Decentralized Learning-based Planning for Multiagent Missions in the Presence of Actuator Failures

N. Kemal Ure, Girish Chowdhary, Yu Fan Chen, Mark Cutler, Jonathan P. How
Aerospace Controls Laboratory, MIT, Cambridge, MA, USA

John Vian
Boeing Research & Technology, Seattle, WA, USA

***Abstract*—We consider the problem of high-level learning and decision making to enable multi-agent teams to autonomously tackle complex, large-scale missions, over long time periods in the presence of actuator failures. Agent health, measured by the functionality of its subsystems such as actuators, can change over time in long-duration missions and may depend on environmental states. This variability in agent health leads to uncertainty that can lead to inefficient plans, and in some cases even mission failure. The joint learning-planing problem becomes particularly challenging in a *heterogeneous* team where each agent may have a different correlation between their individual states and the state of the environment. We present a learning based planning framework for heterogeneous multi-agent missions with health uncertainty that uses online learned probabilistic models of agent health. A decentralized incremental Feature Dependency Discovery algorithm is developed to enable agents to collaborate to efficiently learn representations of the uncertainty models across heterogeneous agents. The learned models of actuator failures allow our approach to plan in anticipation of potential health degradation. We show through large-scale planning under uncertainty simulations and flight experiments with state-dependent actuator and fuel-burn-rate uncertainty that our planning approach can outperform planners that do not account for heterogeneity between agents.**

I. Introduction

Unmanned Aerial Vehicle (UAV) technology is at a stage where single UAV surveillance and reconnaissance missions can be routinely performed under sufficient supervision. However, many of the UAV missions envisaged for the future, including disaster area monitoring, search and rescue, earth observation and mapping, reconnaissance, persistent search and track, and resupply could require that multiple UAVs, with possibly very different physical capabilities and payloads, collaborate over long time-periods and handle significant uncertainty about the mission and the environment [1]. Numerous researchers have developed autonomous planning and mission execution algorithms to address these challenges [2]. The resulting planning frameworks typically consist of at least two levels, the higher-level (possibly distributed) task-allocation or decision-making algorithm that assigns vehicles to tasks based on their capabilities, and the vehicle-level execution algorithms that include motion-control algorithms. In order to guarantee efficient execution and to maximize mission score, these algorithms must work in close harmony.

In particular, the higher level decision making algorithms need to take optimal decisions that take into account any available/online learned estimates of agent health states. This is a multiagent decision making and planning problem, which can be formulated in the Multi-agent Markov decision processes (MMDP) framework. The MMDP can be viewed as a collection of MDPs, one for each agent, that are coupled through common reward or transition models. However, it is well known that finding optimal solutions to MMDPs is computationally intractable as the number of agents increases [3], even when the agent state transition dynamics are perfectly known and approximate MDP solution techniques are applied [4]. It should be noted that the problem tackled here is concerned with optimally planning higher level behavior of the multiagent system to satisfy mission goals. This can be contrasted with the problem of controlling a multiagent formation *given* a higher level behavior (e.g., the literature on scalable consensus-based multiagent control [5]).

One way to tackle the MMDP is to decompose it into several agent level MDPs, and solve these MDPs by explicitly accounting for the coupling between them. This approach is referred to here as (Dec-MDPs) where each agent is responsible only for its own decision [6]. Ref. [7] proposed a Dec-MMDP formulation that decomposed the centralized problem into a set of individual approximate planning problems for each agent. The Dec-MMDP approach uses models

Ph.D. candidate in the Aerospace Controls Lab at MIT, ure@mit.edu
Postdoctoral Associate, School of Aeronautics and Astronautics, MIT, girishc@mit.edu
M.Sc. candidate in the Aerospace Controls Lab at MIT, chenyuf2@mit.edu
Ph.D. candidate in the Aerospace Controls Lab at MIT, cutlerm@mit.edu
Richard C. Maclaurin Professor of Aeronautics and Astronautics, MIT, jhow@mit.edu
Technical Fellow at Boeing R&T, john.vian@boeing.com

of teammate behaviors to enable predicting actions by other agents in the planning process. This formulation results in a large reduction in the computational complexity, which has been validated through flight tests for a Persistent Search and Track mission [8]. Refs. [3] extend the approach to improve the scalability to larger and more complex problems using a Group Aggregated Decentralized MMDP (GA-Dec-MMDP) in which agents approximate all of its teammates collectively with a single, reduced model. The GA-Dec-MMDP formulation reduces the computational complexity to be *polynomial* in number of agents opposed to exponential complexity of the existing approaches, which makes the GA-Dec-MMDP a potentially suitable modeling scheme for large-scale MMDPs. However, this claim has not been yet verified in a realistic mission scenario with multiple agents and realistic transition dynamics. One main contribution of this paper is to validate experimentally the GA-Dec-MMDP approach on a large scale persistent mission scenario that consists of several simulated and physical agents.

Furthermore, in any persistent multi agent mission, the health and motion-related capabilities of agents may change over time due to actuator degradation, faults, or addition of new modes of operation. The change in health and motion-related capabilities of agents is not accounted for by most algorithms for collaborative autonomous planning and mission execution [9]. These algorithms solve the task allocation and decision making problem assuming static vehicle health and capability models. If eventually there is a mismatch between the actual parameters and parameters assumed by the planner, this uncertainty may cause performance degradation and in certain cases may even lead to mission failure [10].

Researchers have showed that parameters of models of health related uncertainty [11] can be estimated online from the interactions with the environment and plans can be recomputed to improve the mission performance. However, these approaches typically assume that unknown parameters are distributed homogeneously across the state space, and/or they are shared commonly by all agents. The first assumption can be violated in situations where the agent health depends on the state of the environment in which it is operating, for example, the fuel burning rate depends on whether the agent is operating in a gusty area. Our previous work presented an efficient solution to tackle the first assumption by using Incremental Feature Dependency Discovery (iFDD) to efficiently incorporate health related state-dependent parametric uncertainties in MDP formulations [8, 12]. The second assumption is violated when considering heterogeneous teams of collaborating agents. Continuing the fuel burning example, while the rate of fuel burning may predictably change based on the location of the agent, the actual magnitude may differ for each agent. In such situations, the commonality in the operating environment means that the agents can collaboratively model the environment using a shared set of features, the heterogeneity in the agents causes the parameters related to these features are different. In this paper, we extend that work to account for heterogeneity in agent health models to tackle the second assumption. In particular, we propose a decentralized iFDD (Dec-iFDD) framework in which agents share features across the environment to efficiently learn own models of health dependent uncertainties.

In this work we consider actuator failures as the degradation of the vehicle capability to perform sub-tasks such as searching and tracking. A novel decentralized iFDD (Dec-iFDD) algorithm was proposed in [13] to collaboratively learn transition models for heterogeneous teams. The Dec-iFDD algorithm allows agents to leverage inter-agent communication to share iFDD discovered features with each-other while allowing for agent heterogeneity. In this paper we presents simulation results that consider more challenging scenarios where a large-scale team of heterogeneous agents need to learn and collaborate under both state-dependent actuator failure and uncertain fuel dynamics. In addition, we present indoor flight test results with mixed real/virtual agents and localized wind disturbances. The main significance of the results is that the considered scenarios are fairly large-scale compared to previous work on MMDP approaches, with an approximate state space size of 120^{10}, which shows the scalability of the developed planning and learning algorithms. Results indicate that the proposed framework is able to efficiently learn the failure/health degradation models and improve the missions performance by anticipating and preventing agent failures accordingly.

II. Problem Definition

This section outlines the objectives and constraints of typical persistent search and task (PST) mission. The formulation builds on previous discussions in Refs. [8]. The mission area is divided into three distinct regions geometrically. These regions are labeled as the *Base*, *Communication Relay* and *Task* areas, as depicted in Figure 1. Aerial agents (UAVs) start at the base area and travel from there to other regions for tasking and communication duties. As fuel depletes, or failures occur, these agents must return to base for refueling or repair. The communication area is a transition region between the base and tasking areas and requires an agent to act as a simulated relay link for communications to/from base. In the tasking area, the UAVs are assigned various tasks using a variant of Consensus-Based Bundle Algorithm (CBBA) [14, 15]. Tasks include searching and tracking of several target vehicles hidden among a number of civilian (e.g. neutral) vehicles, landing and drop-off tasks, and pursuit tasks.

The objective of the mission is to maximize the mission score by performing tasks in the tasking area while making sure a link with the base is formed by ensuring that a vehicle acts as relay in the communication relay zone. This forms a "chain" consisting of a spatially separated string of UAVs

Figures/mission.png

Fig. 1: PST Mission scenario: N autonomous agents cooperate to continuously complete tasks (such as surveillance, drop off etc.) in a specified tasking zone while maintaining constant communication with the base location. The goal is to maximize the number of successfully completed tasks. This behavior is to be persistently maintained even under sensor, actuator and battery health degradations.

which relay messages back and forth between the base and the tasking area [16].

In addition, there exists a number of different constraints on the mission. Each vehicle has limited fuel capacity and can therefore only operate for a limited amount of time in the communication or tasking areas. If a UAV runs out of fuel in either of these areas, it goes into a *crashed* state and cannot be recovered. The battery changing/charging station is located in the base area to refuel the UAVs and enable persistent operation. Moreover, each UAV has a non-zero probability of experiencing a sensor or actuator failure, which may limit their capabilities below that which is required to perform certain aspects of the mission. For instance, a vehicle with a failed sensor cannot perform search or tracking missions in the tasking area. However, it can act as a communication relay. Similarly, a vehicle with a damaged actuator cannot perform search, track or act as a communication relay - and therefore must return to base for repair. The qualitative description of each agent's stochastic health model is as follows. We assume that each agent is equipped with a sensor and multiple actuators. The sensor is required for surveillance and the actuators are required for mobility. At any point during the mission, an agent's sensor or one of the agent's actuators may "fail", with probability p_{sns}, and p_{act} respectively. Both sensor and actuator failures are repaired once the vehicle returns to the base location.

A. Multi-Agent Markov Decision Processes

~~MDPs are used here to model decision making scenarios~~ where state transitions have an aspect of randomness associated with them. An infinite-horizon, discounted multi-agent MDP (MMDP) is specified by the tuple $\langle n, \mathcal{S}, \mathcal{A}, P, g, \alpha \rangle$, where n is the number of agents, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P(s'|s,a)$ gives the transition probability from state s to state s' under action a, and $g(s,a)$ gives the cost (or reward) of taking action a from state s. The model P can either be known, or more realistically, depend on the random state with a parametrized prior probability distribution. Future costs are discounted by a factor $0 < \alpha < 1$. For the persistent UAV missions examined in this paper, α was set to $.99$ so that future costs still have significant importance for larger values of k. A policy of the MDP is denoted by $\pi : \mathcal{S} \to \mathcal{A}$, which maps states to actions. The problem is to minimize the cost-to-go function J_π over the set of admissible policies Π, starting from the initial state s_0:

$$\min_{\pi \in \Pi} J_\pi(s_0) = \min_{\pi \in \Pi} \mathbb{E}\left[\sum_{k=0}^{\infty} \alpha^k g(s_k, \pi(s_k))\right].$$

For notational convenience, the cost and state transition functions for a fixed policy π are defined as $g_s^\pi \equiv g(s, \pi(s))$, $P_{ss'}^\pi \equiv P(s'|s, \pi(s))$ respectively. The cost-to-go for a fixed policy π satisfies the Bellman equation [17]

$$J_\pi(s) = g_s^\pi + \alpha \sum_{j \in \mathcal{S}} P_{ss'}^\pi J_\pi(s') \quad \forall s \in \mathcal{S}, \tag{1}$$

which can also be expressed compactly as $J_\pi = T_\pi J_\pi$, where T_π is the (fixed-policy) dynamic programming operator.

B. Health Aware Planning Framework: An Overview

In a mission that is executed over long durations, the agent health can change over time. It is important to accommodate such changes, and replan to ensure mission success. In the Health Aware Planning (HAP) framework, the goal is to learn probabilistic, state dependent models of agent health dynamics, and use them to predict possible agent health degradation [18]. Planning in anticipation of health degradation has been shown to lead to improved performance. This proactive approach to planning outperforms an approach where policies are changed after failures occur (reactive approach) [18].

Unknown changes to the health dynamics of the agent can be modeled as a parametric uncertainty in the MMDP, which is defined by the tuple, $\langle n, \mathcal{S}, \mathcal{A}, P(\bar{p}), g, \alpha \rangle$, where $\bar{p} = [p_1, ..., p_n]$ is a vector of unknown mappings p_i of the form $\mathcal{S} \to [0,1]$ for each agent i. The transition model $P(\bar{p})$ is an explicit function of the unknown mappings in $\bar{p}$. Note that if p_i are known, then MMDP with parametric uncertainty reduces to description of a regular MMDP. Hence the underlying hypothesis of HAP is that the optimal planning problem for an MMDP with parametric uncertainty can be solved by first estimating the mappings p_i and then solving the corresponding MDP. This hypothesis is comparable to the certainty equivalence principle in indirect adaptive control [19], and boils down to estimating the state-dependent structure of the mapping p_i. However the exact way in which different states influence p_i is assumed to be unknown a priori. Furthermore, note that this formulation accommodates heterogeneity by allowing the mapping p_i to be different for each agent. The homogeneous case is recovered if all agents share the same mapping $\bar{p}$.

Figure 2 depicts the HAP framework used herein. The Dec-iFDD decentralized learning scheme is used to collaboratively learn the mapping $\bar{p}$ for each agent (see Section IV for details) using interactions with the environment. The GA-Dec-MMDP algorithm is used for decentralized re-planning using the online updated estimates of $\bar{p}$ (see Section III).

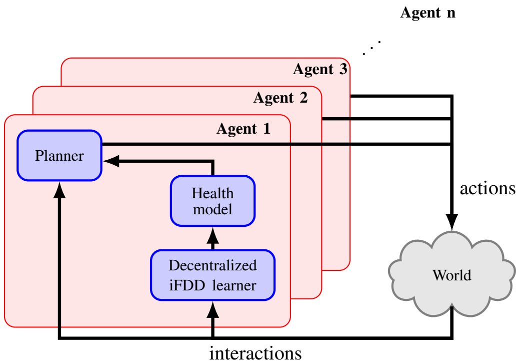

Fig. 2: Health Aware Planning architecture employs decentralized collaborative online learning (Section IV) to learn models of agent health dynamics. The learned online models are used for decentralized replanning in face of changes using the scalable GA-Dec-MMDP planner (Section III). Agents communicate their actions and representations of the uncertainty to each other.

III. Scalable Planning with Decentralized MMDPs

A flexible and powerful approach to modeling cooperative UAV missions is using the framework of Multi-agent Markov decision processes (MMDPs) [20–22]. Using MMDP models, it is theoretically possible to find an optimal policy through dynamic programming (DP) or reinforcement learning (RL) methods in a centralized fashion (if such a policy exists). However, it is well known that attempts to find optimal solutions to these problems through DP or RL techniques can quickly become intractable as the number of agents in the team increases. This problem is well-recognized in DP/RL community and have been approached by using approximation techniques [4, 17]. Another approach, specific to multi-agent problems, has been to investigate decentralized formulations of MDPs (Dec-MDPs) where each agent is responsible only for its own decision[6, 23–26]. Redding [7] proposed an Dec-MMDP formulation where the centralized problem is decomposed into a collection of individual approximate planning problems for each agent. The resulting formulation yielded a substantial computational complexity reduction and has been validated through flight tests for a PST mission. The approach was extended to improve the scalability to larger and more complex problems using a Group Aggregated Decentralized MMDP (GA-Dec-MMDP)[3]. The GA-Dec-MMDP formulation reduces the computational complexity to be *polynomial* in number of agents opposed to exponential complexity of the existing approaches, which makes it a suitable modeling scheme for large scale MMDPs.

The key idea behind the GA-Dec-MMDP approach is that the agent approximates all of its teammates collectively with a single, reduced model [3]. This model is generated using aggregation techniques on the joint state-action space of the teammates. The approach is motivated by the fact that in the decision making process of the individual agent, the agent might ignore a large portion of the state configuration of its teammates. For instance, agent can abstract the state of the rest of the team into a number of features such as the number of healthy agents in the tasking area and ignore the individual fuel and health configuration of the rest of the agents. The advantage of this formulation is that the growth of the size of the state space can be made polynomial in the number of agents, rather than exponential [3]. A full formulation of the GA-Dec-MMDP is available in Ref. [8].

IV. Decentralized Learning with Dec-iFDD

A. Learning State-Dependent Uncertainties

In any persistent multi agent mission, degradation, faults, or external effects could cause the health and motion-related capabilities of agents to change. In order to guarantee mission success, it is important to plan in anticipation of possible failures. Such proactive planning behavior has been shown to have much better mission performance than planning after changes have occurred (reactive planning) [3, 18, 27]. One way to enable proactive planning, is to learn online a probabilistic model of (possibly environment induced) uncertainties that affect vehicle health, such as the probability of actuator failure. In many cases, such a model will be state dependent [12]; for example, there may be a higher probability of health degradation in certain zones of the tasking area (see e.g. Figure 1). The key idea in the HAP framework described in Section II-B is to learn online such state-dependent probabilistic models efficiently. In a multi-agent collaborative missions, the commonality of the operating environment means that the agents can collaborate to improve the efficiency in modeling uncertainty using a shared set of features. However, learning a single model for all agents may not be ideal due to heterogeneity in the agents. In this section, we present a decentralized collaborative method to efficiently learn models of that allows for agent heterogeneity.

Let $p : \mathcal{S} \to [0, 1]$ denote the state dependent uncertainty function that needs to be learned to model the uncertainty in the underlying MDP. Since the state space $\mathcal{S}$ is discrete, p can be though as a $|\mathcal{S}|$ dimensional vector. When the dimension of $|S|$ is small, $p(s)$ for each state can be estimated separately for each state s by counting the state transitions observed from the environment [28]. This method is equivalent to maintaining a table across every state of the values of p. However when the dimension of $|s|$ is large, such as in the large-scale multi-agent UAV missions considered in this paper, such tabular representation of uncertainties becomes intractable. A linear function approximation techniques is much more tractable when modeling state dependent uncertainties. In this framework, for a given set of basis functions $\phi(s)$, the uncertainty is approximated by $p(s) \approx \bar{p}(s) = \theta^T \phi(s)$. One major challenge in this approach is the selection of an appropriate set of basis functions $\phi(s)$. In our earlier work, we showed that the incremental feature dependency discovery (iFDD) method can be used to adaptively generate a set of basis functions from an initial set of bases [29]. Combining iFDD with stochastic gradient descent on $\bar{p}(s)$ led to the development of SGD-iFDD algorithm [12], which is an efficient adaptive estimation algorithm for parameterized models of state dependent uncertainty.

B. Dec-iFDD Algorithm

In many realistic scenarios the model of the uncertainty is not only state dependent, but is also agent dependent. Let n represent the number of agents and let $p^i(s)$ represent the state dependent uncertainty functions that needs to be learned to model the uncertainty in the MDP of the i^{th} agent. A naive generalization of SGD-iFDD to decentralized setting can be obtained by running a separate SGD-iFDD for each UAV. In this setting, each UAV updates its own representation based on the individual observations. This approach does not leverage the ability of the agents to collaborate. In particular, even though the environment may affect each agent in a different way the underlying features used to represent the environment should be common, as the agents all operate in the same environment. The main idea behind the Decentralized iFDD (Dec-iFDD) algorithm presented in this subsection is increasing the efficiency in learning by allowing agents to share the iFDD discovered features with each other under communication constraints. The pseudocode of the Dec-iFDD algorithm is provided in Algorithm 1.

The line by line description of the Algorithm 1 is as follows, the algorithm takes number of agents n, the set of binary features for each agent f_{init} and the cap on shared features f_{cap} as the input. At each step, agents interact with the environment to receive observations (line 10), and then each agent applies the standard iFDD algorithm independent of each other to update it's current set of features f_i as well as the corresponding weights θ_i (line 11). When the current step is a sharing step, each agent sorts it's feature based on the weights (line 5), and then the first f_{cap}/n number of features with the largest weights are broadcast in the network (line 6 and 7). Then all the agents update their feature set with the broadcasted features and then set the weights for their new features (line 8).

There are two striking properties of the algorithm. First, note that the algorithm only allows agents to share features and not the weights each other. This is especially important for the heterogeneous teams, where the agents may share common features due to operating in the same environment, but each feature is weighted differently due to heterogeneity of the team. Secondly, the algorithm takes the communication limits into consideration with capping the total number of shared features in the network by the parameter f_{cap}, and forces agents to only share the more heavily weighed features in their representations at each sharing step. The implications of these two properties are shown in the following simulations.

Algorithm 1: Dec-iFDD Model Learning Algorithm

```
Input: Number of Agents $n$, Set Initial Binary Features $f_{init}$,
       Shared Feature Cap $f_{cap}$
Output: Estimated model $\hat{p}_i$ for each agent $i = 1, ..., n$
1  Initialize $f_i = f_{init}$ Initialize $\theta_i = 0$
2  foreach Step do
3      foreach Agent do
4          if Step = Sharing Step then
5              $f'_i \leftarrow$ Sort Features($f_i, f_{cap}, \theta_i$)
6              Broadcast($f'_i$)
7              $f^+ \leftarrow$ Listen()
8              $f_i \leftarrow f_i \cup f^+$, $\theta_i \leftarrow$ Update Theta($f_i$)
9          else
10             $s, a, s' \leftarrow$ Observe Transition
11             $f_i, \theta_i \leftarrow$ Expand Representation($s, a, s'$)
```

V. Simulation Results

The aim of this section is to verify the performance of the realization of HAP framework with Dec-iFDD and GA-Dec-MMDP algorithms in a large scale mission with a heterogeneous UAV team. In particular, we would like to analyze how well the assumption of a homogeneous team works under different scenarios that include teams with varying degrees of heterogeneity and we would like to eventually show that Dec-iFDD offers better performance improvement as the learning progresses. In addition, since the health of agents contribute to mission performance significantly, we would like to analyze the failure rates of the team under different planning approaches and show that the proposed framework leads to proactive policies that prevents these failures. Furthermore, we would like to examine the load of Dec-iFDD feature sharing process on the network to compare it against the load of sharing measurements directly.

For simulations, a large scale PST simulation with 10 collaborating UAVs was considered. It was assumed that the state-correlated actuator failure probability $p_a(s)^i$ and state-correlated nominal fuel burn rate for each agent i is unknown at the beginning of the mission, and must be learned through interactions with the environment. The baseline probability of actuator failure model that is used in the simulations is given at Table I and the baseline nominal fuel burning rate is given at Table II. State-dependency of the fuel burning rate was motivated from the possibility of having non-uniform wind in the mission environment and thus each region induces a different fuel burning rate. Table II shows the probability of incurring nominal fuel burn in the respective states. Non-nominal fuel burn rate is set as twice the nominal fuel burn rate.

The individual model for each UAV was obtained by randomly perturbing the baseline model within a specified

TABLE I: Baseline probability of actuator failure across different locations and fuel levels.

	Fuel Level	
Location	High Fuel Level	Low Fuel Level
Base	0.0	0.0
Communication	0.05	0.1
Surveillance	0.2	0.3

TABLE II: Baseline probability of nominal fuel rate across different locations and health states.

	Health		
Location	Healthy	Sensor Fail	Actuator Fail
Base	0.0	0.0	0.0
Communication	0.95	0.92	0.86
Surveillance	0.95	0.92	0.86
Surveillance (Windy)	0.90	0.80	0.75

range. We consider two simulations, where the perturbation range is set to 5% (almost homogeneous team) and 20% (heterogeneous team).

Figure 3 compares the cumulative cost performance of several instances of the Dec-iFDD algorithm with different feature sharing caps for almost homogeneous UAV team. In this case, it can be seen that as the cap on features to be shared is increased, the performance of the algorithm approaches that of the centralized algorithm. This result reinforces the intuition that the decentralized algorithm can do no better than a centralized one when the agent models of uncertainty are homogeneous. Figure 4 on the other hand, shows the performance of the Dec-iFDD algorithm for the same set of feature caps but in a network of collaborating UAVs with more heterogeneity (20% perturbation from a nominal model). In this case, it can be seen that planning using the output of the decentralized algorithm results in much less cumulative cost, because the planner is able to adjust the plan to suite the capability of each individual agent.

In order to demonstrate the proactive behavior of the resulting policy, two different mission metrics were averaged over 100 simulation runs and results are displayed on Table III. It is seen that the proposed planning-learning approach leads to policies with less number of total failures in the expense of commanding vehicles to return to base more frequently. This behavior is due to ability of Dec-iFDD to learn a specific model for each agent, opposed to centralized planner.

Finally, it is possible to compare feature sharing to an alternative approach in which all observations are shared by estimating the average number of shared features, in terms of communication cost. Let m be the size of a message that can be passed in the network, which corresponds to an single integer. Thus, each initial feature and a component of the state vector corresponds to messages of size m. The average number of features shared in the simulations were computed to be $\approx 4.2m$. Hence the total load on the network is $4.2f_{cap}m$, since he maximum amount of features shared in the network is limited to f_{cap} by Algorithm 1. In the considered scenario, each observed state transition corresponds to $84m$ sized messages. Hence if n agents share observations instead of features, they would need to send $84mn$ messages. For the simulation results presented above, number of agents $n = 10$, thus the load on network for pass-

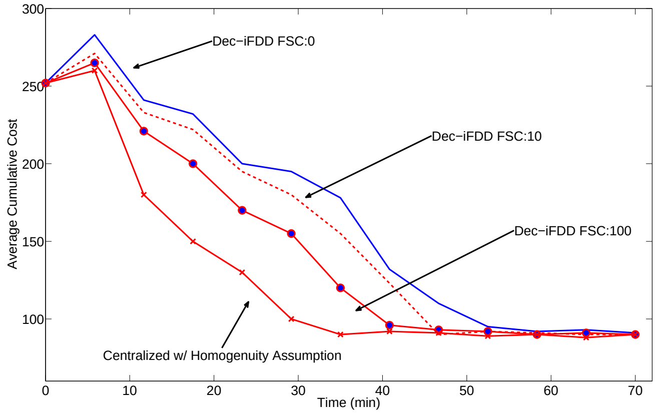

Fig. 3: Comparison of centralized approach versus decentralized approaches with different feature sharing caps (FSC) for the scenario where model perturbation is 5% from the nominal model. Results are averaged over 100 runs.

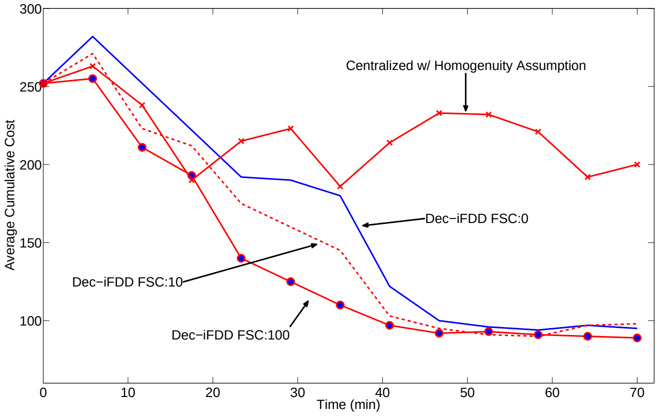

Fig. 4: Comparison of centralized approach versus decentralized approaches with different feature sharing caps (FSC) for the scenario where model perturbation is 20% from the nominal model. Results are averaged over 100 runs.

TABLE III: Evaluation of averaged mission metrics for centralized planner with homogeneity assumption and coupled GA-Dec-MMDP Dec-iFDD planner with FSC = 100.

Planner	# Failures	# Base Visits
Centralized Homogeneous	90.2	81.2
GA-Dec-MMDP with Dec-iFDD	55.1	125.4

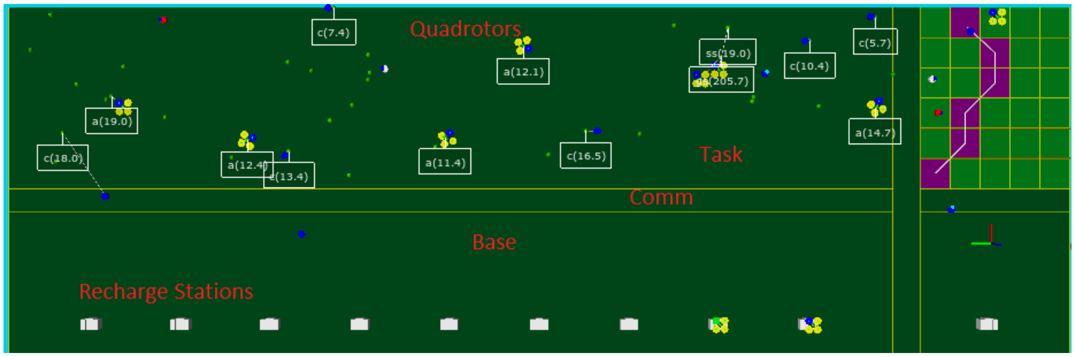

Fig. 5: Visual representation of the test environment showing Base, Communication and Tasking areas. The real quadrotor is constrained to operate in the RAVEN environment (right part of figure), and the rest of the team (9 quadrotors) operate in the simulated area to the left. The recharge station is shown at the bottom.

ing observations instead of features is around $840m$. Hence, even with a feature sharing cap of 100, sharing features require less load on network in terms of communication cost, compared to sharing observations. In addition, it is seen that size of observations scale linearly with number of agents, while the number of shared features is fixed for a constant f_{cap}. Hence the designer can tune f_{cap} solely based on the limits of the communication network, while direct sharing of measurements is limited by the size of the observations and the number of agents in the scenario.

VI. Experimental Results

This section presents the results of an hour long indoor flight test on a large-scale PST mission with 10 UAV agents, 10 recharge stations [8, 30], and several other ground agents. Of the 10 UAV agents, 9 are simulated (referred to as virtual agents) using physics based simulations, and 1 agent is physically present in the experiment area (referred to as real agent). The experiment is conducted at Aerospace Controls Lab (ACL) RAVEN testbed [31], which is equipped with a Vicon motion capture system. Custom designed quadrotor UAVs and an autonomous battery swap/recharge station [8, 30] are used.

The test environment is depicted in Fig. 5. The mission environment is separated such that real and virtual agents do not cross each other's operational spaces. The actuator failure probabilities are set to the values used for the heterogeneous team model described in the Section V, and the nominal fuel burn rate probabilities for all the virtual agents are set according to the heterogeneous team model described in the Section V. The nominal fuel burn probability of the real agent is influenced by the custom setup of vertical fans implemented in the experiment area (see Figs. 6 and 7), which is explained later in this section. The GA-Dec-MMDP planner described in Section III is used to coordinate the 10 agents across the base, communication and tasking areas, however the health models are not known to the planner, and must be estimated during the experiment. Objective of the experiment is to show that the Dec-iFDD algorithm succeeds in learning these models and the GA-Dec-MMDP planner can lower the mission cost progressively.

The tasking area for the virtual agents is populated with randomly generated tasks, which are represented by the colored dots in Fig. 5. Allocation of these tasks among virtual agents are handled by the CBBA task allocation algorithm [32].

The tasking environment for the real agent is shown in Figs. 6 and 7. As depicted in these figures, the real agent has two static target observation tasks (task 1 has a higher reward than task 2) and the agent must fly through a region with localized wind disturbances, created using two vertically aligned fans, in order to perform the tasks. However the locations and the magnitude of the localized wind disturbances are unknown to the agent a priori. The wind disturbances generated by the fans significantly impact the battery life of the quadrotor, as can be inferred from Fig. 8. The figure shows that flying under the wind (blue line) results in approximately 17% more current drawn from the battery compared to not flying under the wind (red line), thus it can be stated that flying under the wind corresponds to higher battery discharge rate, which results in shorter flight times before returning to base for battery swap. We would like to emphasize that impact of the wind on the battery life of the agent is not known to the planner in the beginning of the mission.

The tasking space of the real agent is separated into several grids, with each grid inducing a different unknown fuel burn rate based on whether the fan above the grid is active or not. Since the wind can appear and disappear during the mission, it is more appropriate to model their effect probabilistically rather than having a binary wind or no wind value for each grid. Hence, the probability of nominal fuel burn rate p_{fuel} is a function of the agent's grid location. Therefore, p_{fuel} can be treated as a state-dependent uncertainty (see Section IV) and is learned here using the iFDD algorithm. In the actual experiment, during the period where the real agent

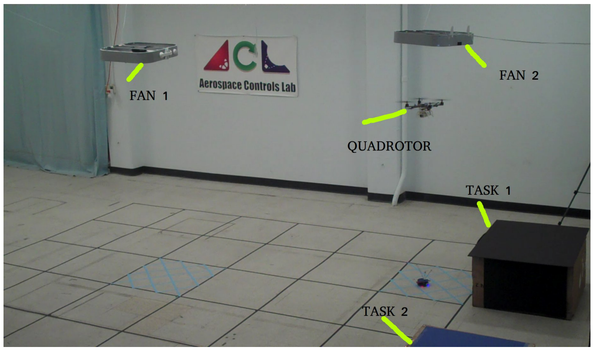

Fig. 6: View of the Task area of the real agent.

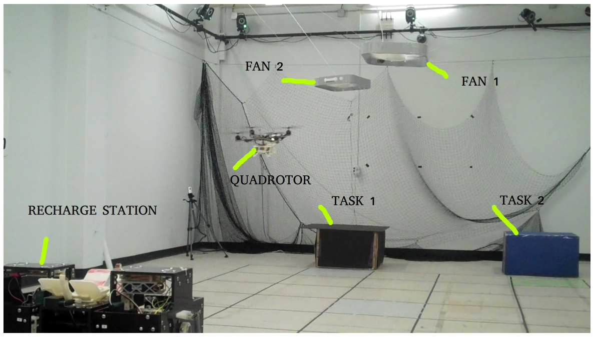

Fig. 7: View of the Task area of the real agent from Base.

is in the tasking area, the agent collects observations of battery usage at each grid it had traveled to. When the agent returns to the base, the iFDD algorithm processes this batch of observations to generate an estimate of $p_{fuel}(grid)$. This process is referred to as one iteration of wind learning. After each wind learning iteration, a dynamic programming algorithm is used to re-plan a trajectory that minimizes the based on the current estimate of $p_{fuel}(grid)$. The actual policy that the agent implements is randomly chosen from the optimal policy obtained from the DP algorithm or from a random policy, with the probability of choosing random actions reducing over time . This is an ϵ-greedy approach (see [28]), with $\epsilon = 0.2$, designed to encourage exploration.

In the beginning of the mission, only the Fan 1 is on and the fan on the the top of Task 1 (Fan 2) is turned off. Fig. 9 displays a sequence of selected trajectories for the real agent during different stages of the learning process. At the first wind learning iteration, the planner routes the agent towards the task with the higher reward (Fig. 9a), during agents 3^{rd} visit to the tasking area The Fan 2 is manually turned on unknown to the agent. It can be seen that the learning algorithm identifies the non-zero probability of experiencing wind for this grid (Fig. 9b). After taking more observations of the environment, the planner discovers that doing Task 1 corresponds to higher probability of non-nominal fuel burn rates and starts to route the agent towards Task 2 (Fig. 9c). After 10 wind learning iterations, planner converges to a trajectory that has the least probability of burning non-nominal fuels (Fig. 9d).

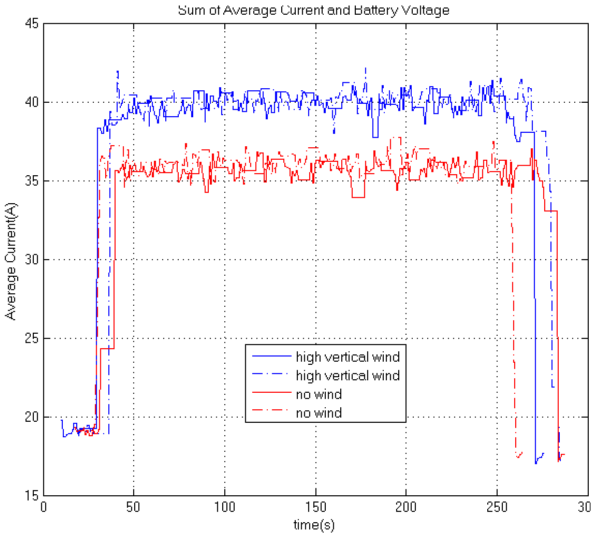

Fig. 8: Compares the current drawn from the battery while flying under the fan with current drawn while flying away from it.

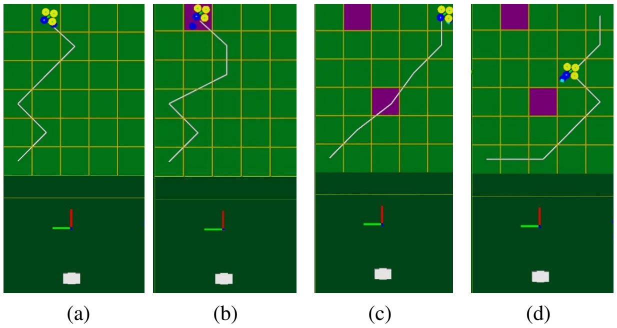
(a) (b) (c) (d)

Fig. 9: Selected trajectories of the real agent during the wind learning process. Purple grids, which are learned by the agent, correspond to grids with non-zero probability of experiencing wind. 9a: After 1 wind learning iteration, 9b: After 3 wind learning iterations ,9c: After 7 wind learning iterations, 9d: After 10 wind learning iterations

The overall planning performance of the whole team is presented in the average cumulative cost versus time plot on Fig. 10. These results are consistent with their simulation counterparts in Section V, in the sense that the Dec-iFDD algorithm learns the actuator failure and fuel burn models across the team and the GA-Dec-MMDP planner is then able to provide improved (lower) average cost. These flight experiment and simulation results verify the applicability of the GA-Dec-MMDP planner and the Dec-iFDD learner for large-scale UAV missions with unknown health models and heterogeneous team structure.

VII. Conclusions

We demonstrated through a large-scale simulation and flight experiment that the Group Aggregate Decentralized Multi agent Markov Decision Process (GA-Dec-MMDP)

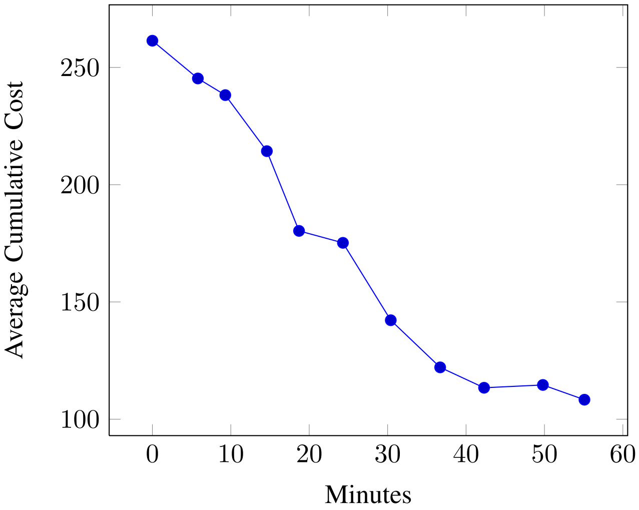

Fig. 10: Average cost of the mission as the learning and replanning progresses for the large-scale flight experiment

planning algorithm coupled with the decentralized Incremental Feature Dependency Discovery (Dec-iFDD) learning algorithm enables efficient distributed learning of actuator failure models by enabling the ability to re-plan online. Furthermore, we showed that when there is significant heterogeneity in agents, a planning architecture that periodically re-plans using different online estimated models of agent health outperforms an approach that assumes homogeneity in agents. These results establish the feasibility of using online learning based MMDP techniques for online decision making and planning in complex, multiagent, heterogeneous UAV missions that are executed over persistent duration. Future work will include a detailed formal analysis of computational complexity and communication overload of Dec-iFDD algorithm and the investigation of different types of actuator and sensor degradation models in both simulation and hardware setting.

Acknowledgments

This research was generously supported by Boeing Research & Technology. We thank Matthew A. Vavrina for his valuable insights and help in early phases of this work.

References

[1] Office of the Secretary of Defense. Unmanned aerial vehicles roadmap 2002-2027. Technical report, December 2002.

[2] R. Murray. Recent research in cooperative control of multivehicle systems. *ASME Journal of Dynamic Systems, Measurement, and Control*, 2007.

[3] J. D. Redding, N. Kemal Ure, J. P. How, M. Vavrina, and J. Vian. Scalable, MDP-based Planning for Multiple, Cooperating Agents. In *American Control Conference (ACC)*, June 2012.

[4] Lucian Busoniu, Robert Babuska, Bart De Schutter, and Damien Ernst. *Reinforcement Learning and Dynamic Programming Using Function Approximators*. CRC Press, 2010.

[5] Magnus Egerstedt and Mehran Mesbahi. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[6] Matthijs S. Spaan, Geoffrey J. Gordon, and Shlomo Zilberstein. Decentralized planning under uncertainty for teams of communicating agents. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 2006.

[7] J. D. Redding, T. Toksoz, N. Kemal Ure, A. Geramifard, J. P. How, M. Vavrina, and J. Vian. Persistent distributed multi-agent missions with automated battery management. In *AIAA Guidance, Navigation, and Control Conference (GNC)*, August 2011. (AIAA-2011-6480).

[8] Nazim Kemal Ure, Girish Chowdhary, Joshua Redding, Tuna Toksoz, Jonathan How, Matthew Vavrina, and John Vian. Experimental demonstration of efficient multi-agent learning and planning for persistent missions in uncertain environments. In *Conference on Guidance Navigation and Control*, Minneapolis, MN, August 2012. AIAA.

[9] George Vachtsevanos, Liang Tang, Graham Drozeski, and Luis Gutierrez. From mission planning to flight control of unmanned aerial vehicles: Strategies and implementation tools. *Annual Reviews in Control*, 29(1):101 – 115, 2005.

[10] Luca F. Bertuccelli. *Robust Decision-Making with Model Uncertainty in Aerospace Systems*. PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge MA, September 2008.

[11] H.S. Baik, H. S. Jeong, and D. M. Abraham. *Estimating Transition Probabilities in Markov Chain-Based Deterioration Models for Management of Wastewater Systems.* January/February, 2006.

[12] N. Kemal Ure, Alborz Geramifard, Girish Chowdhary, and Jonathan P. How. Adaptive Planning for Markov Decision Processes with Uncertain Transition Models via Incremental Feature Dependency Discovery. In *European Conference on Machine Learning (ECML)*, 2012.

[13] Nazim Kemal Ure, Girish Chowdhary, Yu Fan Chen, Jonathan P. How, and John Vian. Health-aware decentralized planning and learning for large-scale multiagent missions. In *Conference on Guidance Navigation and Control*, Washington DC, August 2013. AIAA.

[14] Sameera S. Ponda. *Robust Distributed Planning Strategies for Autonomous Multi-Agent Teams*. PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, September 2012.

[15] Luke Johnson. Decentralized Task Allocation for Dynamic Environments. Master's thesis, Massachusetts Institute of Technology, January 2012.

[16] C. Dixon and E.W. Frew. Maintaining optimal communication chains in robotic sensor networks using mobility control. In *Proceedings of the 1st international conference on Robot communication and coordination*. IEEE Press Piscataway, NJ, USA, 2007.

[17] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. I-II, 3rd Ed.* Athena Scientific, Belmont, MA, 2007.

[18] Nazim Kemal Ure, Girish Chowdhary, Jonathan P. How, Mathew Vavarina, and John Vian. Health aware planning under uncertainty for uav missions with heterogeneous teams.

In *Proceedings of the European Control Conference*, Zurich, Switzerland, July 2013. (to appear).

[19] A.S. Morse. Towards a unified theory of parameter adaptive control. ii. certainty equivalence and implicit tuning. *Automatic Control, IEEE Transactions on*, 37(1):15 –29, jan 1992.

[20] M. Valenti, B. Bethke, J. P. How, D. P. de Farias, and J. Vian. Embedding Health Management into Mission Tasking for UAV Teams. In *American Control Conference (ACC)*, pages 5777–5783, New York City, NY, 9-13 July 2007.

[21] B. Bethke, J. P. How, and J. Vian. Group health management of UAV teams with applications to persistent surveillance. In *American Control Conference (ACC)*, pages 3145–3150, Seattle, WA, 11-13 June 2008.

[22] B. Bethke, J. P. How, and J. Vian. Multi-UAV Persistent Surveillance With Communication Constraints and Health Management. In *AIAA Guidance, Navigation, and Control Conference (GNC)*, August 2009. (AIAA-2009-5654).

[23] Sven Seuken and Shlomo Zilberstein. Formal models and algorithms for decentralized decision making under uncertainty. *Autonomous Agents and Multiagent Systems*, 17(2):190–250, 2008.

[24] Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of Operation Research*, 27(4):819–840, 2002.

[25] Claudia V. Goldman and Shlomo Zilberstein. Decentralized control of cooperative systems: Categorization and complexity analysis. *Journal of Artificial Intelligence Research*, 22:143–174, 2004.

[26] Victor Lesser Ping Xuan and Shlomo Zilberstein. Communication decisions in multi-agent cooperation: Model and experiments. In *Proceedings of the fifth international conference on Autonomous agents*, 2001.

[27] J. Redding, A. Geramifard, A. Undurti, H. Choi, and J. How. An intelligent cooperative control architecture. In *American Control Conference (ACC)*, pages 57–62, Baltimore, MD, July 2010.

[28] R. Sutton and A. Barto. *Reinforcement Learning, an Introduction*. MIT Press, Cambridge, MA, 1998.

[29] Alborz Geramifard, Finale Doshi, Joshua Redding, Nicholas Roy, and Jonathan How. Online discovery of feature dependencies. In Lise Getoor and Tobias Scheffer, editors, *International Conference on Machine Learning (ICML)*, pages 881–888. ACM, June 2011.

[30] Tuna Toksoz. Design and Implementation of an Automated Battery Management Platform. Master's thesis, Massachusetts Institute of Technology, August 2012.

[31] J. P. How, B. Bethke, A. Frank, D. Dale, and J. Vian. Real-time indoor autonomous vehicle test environment. *IEEE Control Systems Magazine*, 28(2):51–64, April 2008.

[32] H.-L. Choi, L. Brunet, and J. P. How. Consensus-based decentralized auctions for robust task allocation. *IEEE Transactions on Robotics*, 25(4):912–926, August 2009.