liable to occur that can, at best, hinder the robot from completing its task, and at worst, cause it to fail in spectacular fashion [4]. The role of human operators in these human-robot teams often involves monitoring the full system to identify when abnormal scenarios arise and implementing counter-measures on-the-fly to realign robot behavior when errors are detected. This places a significant load on the operators to be constantly attentive to the mission state and is a key hurdle to reducing the human-to-robot ratio in these types of collaborations.

Much of the existing work in human-robot interaction acknowledges the role of the human as a mission expert. Policy learning methods such as learning from demonstration (LfD) rely on humans to provide examples of successful trajectories through the state-action space that can be used as training data [5], [6]. LfD approaches typically require a significant number of demonstrations, however, the latest work in the field has aimed at reducing this load on the human expert [7]. Co-active learning approaches seek to incorporate human feedback online by selecting state-action pairs to present to an expert for approval or correction [8], [9], [10], [11]. Again, one of the major concerns in this area is reducing the number of queries necessary to learn an adequate policy. In a similar vein, techniques for shared autonomy seek a balance between human and robot control that best leverages the unique capabilities of each member within the collaboration [12].

All policy learning methods suffer from the same limitation, which is that a high level of prescience is required to incorporate all possible scenarios into the training data. Existing methods cannot easily detect scenarios that lie outside of the training data during online execution; having a robot or robot team respond to those scenarios effectively is an even greater challenge. Thus, human expertise is often sought to manage these mission-level risks.

The goal of this work is to tackle the first component of this challenge: automating the discovery of mission-level anomalies such that a multi-robot team can identify when human assistance is needed to realign team behavior. We introduce a robot introspection technique that draws from the user expertise modeling work of [12]. In our work, each robot within a multi-robot team uses a POMDP model to represent its own *policy expertise* based on the observed team performance score. Given the perceived policy expertise, robots can decide to either continue executing their current policy or request an update from a human expert.

We investigate the performance of our robot introspection algorithm on a multi-rover exploration domain where a team of ground robots are trained to explore and observe points of interest (POIs) in an initially unknown space. An additional layer of mission complexity is introduced by incorporating a *target POI*. Once the target POI is observed by any rover in the team, the goal of the mission changes such that all other POIs are no longer relevant and only the target POI must be observed. Preliminary results demonstrate that multi-robot teams capable of introspection via policy expertise modeling can correctly identify when the change in mission specifications occurs and request help from a human expert. Results also show that the system is averse to making superfluous requests, which is desirable for seamless human-robot interaction.

The authors are with the School of Mechanical, Industrial and Manufacturing Engineering, Oregon State University, OR, 97331, USA. {jenjen.chung, millikel, geoffrey.hollinger, kagan.tumer}@oregonstate.edu. This research was funded in part by NASA grant NNX14AI10G, NSF grant IIS-1317815, and support from PCC Structurals, Inc. and Oregon Metals Initiative.

In the following section we describe the robot introspection algorithm and define the parameters of the POMDP expertise model. Section III describes the simulation domain setup and online testing of the algorithm. An analysis of the results are provided in Section IV, while the implications of these results and avenues for further research are explored in Section V.

II. Robot Introspection

In order to decide if human assistance is required, each robot must maintain a belief of its current *policy expertise*, which is defined as the ability of the robot's control policy to achieve the current mission objectives. Since this value is not directly observable, we use a POMDP framework to model the expertise belief state, and update this belief based on the observed team performance during mission execution. We define the policy derived from the POMDP as the *query policy*, that is, the policy used by the robot to decide whether or not to query for human assistance. This is distinct from the robot *control policy*, which is trained according to the initial mission objectives.

A. POMDPs for Modeling Expertise

The POMDP tuple $\langle \mathscr{S}, \mathscr{A}, \mathscr{O}, T, Z, R, b_0, \gamma \rangle$ specifies the state, action and observation sets, as well as the conditional state transition and observation probabilities, reward function, initial belief and discount factor, respectively. The possible states, actions and observations in our expertise modeling experiments are shown in Table I. Although finer resolution of the policy can be achieved by further refining the set of observations, this level of discretization was experimentally found to be suitable for our test domain. Further discussion on how we defined the thresholds between each discrete observation is provided in Section III.

TABLE I
POMDP Parameters

Set	Elements
$\mathscr{S}$	Novice, Expert
$\mathscr{A}$	NoRequest, RequestAssistance
$\mathscr{O}$	LowReward, AverageReward, HighReward

For our experiments, we generated the POMDP query policy offline using hand-tuned values to define the transition and observation probabilities and the reward function. We used the Approximate POMDP Planning Toolkit (APPL) [13], which implements the SARSOP algorithm to approximately solve the POMDP for the query policy that maximizes the expected total reward [14].

B. Asking for Help

The output action of the query policy determines whether or not the robot requests assistance from a human expert. Ideally, the query policy is able to accurately identify when the robot control policy transitions from an Expert state to a Novice state and will execute the RequestAssistance action once the Novice state is detected.

During mission execution, the true transition between expertise states occurs when the mission objectives change. Although a conservative query policy that frequently requests assistance will likely lead to a rapid control policy update after the mission change, a large number of superfluous requests is undesirable. One of the main goals of this work is to streamline human-robot collaboration through expertise-driven interactions, thus we seek a query policy that initiates an interaction only when it is confident that the robot's own expertise is lacking and a human operator's advice is needed. To this end, the POMDP reward function is designed to not only encourage requesting assistance when the Novice state is detected, but also disincentivizes requesting assistance when in the Expert state.

This POMDP framework is generally applicable to single or multi-robot domains where a performance signal is available to indicate how well the robot or robot team is meeting the mission objectives. In the following section we will describe the multi-rover exploration domain in which we test our robot introspection algorithm.

III. Multi-rover Exploration Domain

In the multi-rover exploration domain [15], a set of ground robots (rovers) on a two-dimensional plane must coordinate in order to observe points of interest (POIs) scattered across the search space. Each POI has an associated value and an observability radius, r_{POI}. A rover that passes within the observability radius of a POI yields a reward which is weighted by the POI value and is inversely proportional to the distance that the rover is from the POI. The distance metric used in this domain is the Euclidean norm, bounded by a minimum observation value to prevent division by zero:

$$\delta(x,y) = \min\{||x-y||_2, \delta_{min}\}. \quad (1)$$

The objective of the rover team is to maximize the observation rewards over an episode, computed as:

$$G = \sum_j \frac{V_j}{\min_{i,t} \delta(L_j, L_{i,t})}, \quad \forall \delta(L_j, L_{i,t}) \leq r_{POI}, \quad (2)$$

where V_j is the value of POI j, L_j is the location of POI j, and $L_{i,t}$ is the location of the ith rover at time t. Although any rover may observe any POI during an episode, the system evaluation only takes into account the closest observation made for each POI across the entire episode.

Each rover is able to observe the locations of the POIs as well as the locations of the other rovers in the team. These range and bearing measurements are discretized into the four body-frame quadrants for each rover, resulting in an 8-dimensional state. The first four dimensions represent the condensed rover observations in each body-frame quadrant q of rover i,

$$s_{rover,q,i} = \sum_{i' \in \mathscr{M}_q} \frac{1}{\delta(L_{i'}, L_i)}, \quad (3)$$

and the last four represent the condensed POI observations in each quadrant,

$$s_{POI,q,i} = \sum_{j \in \mathscr{N}_q} \frac{V_j}{\delta(L_j, L_i)}. \quad (4)$$

Here, $\mathscr{M}_q$ and $\mathscr{N}_q$ represent the set of rovers and POIs in quadrant q, respectively.

A. *Offline Policy Training*

The control policy for each rover in the team is represented by a neural network with a single hidden layer. Given the input state, as described above, the output control action is a unit displacement in the *xy*-plane. The weights of the control policies are trained via a cooperative coevolutionary algorithm [16], with each rover agent using the Difference Evaluation Function [15], [17] to compute the fitness of each evolving control policy.

A team of five rovers was used in the following experiments. Each rover was initialized with a population of 15 random control policies which were trained over 1000 learning epochs. The training environments were 100×100 square units in size and each contained 25 randomly placed POIs. The POI values were also drawn from a uniform random distribution between $[1, 10]$. Rovers began each episode towards the center of the search space and executed their control policies for 100 1s timesteps.

The final population of trained control policies was stored, alongside the observed stepwise team performance for each team during the final training epoch. This latter data set was used to define the expected team performance thresholds of the POMDP expertise model. These were computed by analyzing the running average of the stepwise G, calculated from (2) for each timestep, over a fixed time window. Subsequently, appropriate thresholds were found for each discretized POMDP observation as functions of the running average $\bar{G}$:

$$o = \begin{cases} LowReward, & \text{if } \bar{G} < 0.01, \\ AverageReward, & \text{if } 0.01 \leq \bar{G} < 0.3, \\ HighReward, & \text{otherwise.} \end{cases}$$

B. *Online Execution*

Fifteen separate rover teams were constructed by sampling, without replacement, from each of the rover control policy populations. Each team was executed in a new testing environment for 1000s. During this execution phase, the stepwise team performance is broadcast to all rovers. Team performance is calculated according to (2) until the assigned target POI is observed by a member of the team. Once this occurs, the team performance metric changes to only reward successful observations of the target POI,

$$\hat{G} = \frac{V_{target}}{\min_{i,t} \delta\left(L_{target}, L_{i,t}\right)}, \quad \forall \delta\left(L_{target}, L_{i,t}\right) \leq r_{POI}. \qquad (5)$$

Each rover maintains a running average of the received team performance signals to update its current expertise belief state and select the appropriate NoRequest or RequestAssistance action according to the trained query policy. In this domain, the performance observation is common across all rovers, thus the belief propagation will also be identical across all members of the team. However, the framework we propose is decentralized and so the belief states of each rover can diverge if unique performance observations are used by each rover to update their belief.

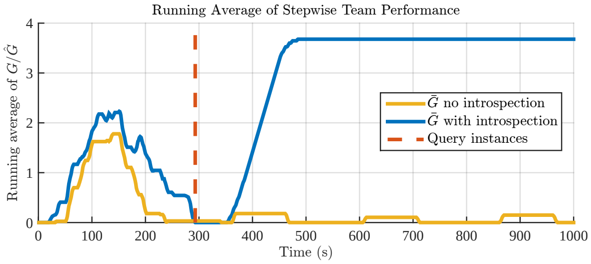


Fig. 1. Comparison of team performance with and without introspection and query capabilities. The dashed red line indicates instances when expert assistance is requested by the team. In this trial, as is common across the other trials, only a single request is sent during the entire mission.

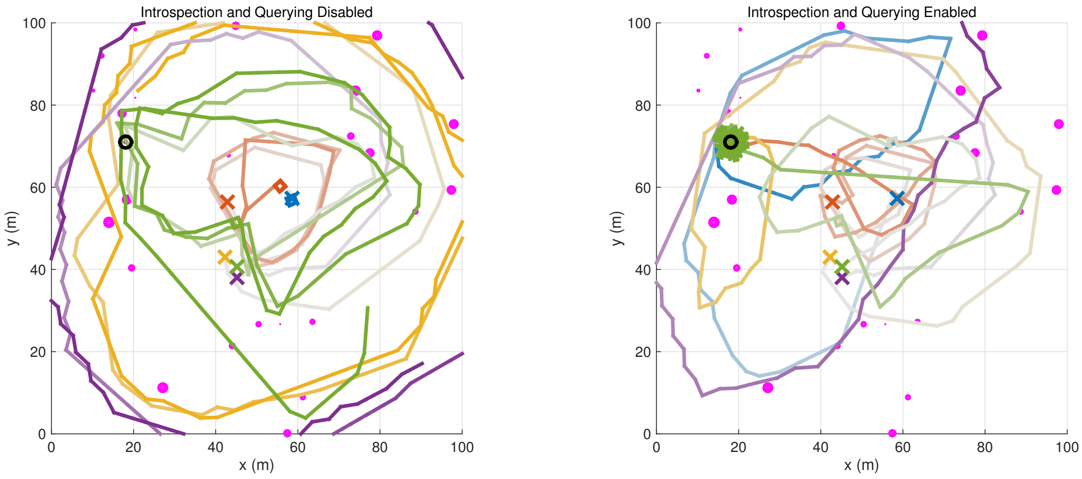
C. *Incorporating Human Feedback*

Once a request for assistance is submitted, the robot must wait for an external control policy update. Our robot introspection algorithm does not place restrictions on the form of this update, which will typically be driven by mission-specific criteria. For the multi-rover domain, this update arrives in the form of a new observation set that ignores all rovers and all POIs apart from the target POI. Since the trained control policies of each rover are reasonably generalizable to the new mission objective, the observation update serves to mask out all confounding elements in the space such that only the target POI is observable to each rover. As we show in the following results, this update is sufficient to realign the rover team control policies to the new mission objective.

IV. Results and Analysis

The running average of the stepwise team performance, as calculated using (2) and (5), for one experimental trial is shown in Fig. 1. This figure also plots the instances during mission execution when expert assistance was requested by the team. Since each rover uses the same $\hat{G}$ value to update its belief, consensus on when to request help is naturally achieved by the team. Across all 15 team trials, 9 teams requested assistance exactly once, 5 teams did not request assistance as the target POI was never successfully observed, and in only 1 trial were three requests sent during the mission. Further analysis of the rover trajectories during this latter trial showed that a number of the rovers were a significant distance from the target POI when it was first detected. Thus, the resultant delay in the team converging towards the target POI caused the running average to remain low, triggering additional requests for assistance.

Figure 2 compares an instance of the executed paths of two rover teams, one where introspection and querying are disabled (Fig. 2a) and the second where these properties are enabled in each rover (Fig. 2b). There is a clear difference between the rover team behavior across the two cases. When introspection and querying are disabled, the team is unable to internally identify the change in mission objectives when the target POI is detected. The rovers in this team will continue executing their obsolete policies with no way to actively adapt to the new circumstances. Compare this to the case where the rovers are able to assess their control policy



(a)

(b)

Fig. 2. Comparison of rover paths when introspection and querying is disabled 2a, and when it is enabled 2b. The initial rover locations are marked with × and the path color intensity increases over timesteps. The target POI is circled in black. When introspection is enabled, rovers in the team are able to identify when a control policy update is necessary to adapt to the new mission objective (i.e. converge at the target POI).

expertise. The trajectories of these rovers show that the team correctly identified when mission conditions changed and actively queried for a control policy update. The sharp change in heading along each rover path indicates the moment when the control policy update was received and the rovers began converging towards the target POI.

V. Discussion

The experiments conducted in this preliminary study demonstrate the ability of robot introspection for identifying misalignments between the current mission objective and the control policies of the multi-robot team. The experimental domain we used is reasonably simple and admittedly there are a number of possible policies that can be hand-designed to handle the change in mission objectives investigated here. However, the goal of our work is to generalize this paradigm–introspection for expertise-driven human-robot interaction–to larger and more complex scenarios where universally applicable control policies are not easily defined.

Future work will investigate how ideas from the reward shaping community can be incorporated to improve policy expertise observability. In particular, one aim is towards greater decentralization and customization for individual robots within the team. We hypothesize that this will allow for increased heterogeneity and coupling in the mission tasks. An associated avenue of future work is to investigate techniques for handling team consensus prior to requesting assistance. The ultimate goal is to develop a framework for identifying policy misalignments, performing inter-robot policy adjustment, and seeking external assistance from a human expert when deemed necessary.

References

[1] H. A. Yanco, A. Norton, W. Ober, D. Shane, A. Skinner, and J. Vice, “Analysis of human-robot interaction at the DARPA Robotics Challenge trials,” *Journal of Field Robotics*, vol. 32, no. 3, pp. 420–444, 2015.

[2] R. L. Nussbaum, “Changing the tooth-to-tail ratio using robotics and automation to beat sequestration,” *Air & Space Power Journal*, pp. 75–84, 2015.

[3] J. P. Grotzinger, J. Crisp, A. R. Vasavada, R. C. Anderson, C. J. Baker, R. Barry, D. F. Blake, P. Conrad, K. S. Edgett, B. Ferdowski, R. Gellert, G. J. B, M. Golombek, J. Gómez-Elvira, D. M. Hassler, L. Jandura, M. Litvak, P. Mahaffy, J. Maki, M. Meyer, M. C. Malin, I. Mitrofanov, J. J. Simmonds, D. Vaniman, R. V. Welch, and R. C. Wiens, “Mars Science Laboratory mission and science investigation,” *Space Science Reviews*, vol. 170, no. 1-4, pp. 5–56, 2012.

[4] IEEE Spectrum, “A compilation of robots falling down at the DARPA Robotics Challenge,” Youtube video https://www.youtube.com/watch?v=g0TaYhjpOfo, 2015, accessed on 13/2/2017.

[5] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, “A survey of robot learning from demonstration,” *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[6] C. G. Atkeson and S. Schaal, “Robot learning from demonstration,” in *International Conference on Machine Learning*, 1997, pp. 12–20.

[7] B. Akgun and A. Thomaz, “Simultaneously learning actions and goals from demonstration,” *Autonomous Robots*, vol. 40, no. 2, pp. 211–227, 2016.

[8] R. Goetschalckx, A. Fern, and P. Tadepalli, “Coactive learning for locally optimal problem solving,” in *AAAI Conference on Artificial Intelligence*, 2014, pp. 1824–1830.

[9] A. Jain, S. Sharma, T. Joachims, and A. Saxena, “Learning preferences for manipulation tasks from online coactive feedback,” *International Journal of Robotics Research*, vol. 34, no. 10, pp. 1296–1313, 2015.

[10] P. Shivaswamy and T. Joachims, “Coactive learning,” *Journal of Artificial Intelligence Research*, vol. 53, pp. 1–40, 2015.

[11] T. Somers and G. A. Hollinger, “Human–robot planning and learning for marine data collection,” *Autonomous Robots*, vol. 40, no. 7, pp. 1123–1137, 2016.

[12] L. Milliken and G. A. Hollinger, “Modeling user expertise for choosing levels of shared autonomy,” in *International Conference on Robotics and Automation*, 2017, to appear.

[13] National University of Singapore, “Approximate POMDP planning software,” http://bigbird.comp.nus.edu.sg/pmwiki/farm/appl/, 2014.

[14] H. Kurniawati, D. Hsu, and W. S. Lee, “SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces,” in *Robotics: Science and Systems*, 2008.

[15] A. Agogino and K. Tumer, “Analyzing and visualizing multiagent rewards in dynamic and stochastic environments,” *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 17, no. 2, pp. 320–338, 2008.

[16] S. G. Ficici, O. Melnik, and J. B. Pollack, “A game-theoretic and dynamical-systems analysis of selection methods in coevolution,” *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 6, pp. 580–602, 2005.

[17] A. Agogino and K. Tumer, “Efficient evaluation functions for multi-rover systems,” in *Genetic and Evolutionary Computation Conference*, 2004, pp. 1–11.