2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

Health Aware Planning Under Uncertainty for UAV Missions with Heterogeneous Teams

N. Kemal Ure, Girish Chowdhary, Jonathan P. How, Matthew A. Vavrina, John Vian

Abstract— **In large-scale persistent missions, the vehicle capabilities and health often degrade over time. This paper presents a Health Aware Planning (HAP) Framework for long-duration complex UAV missions by establishing close feedback between the high-level planning based on Markov Decision Processes (MDP) and the execution level learning-focused adaptive controllers. This feedback enables the HAP framework to plan by anticipating the failures and reassessing vehicle capabilities after the failures. This proactive behavior allows for efficient replanning to account for changing capabilities. Simulations for a 4 UAV target tracking scenario is presented to demonstrate the effectiveness of the proactive replanning capability of the presented HAP framework.**

I. Introduction

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 will require multiple UAVs, with possibly diverse capabilities, to collaborate over long time-periods with significant uncertainty about the environment [1] . The development of autonomous planning and mission execution algorithms for collaborating UAVs is an active area of research [2, 3]. The control algorithms for such missions typically have 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, both these algorithms have to work in close harmony.

In particular, the mission-planning algorithm needs to be aware of each UAV's capabilities, which are related to vehicle health, and can be captured through a *capability model* that relates health to capabilities. Therefore, if the health of an agent changes over time as a result of failures, environmental effects such as winds, addition/removal of new agents, or the introduction of new modes of operations, the available set of agent capabilities also changes. Marier et al. considered the problem of health-aware coverage control with small UAVs in which they accounted for changes in the sensing and communication hardware health and complete loss-of-vehicle scenarios [4]. However, while hardware health and vehicle activity can be detected to a large extent by monitoring diagnostic tools provided with sensing and communication electronics, no such straight forward tools are available for assessing changes in motion-related capabilities of a UAV. This work extends the prior work by Marieri et al. by considering more general health and capability models applicable to a wider variety of UAV missions developed in Section II.

This change in health and motion-related capabilities of agents is not accounted for by most algorithms for collaborative autonomous planning and mission execution (see e.g. [3, 5, 6]). These algorithms solve the task allocation and decision making problem independently of the vehicle level controllers using, for example, the framework of Markov decision processes, and rely on vehicle level controllers to execute the mission [7–9]. The underlying assumption in these approaches is that the vehicle health and capabilities do not change during the duration of the mission is restrictive, and cannot be justified in real-world scenarios that involve persistent missions, or operation in adversarial/contested environments [1, 10].

One possible solution to account for changing agent health is to attempt to solve a large MDP that consists of the vehicle motion, capabilities, and the mission goals. However, this approach is computationally intractable [11, 12], even when the transition dynamics are perfectly known and approximate MDP solution techniques are applied [13–15]. Researchers have thus turned to adding adaptation to the vehicle level controllers to mitigate the effect of execution-level uncertainties. In particular adaptive controllers, in the framework of Model Reference Adaptive Control, have been developed and flight-tested to accommodate environmental disturbances, modeling uncertainties, and vehicle damage [7, 16–18]. The main idea is that the adaptive controller can attempt to maintain a consistent set of performance and capabilities by overcoming, at least to a limited extent, changes in the vehicle health. However, while the adaptive controller can delay the need to replan at the mission level, it cannot completely eliminate it. To enable rapid replanning, a feedback needs to be established between the vehicle level adaptive controller and the higher level planner to provide

PhD. Candidate, Aerospace Controls Laboratory, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA ,USA ure@mit.edu

Post-doctoral Associate, Laboratory for Information and Decision Systems, MIT, girishc@mit.edu

Richard C. Maclaurin Professor of Aeronautics and Astronautics, Aerospace Controls Laboratory (Director), MIT, jhow@mit.edu

Research Engineer at Boeing Research and Technology, Seattle, WA, USA matthew.vavrina@boeing.com

Technical Fellow at Boeing Research and Technology, Seattle, john.vian@boeing.com

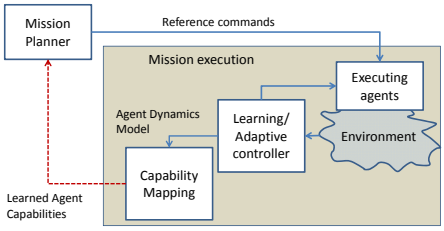

Fig. 1: The Health Aware Planning architecture presented in this paper with explicit feedback between the high-level planning algorithms and low-level concurrent-learning adaptive controllers. The mission planning algorithm uses own-models of agent capabilities learned through individual agents' concurrent learning adaptive controllers and proposed efficient MDP solution techniques to proactively adjust to changing agent health and capabilities.

the planner with estimates of changes in vehicle capabilities. The main idea of the Health Aware Planning (HAP) approach presented in this paper is to provide this feedback by estimating vehicle capabilities using the adaptive controller's model of the vehicle health.

Note that a similar approach was explored by Redding et al. [19] who proposed to use the internal parameters of the adaptive controller to form an estimate of the vehicle health model to enable HAP. However, the MRAC framework employed in that work (based on [20]) was designed to favor suppressing instantaneous tracking error, and hence had weak guarantees of convergence of parameters. In fact most popular MRAC methods focus on instantaneously suppressing the tracking error and not on learning the uncertainty [21–26]. Therefore the use of such adaptive control methods leads to very limited useful information that could be used for replanning.

In contrast, the Concurrent Learning (CL) adaptive framework developed by Chowdhary and Johnson [27, 28] is an example of a learning-focused, extensively flight-tested [28–30] direct adaptive control framework that provides strong guarantees on *simultaneous learning of uncertain parameters and closed-loop system stability*. The extensions to the CL framework outlined in this paper enable us to integrate the adaptive control and the mission planner to a much **tighter degree** than possible in Ref. [19]. This improvement occurs because: 1) our extensions to CL guarantee convergence of parameters under much milder conditions such as only requiring to know sign of the B matrix, and therefore ensure that sufficiently good capability estimates can be provided to the planner in finite time, increasing the reliability of the architecture; and 2) all uncertain parameters of both A and B matrix are estimated allowing for more general vehicle dynamical health models to be built allowing better estimate of vehicle capabilities.

This paper presents a multi-agent planning-learning HAP architecture for multi-UAV complex missions in uncertain environments (see Figure 1). Our approach can handle far more general models of health and capability changes than Redding et al. [19]. This is enabled by the learning-focused concurrent learning adaptive control algorithm that explicitly enables changes in the vehicle capabilities to be fed-back to an MDP based planning algorithm by mapping learned own-models to capabilities. Furthermore, we leverage efficient exploration and adaptive value-function approximation algorithms to efficiently solve the associated MDP online. The main contributions of this paper are:

- A comprehensive Health Aware Planning (HAP) framework is introduced that enables a feedback between mission planning and vehicle-level adaptive control algorithms through online adaptive-controller learned models of agent health and capabilities. This feedback enables health-aware replanning.
- The main difficulty in solving higher level planning MDP online is ensuring efficiency in the exploration of the state-action space for finding a good solution. An efficient exploration method that drives the exploration into unexplored parts of the state-action space using the notion of "knownness" of state-action pairs [31–33] was implemented. The Incremental Feature Dependency Discovery (iFDD) adaptive function approximator [34] is used to efficiently approximate the knownness function.
- The concurrent learning adaptive control architecture [28] is extended to accommodate uncertainty in the control assignment matrix (B matrix).

II. Problem Definition

A. Vehicle Health, Capability and Task models

1) Health Models: Health here refers to a measure of wellness/functionality of a vehicle, which depends on the wellness/functionality of vehicle components/subsystems, such as airframe, actuators, and electronics. The health of the vehicle can be inferred by directly monitoring the health of individual components. Furthermore, since degradation/changes in some of these components may affect the dynamics of the vehicle, health of the vehicle may also be inferred from the vehicle's dynamic response. Let n_{veh} denote a fixed number of collaborating vehicles, and let n_{health} denote main components of interest of each vehicle. Similar to aircraft handling qualities rating scales [35], a discrete scale is used to represent the health of each vehicle. In particular, the scalar $h^i_j \in 0, 1, \ldots, 10$ represents the health of the j^{th} subsystem of i^{th} vehicle, where $i = 1, 2, \ldots, n_{veh}$, $j = 1, 2, \ldots, n_{health}$, and $h^i_j = 0$ if a component is completely non-functional ($h^i_j = 10$ if a component is fully functional). Note that this choice of scale is arbitrary. Let $h^i = [h^i_1, h^i_2, \ldots, h^i_{n_{health}}]$ represent the health vector of the i^{th} vehicle. Both due to internal effects such as fuel consumption and external effects such as disturbances from the environment, the health of a vehicle changes during the mission. Let k represent the discrete time index, then the transition between $h^i_j(k)$ and $h^i_j(k+1)$ is modeled

probabilistically as, $\mathcal{P}\left(h_j^i(k+1)=v \middle| h_j^i(k)=u\right) = p_{u,v}^{h_j^i}$ where $p_{u,v}^{h_j^i} \in [0,1], u,v = 0,1,\ldots,10$ are the transition probabilities. In this work it is assumed that these transition probabilities are available to the designer, if that is not the case, algorithms similar to [36] can be used to estimate these transition probabilities online.

The UAV health is characterized here into four different categories:

- Structural/Actuator health: Represents damage & functionality status of structural & actuator components of a UAV. Failures/faults such as wing damage, rudder damage, blade damage and mainframe damage in included in this category. These health components impact motion related abilities, such as target pursuit.
- Sensor health: Represents the functionality of the sensor-hardware such as imaging sensors, video recorders, sound detectors and infrared sensors.
- Communication: Represents the functionality of the communication hardware of the UAV, such as its wireless modem. Health of this component affects the distance and reliability of UAV communication.
- Fuel: Represents fuel or power consumption quantity of the UAV per transition which impacts all abilities.

2) Capability Ratings: A capability of an agent is defined as a measure on how well an agent can perform a task, such as target pursuit, or steadily capturing images. A reasonable planning algorithm would assign vehicles to appropriate tasks that best match the vehicle capabilities. Let n_{cap} represent the number of capabilities for each vehicle. Each capability is rated by a capability rating $c_j^i \in 0,1,2,\ldots,10$, $i = 1,2,\ldots,n_{veh}$, $j = 1,2,\ldots,n_{cap}$. Here, $c_j^i = 0$ means that vehicle is incapable of performing a task requiring that capability and $c_j^i = 10$ means that vehicle is completely capable of executing that task. The capability rating depends on the health h^i of each vehicle. In a generic form, this relationship can be written implicitly as $c_j^i = \zeta^i(h^i)$, where h^i is the health vector and ζ^i is the function that maps the health vector to capability rating (see subsection IV-B). Note that since health is a dynamic quantity, so are the capability ratings.

3) Task Models: A *task* is defined as sub-goal of the overall mission that can be completed by a single or multiple agents. Tasks have associated uncertainties, such as uncertain target dynamics in a tracking task or uncertain weather conditions in a precision drop-off task. Therefore, the success of completion of a task is a random variable that depends on the agent's capability. Let n_{task} be the total number of tasks defined for the mission. Here, the probability that agent i successfully completes a task l is modeled as an Bernoulli random variable χ_l^i, where χ_l^i takes the value 1 with probability p_l^i and takes the value 0 with probability $1-p_l^i$, and $p_l^i \in [0,1]$ is the probability of completing the task successfully. Probability of task completion p_i depends on capability c_i and a linear model is chosen to represent this dependency, $p_l^i = S^l \sum_{j=1}^{n_{cap}} s_j^l c_j^i$, where s_j^l are scalar weights that represent relative importance of each capability for the task, and S^l is a normalization constant, which ensures that $p_l^i \in [0,1]$.

It is assumed that a task is considered to be complete when any one of the agent succeeds in completing it. Hence, although task probabilities are defined per agent, if more than one agent is assigned to the same task, the total probability of task completion increases.

Table I illustrates the relative impact factor of different capabilities upon task completion probabilities, for some common UAV tasks. This table highlights how some tasks require combinations of capabilities, for example, target tracking require both trajectory tracking capability and image sensing capability.

B. Health Aware Planning Problem

The generic HAP problem considered here requires routing UAVs through different parts of the environment to accomplish different tasks such that number of cumulative completed tasks is maximized. Furthermore, it is assumed that UAV health can degrade over time, and the health-aware planner needs to determine when to optimally return UAVs to base for repair. The environment can be visualized by imagining a graph whose vertices are task *zones* and edges depict zones between which the UAV can transition (see Fig. 2). Each zone has a mixture of tasks to be completed. UAVs start in the *base* zone with full health and capability, furthermore, whenever a UAV returns to this zone, its health is assumed to be fully restored. UAV health degrades due to effects such as fuel consumption, wear, external effects such as winds and gusts, and damage. Hence the objective of the planner is to find a balance between routing UAVs to zones based on their health/capabilities and recalling them back to base for health restoration so that the mission can potentially be indefinitely extended.

C. Mission Level MDP Formulation

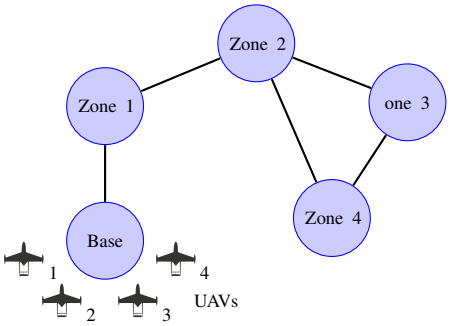

Fig. 2: Health Aware Planning Problem on a graph based structure. UAVs needs to be allocated between zones, where each zone has a list of dynamic tasks. UAVs are repaired and refueled when they return to Base.

TABLE I: Relative impact factors of capabilities on task completion rates. 0: no impact, 1: medium impact, 2: high impact. These relative impact factors can be used as a guideline to setup the capability ratings c_j^i used in Eq. II-A.3.

Task /Capability	Fly Straight to Waypoint	Trajectory Tracking	Relay Information	Perform Precision Landing	Perform Agile Maneuvers	Collect Images/Videos
Loiter	2	1	0	0	0	0
Track Ground Target	2	1	0	0	0	1
Track Aerial Target	1	2	0	0	2	1
Payload Drop-off	2	0	0	2	0	0
Communication Relay	2	0	2	0	0	0
Search/Reconnaissance	1	0	0	0	0	2

1) State Space: Let n_{zone} be the number of zones in the mission including the base (Zone_0). The local state space of each UAV consists of its location (zone) z^i, fuel f^i and health vector h^i. Let U^i be the local state space of i^{th} UAV, Thus the state vector can be written as, $u^i = [z^i, f^i, h^i]^T$, where $z^i \in \{\text{Base}, \text{Zone}_1, \text{Zone}_2, \ldots, \text{Zone}_{n_{zone}-1}\}$, $f^i \in \{0, 1, 2, \ldots, f^i_{\max}\}$ and $h^i = h^i = [h^i_1, h^i_2, \ldots, h^i_{n_{health}}]$. The overall state space for the mission is the space generated by the Cartesian product of local spaces of each UAV,$\mathcal{S} = \prod_{i=1}^{n_{veh}} U^i$

2) Action Space: The actions for each UAV are denoted $a^i \in 0, 1, \ldots, n_{zone} - 1$, where $a^i = k$ means move UAV i to Zone_k. Given that UAV i is in zone Zone_j, available actions are given as,$A^i_{z_j} = \{m \in 0, 1, \ldots, n_{zone} - 1 | g_{i,m} = 1\}$, where $g_{i,m} = 1$ if zone i and m are connected.

Note that we define $g_{j,j} = 1$, so a UAV can always be commanded to stay in its current zone. The overall action space is the Cartesian product of all the action spaces of individual UAVs

$$\mathcal{A} = \prod_{i=1}^{n_{veh}} \prod_{j=1}^{n_{zone}} A^i_j \tag{1}$$

.

3) Transition Model: The location of the UAVs evolves deterministically as

$$z_i(k+1) = \begin{cases} z_{a_i}, a_i \in A^i_{z_i(k)}, \text{if } f^i(k) \neq 0 \\ z_i(k) \text{if } f^i(k) = 0 \end{cases} \tag{2}$$

. Thus a UAV always travels to the zone it is commanded to as long as it has non-zero fuel. Otherwise, UAV is out of fuel and stays in its current location indefinitely (crashed). The fuel dynamics evolve according to,

$$f_i(k+1) = \begin{cases} f^i(k) - f^i_{nom}, \text{WP } p_{fnom}(z^i(k), z^i(k+1))) \\ f^i(k) - f^i_{nnom}, \text{WP } 1 - p_{fnom}(z^i(k), z^i(k+1))) \\ f_i(k+1) = 0 f^i(k) = 0 \\ f_i(k+1) = f_{\max}(i) z^i(k) = 0 \end{cases} \tag{3}$$

where WP refers to *with probability*. Here $f^i_{nom}(Zone_p, Zone_q)$ is the nominal fuel consumption rate of the UAV and $f^i_{nnom}(Zone_p, Zone_q)$ is the non-nominal fuel consumption rate while transitioning from $Zone_p$ to $Zone_q$ or i^{th} UAV. The last line in Eq. 3 indicates that UAV is re-fueled at base.

4) Reward Model: Reward at each step is simply the weighted sum of successfully completed tasks at that step. Let $d_l \geq 0, i = 1, \ldots, n_{task}$ denote the positive reward obtained for completing the l^{th} task and $e_l \geq 0, i = 1, \ldots, n_{task}$ be the penalty for not completing the task successfully. The total reward at time k is

$$r_k = \sum_{l=1}^{n_{task}} \sum_{i=1}^{n_{veh}} d_l \chi^i_l + e_l (1 - \chi^i_l). \tag{4}$$

III. Planning with On Trajectory Approximate Dynamic Programming

A. On Trajectory Asynchronous DP using Knowledge Functions

Approximate Dynamic Programming Methods [14], apply linear function approximation by using basis functions $\phi_i : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}, i = 1, ..., n$, such that $Q(s,a) \approx \tilde{Q}(s,a)\theta^T \phi(s,a)$ where $\theta \in \mathbb{R}^n$ is the weight vector and $\phi = [\phi_1(s,a), \phi_2, ..., \phi_n(s,a)]$. Approximate Value Iteration (AVI)[13] methods updates the weight θ as, $\theta \leftarrow \theta + \delta(s,a)\phi(s,a)$, where $\delta(s,a) = Q^+(s,a) - Q(s,a)$ and $Q^+(s,a) \leftarrow \sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'}[\mathcal{R}^a_{ss'} + \gamma \max_{a'} Q(s', a')]$. Note that AVI only requires storage of n parameters. The value $\delta(s,a)$ is a measure of how far is the estimated value from the optimal value function and usually referred to as *Bellman Error*. Since sweeping through all possible values of (s,a) becomes intractable in large problems, an practical alternative is to perform Bellman updates to trajectories with fixed policy L_π sampled from the model, which results in the Trajectory Based Value Iteration Algorithm (TBVI).It can be showed that, if all state-action pairs are visited infinitely often, TBVI converges to the optimal value function asymptotically [37].

As discussed in Section I, incorporation of the idea of *knownness* [31] to the planning framework, has proven to be an efficient exploration scheduling technique. Knownness of a state-action pair $k(s,a) \in [0,1]$ is defined as $k(s,a) = \max(1.0, \#\text{ (s,a) is visited } / m_{known})$ where $m_{known} \in \mathbb{Z}$ is the number of times (s,a) should be visited before the agent can reliably predict the transition and reward

model related to (s,a). Although m_{known} is a designer selected parameter, bounds on the order of this parameter is well-understood [38], and these bounds can be used to guide parameter tuning process. The exploration-exploitation dilemma is handled internally by augmenting the value function with the knownness function, $\hat{Q}(s,a) = (1 - k(s,a))\frac{r_{max}}{1-\gamma} + Q(s,a)$, where r_{max} is the maximum one step reward that can be obtained in the MDP. Note that this augmentation results in giving higher values to state-action pairs with low knownness, hence encouraging exploration.

B. Approximate Dynamic Programming and Exploration using Incremental Feature Dependency Discovery

The previous subsection pointed out that both the value $Q(s,a)$ and knownness $k(s,a)$ functions must be approximated to enable tractable planning in large-scale problems. However determining a good set of features ϕ is a complex problem in itself, and is a topic of considerable interest to the DP and RL communities [14]. One way to address the problem of finding good set of features is to use adaptive function approximation techniques that constructs the set of features ϕ online based on the performance of the planner. Recently developed Incremental Feature Dependency Discovery (iFDD) algorithm [34] has proven to be an efficient algorithm for online approximation of value functions. The basic idea in iFDD is to expand a set of binary features based on the correlation of active features with some error metric, such that feature conjunctions with the highest error is added to set of features at each step. Here iFDD representation is applied to approximate both $Q(s,a)$ and $k(s,a)$, using the Bellman Error $\delta(s,a)$ as the error metric that drives the expansion. Idea behind applying Bellman Error to expand the knownness function relies on the intuition that (s,a) pairs with high Bellman Errors should be explored more frequently. This idea is reminiscent of prioritized sweeping [39], however the idea of approximating the knownness function by an adaptive representation is completely novel. The deatiled description of the planning algorithm TBVI-iFDD with Approximate Knownness can be found in [40].

IV. Online Health Monitoring and Capability Estimation using Concurrent Learning Adaptive Control

Avionics related capabilities of can be assessed using the diagnostic offered by those subsystems. However, estimating the motion-related capabilities of the UAV is difficult, because they must be inferred through the UAV's dynamic response. To counter this problem, online generated dynamic health models are used to characterize vehicle capabilities after failures. This section provides the development of agent-level concurrent learning adaptive controllers that simultaneously estimate UAV dynamic health models for use in capability mapping modules (see Fig. 1) while ensuring the UAV stays stable after faults.

A. Simultaneous Model Estimation and Stabilization using Concurrent Learning Adaptive Control

Dynamics of each UAV can be cast in the form of a switching linear dynamical system as follows,$\dot{x}^i(t) = A_{\sigma^i(t)}x^i(t) + B_{\sigma^i(t)}u^i(t), i = 1,..,n_{veh}$, where $\sigma^i \in [1,...,q]$ describes the mode of the system, and $A^i_\sigma \in \mathbb{R}^{n\times n}$, $B^i_\sigma \in \mathbb{R}^{n\times n}$. It is assumed that modes of the system evolve at discrete time, that is $\sigma_i(t)$ is always constant on a fixed time interval Δt_{switch}. Switched linear system formulation enables modelling the impact of actuator failures in system dynamics, thus the switching time for the system refers to failure times in the mission. We assume that initial model of the system $(A_{\sigma(0)}, B_{\sigma(0)})$ is known, however the model after failure at time t_{fail}, $(A_{\sigma(t_{fail})}, B_{\sigma(t_{fail})})$ is unknown. Objective of the control law is to stabilize the system after failure and simultaneously estimate the new model $(A_{\sigma(t_{fail})}, B_{\sigma(t_{fail})})$. This is achieved using concurrent learning adaptive control [27, 28]

1) Concurrent Learning Model Estimation: Let $\dot{x} = Ax + Bu$ represent a linear dynamical system with unknown (A,B) matrices, note that we have dropped the switching signal subscripts to facilitate exposition. Assume that the state of the system $x(t)$ and the input signal $u(t)$ is available or can be constructed from the measurements. Let $(\hat{A},\hat{B})$ represent the estimate of (A,B) and let $\dot{\hat{x}} = \hat{A}x + \hat{B}u$. Define error dynamics as, $\epsilon(t) = ([\hat{A},\hat{B}] - [A,B])[x(t),u(t)]^T = \hat{x} - \dot{x}$. Objective of the model estimation algorithm is to drive $\epsilon(t) \to 0$ asymptotically. Let $x_i, u_i, i = 1,..,p$ be the data points recorded online at times t_i. Concurrent learning model estimation updates are given as follows,

$$\dot{\hat{A}}(t) = -\Gamma_A[x(t)\epsilon(t) - \sum_{j=1}^{p} x_j\epsilon_j] \tag{5}$$

$$\dot{\hat{B}}(t) = -\Gamma_B[u(t)\epsilon(t) - \sum_{j=1}^{p} u_j\epsilon_j], \tag{6}$$

where $\Gamma_A, \Gamma_B > 0$. The following theorem can be proven using arguments in [27, 41]

Theorem 1: Assume that the control signal $u(t)$ is exciting over a finite interval and that the data points for concurrent learning are selected online using the singular value maximizing Algorithm (Algorithm 1 from [41]), then, $\hat{A} \to A$ and $\hat{B} \to B$ exponentially fast.

Remark 1: Note that model estimation law 5 requires the knowledge of $\dot{x}$. Mühlegg et. al showed that if a noisy estimate of $\dot{x}$ is available the adaptation law is guaranteed to be uniformly bounded under some mild additional assumptions [42]. Also note that, the $\epsilon(t)$ feedback term on the model update law can be dropped, and estimates of ϵ_j can be improved using a fixed point smoother [28].

2) Concurrent Learning Adaptive Control with unknown A and B Matrix: Let $\dot{x}_{rm} = A_{rm}x_{rm} + B_{rm}r$ represent the dynamics of the reference model, where $r(t)$ is the reference signal. Assume A_{rm} is Hurwitz and P is the solution to Lyapunov equation $A_{rm}^T P + PA_{rm} + Q = 0$, where $Q \in$

$\mathbb{R}^{n\times n}$ is a positive definite matrix. Let the control law be of the form

$$u(t) = K^T(t)x(t) + K_r^T(t)r(t) \tag{7}$$

, where $K \in \mathbb{R}^{n\times m}$ and $K_r \in \mathbb{R}^{1\times m}$. Assume that matching conditions hold, i.e. there exits as K^* and K_r^* such that,$A+BK^{*T} = A_{rm}$,$BK_r^{*T} = B_{rm}$. After substituting the control law to the system equation and performing algebraic manipulations, the following form of the error dynamics $e(t) = x - x_{rm}$ is obtained, $\dot{e} = A_{rm}e + B\tilde{K}^T x + BK_r^T t$. Objective of the controller is to update K, K_r such that that error dynamics $e(t) = x - x_{rm}$ and weight error dynamics $\tilde{K} = K - K^*, \tilde{K_r} = K - K_r^*$ are asymptotically stable. When the control assignment matrix B is available to the designer, it has been shown in [28], that concurrent learning satisfies this objective with exponential decay rate without needing persistency of excitation. In this section we will extend this work to deal with case where B matrix is unknown. Let $\hat{B}$ be a fixed estimate of the B matrix which is available to the controller. Let x_i, r_i be the i^{th} data point recorded online, define the error variables $\hat{\epsilon}_{K_j}$ and $\hat{\epsilon}_{K_r}$ as, $\hat{\epsilon}_{K_j} = \hat{B}^{-1}(\dot{x_j} - A_{rm}x_j - B_{rm}r_j - \hat{B}\epsilon_{K_{r_j}})$,$\hat{\epsilon}_{K_{r_j}} = K_r^T r_j - \hat{B}^{-1}B_{rm}r_j$. The concurrent learning weight update laws are given as,

$$\dot{K} = -\Gamma_x[xe^T P\hat{B} + \sum_{j=1}^{p} x_j\hat{\epsilon}_{K_j}^T] \tag{8}$$

$$\dot{K_r} = -\Gamma_r[re^T P\hat{B} + \sum_{j=1}^{p} r_j\hat{\epsilon}_{K_{r_j}}^T] \tag{9}$$

The following theorem states that as long as the estimate $\hat{B}$ is close enough to B the system $[e, \tilde{K}, \tilde{K_r}]$ is bounded.

Theorem 2: Consider the control law in 7 and weight update laws in 8-9. Assume that the control signal $u(t)$ is exciting over a finite interval and that the data points for concurrent learning are selected using the singular value maximizing Algorithm (Algorithm 1 from [41]). In addition assume that $||B - \hat{B}||$ is bounded and $sign(B) = sign(\hat{B})$ and pairs (A, B) and $(A, \hat{B})$ are controllable. Then the system $[e, \tilde{K}, \tilde{K_r}]$ is bounded.

Proof: Can be found in [40]. ■

Corollary 1: In addition to the assumptions of Theorem 1, if $\tilde{B} = 0$, i.e. the control allocation matrix B is known to the controller, then the system $[e, \tilde{K}, \tilde{K_r}]$ is exponentially stable, that is the tracking error and weight error dynamics converge to zero exponentially fast.

Proof: This case is proved in [28], Theorem 1. ■

3) Safe Model Estimation Using Switched Control: The model estimation method described in section IV-A.1 and the control law described in section IV-A.2 can be combined to build a switched control-model estimation algorithm that concurrently learns the model and stabilizes the UAV after failures. We assume that there is a separate health monitoring system that signals to controller that a failure is occurred. Basic idea of the switched control algorithm is to use the control law developed in section IV-A.2 to keep the system bounded without the knowledge of the system after failure, while the parameter estimation law developed in section IV-A.1 estimates the new system model in the background. Once the new system model is estimated, controller switches to newly learned model to stabilize the system. This process is described in detail on [40]. Note that, boundedness of the system during the model estimation process is guaranteed by Theorem 2 and since the model estimation law is proven to be exponentially fast by Theorem 1, as long as time between two failures is big enough, safe model estimation algorithm is guaranteed to converge to the new model after the failure in finite time. After the algorithm terminates, Corollary 1 implies that the control law 7 will stabilize the system.

B. Capability Mapping

It is necessary to transfer the knowledge contained in the estimated dynamic model of an agent after a failure to the planning layer as a change in agent capabilities. Hence it is necessary to develop mappings that takes the model $(\hat{A}, \hat{B})$ and transfers them to capability ratings (see e.g. Table I). Here we give examples of several metrics related to motion capabilities of the system,

- Translational Reachability ($X_{\Delta t}$): Represents how far UAV can travel with a bounded input, in a fixed time interval Δt.
- Attitude Reachability ($M_{\Delta t}$): Represents the attitude envelope of the UAV, that is set of orientations that can be reached from the origin within fixed time ΔT, and bounded inputs.
- Attitude Precision ($M_{r\Theta}$): Represent the norm of the tracking error associated with tracking a reference attitude trajectory $\Theta(t)$.
- Disturbance Rejection (W_r): Represent the norm of the tracking error associated with tracking a reference trajectory $r(t)$ under the effects of disturbances induced by the environment.

Computation of these reachability sets can be pursued as a optimization problem [43], however such approaches are only shown to be tractable for low dimensional dynamical systems. We generate trajectories using the models of the system provided by the adaptive controller to approximate these reachability sets instead of performing optimization over the reachability sets of the system. Once these sets are determined, they can be translated into capability metrics in Table I.

V. Simulation Results

To verify the properties of the presented framework, a multi-agent target tracking scenario with a teams of targets and agents with heterogeneous capabilities is investigated. Mission consists of a team of $n_{uav} = 4$ UAVs with different target tracking capabilities, and $n_{zone} = 10$ zones including the base. As an example of heterogeneous task requirements, each zone consists of a target tracking task with different agility level, which corresponds to Track Ground Target

and Track Aerial Target tasks in Table I. Note that it is straightforward to add other types of tasks requiring different types of capabilities.

Presented framework is compared with two alternative approaches. First planner, labeled as "Non-proactive, No capability re-assessment" (NPR-NC), does not include the health dynamics at the MDP model other than fuel dynamics, and does not replan after UAV failures. Second approach, labeled as "Non-proactive, Capability re-assessment enabled" (NPR-C), also does not include any health dynamics/failure models but updates the motion capabilities of the UAVs based on the new models provided by the adaptive control algorithm. This situation corresponds to having an adaptive controller onboard, but not establishing feedback between the planner and the controller. The third approach is the presented HAP framework, where the health dynamics are included in the MDP model and capability re-assessment is enabled. Cumulative reward obtained by these 3 different approaches averaged over 30 runs is compared in Fig. 3. The results show that the NPR-NC and NPR-C approaches have collected better cumulative reward on average compared to the HAP framework for the first 500 steps. However, as the mission progresses, and UAVs start to experience more failures, the performance of the planner with no capability re-assessment starts to degrade due to the mismatch between model and actual capabilities of UAVs. This indicates that the HAP framework generated the policy that performs more successful tasks compared to other two planners over the long run, even though its initial performance may be relatively conservative.

TABLE II: Mission metrics for different planning schemes.

Planner	# Failures	# Base Visits	Agents per Target	# Completed Tasks
NPR-NC	80.8	678.2	1.1	43.2
NPR-C	71.3	665.1	1.2	67.4
HAP	14.3	923.4	1.9	96.2

To gain insight in the difference between the policies generated by the different planners, three different metrics were evaluated, see Table II. The *failure* metric counts the average number of structural/actuator and sensor failures that occurred during the mission. Note that the failures are not explicitly penalized in the reward function (see subsection II-C.4), however since UAV capabilities/health are strongly coupled with probability of task completion (see subsection II-A.3), this metric serves as a good indicator of mission performance. The second metric we examine is the average number of returns to base, which indicates how frequently vehicles are called back to base for failure repairs and re-fuel. The *agents per target* metric captures the average number of agents a planner assigns to target. Finally, *number of completed tasks* tasks metric, counts the number of successfully completed target tracking tasks, which is the major contributor to the reward function.

From Table II it is clear that HAP approach resulted in much lesser number of failures compared to NPR-NC and NPR-C approaches, which is also supported by higher number of visits to the base. This is due to inclusion of probabilistic health and capability models in the HAP architecture, which allows planner to anticipate the failures and take the necessary actions beforehand. Furthermore, since the MDP utilized by the HAP framework includes failure models, the policy tends to assign nearly two agents to a single target in order to increase the chance of completing the task successfully in presence of failures. The importance incorporating a tight coupling feedback between control and planning layers (see Introduction) is also seen on comparison of NPR-NC and NPR-C approaches, where NPR-C incorporates this feedback by by capability reassessment and provides higher number of completed tasks compared to NPR-NC, as seen on the last column of Table II. The combination of the aforementioned proactive behaviours demonstrated by the HAP policy and the capability reassessment are the underlying reasons why HAP approach yields higher number of completed tasks and accumulates more reward than the alternative approaches, NPR-NC and NPR-C in Fig. 3.

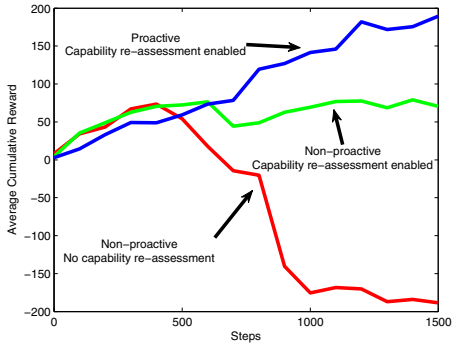

Fig. 3: Average cumulative reward obtained by different planning schemes.

VI. CONCLUSIONS

This paper proposed a Health Aware Planning Framework for persistent complex multi-UAV missions where likely health-degradation and failures of individual UAVs are accounted for. Our framework establishes a close feedback between the high-level planning based on Markov Decision Processes (MDP) and the execution level learning-capable adaptive controllers. This feedback enables the HAP framework to anticipate failures at the planning level, and both proactively and reactively replan to account for changing capabilities. The feedback was enabled by using UAV health-parameter estimates from a concurrent learning adaptive controller to form dynamic UAV capability models which can be used for re-planning to account for failures and degradation. The presented HAP framework was tested on

a large-scale ($\approx 10^{10}$ state-action pairs) target tracking scenario using a novel on-trajectory exploration algorithm, and demonstrated to sustain mission performance by reducing the number of failures and re-assessing UAV capabilities. The presented HAP framework was shown to outperform planning frameworks that lack health awareness and/or capability re-assessment feedback. These results clearly demonstrate that establishing a feedback between higher-level planning and execution-level controllers is crucial to guarantee safety and functionality of persistent complex multi-agent missions.

ACKNOWLEDGMENTS

This research was generously supported by Boeing Research & Technology in Seattle, WA.

REFERENCES

[1] Office of the Secretary of Defense. Unmanned aerial vehicles roadmap 2002-2027. Technical report, December 2002. URL http://www.acq.osd.mil/usd/uav$_$roadmap.pdf.

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

[3] E. Semsar-Kazerooni and K. Khorasani. Multi-agent team cooperation: A game theory approach. *Automatica*, 45(10):2205–2213, 2009.

[4] J.-S. Marier, C. A. Rabbath, and N. L andchevin. Health-aware coverage control with application to a team of small uavs. *Control Systems Technology, IEEE Transactions on*, PP(99):1, 2012. ISSN 1063-6536. doi: 10.1109/TCST.2012.2208113.

[5] Sameera S. Ponda, Luke B. Johnson, and Jonathan P. How. Distributed chance-constrained task allocation for autonomous multi-agent teams. In *American Control Conference (ACC)*, June 2012. URL http://acl.mit.edu/papers/ACC2012_ChanceConstrainedCBBA_final_submitted.pdf.

[6] 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. ISSN 1367-5788. doi: 10.1016/j.arcontrol.2004.11.002. URL http://www.sciencedirect.com/science/article/pii/S136757880500009X.

[7] Suresh Kannan, Girish Chowdhary, and Eric N. Johnson. *Handbook of Unmanned Aerial Vehicles*, chapter Adaptive Control of Unmanned Aerial Vehicles - Theory and Flight Tests. Springer, 2012).

[8] Jonathan P. How, Emilio Frazzoli, and Girish Chowdhary. *Handbook of Unmanned Aerial Vehicles*, chapter Linear Flight Contol Techniques for Unmanned Aerial Vehicles. Springer, 2012.

[9] Guillaume JJ Ducard. *Fault-tolerant flight control and guidance systems*. Springer, 2009.

[10] 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.

[11] Joshua D. Redding. *Approximate Multi-Agent Planning in Dynamic and Uncertain Environments*. PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge MA, February 2012.

[12] 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.

[13] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*, volume I–II. Athena Scientific, PO Box 391, Belmont, MA 02178, 2007.

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

[15] Alborz Geramifard, Nazim K. Ure, Stefanie Tellex, Girish Chowdhary, Nicholas Roy, and Jonathan P. How. A tutorial on linear function approximators for dynamic programming and reinforcement learning. *Foundations and Trends in Machine Learning*, 2012 (submitted).

[16] A. Calise, N. Hovakimyan, and M. Idan. Adaptive output feedback control of nonlinear systems using neural networks. *Automatica*, 37(8):1201–1211, 2001. Special Issue on Neural Networks for Feedback Control.

[17] Eugene Lavretsky and Kevin Wise. Flight control of manned/unmanned military aircraft. In *Proceedings of American Control Conference*, 2005.

[18] Girish Chowdhary, Eric N. Johnson, Rajeev Chandramohan, Scott M. Kimbrell, and Anthony Calise. Autonomous guidance and control of airplanes under actuator failures and severe structural damage. *Journal of Guidance Control and Dynamics*, 2012. in-press.

[19] J. Redding, Z. Dydek, J. P. How, M. Vavrina, and J. Vian. Proactive planning for persistent missions using composite model-reference adaptive control and approximate dynamic programming. In *American Control Conference (ACC)*, pages 2332–2337, June 2011.

[20] E. Lavretsky. Combined/composite model reference adaptive control. *Automatic Control, IEEE Transactions on*, 54(11):2692 –2697, nov. 2009. ISSN 0018-9286. doi: 10.1109/TAC.2009.2031580.

[21] Gang Tao. *Adaptive Control Design and Analysis*. Wiley, New York, 2003.

[22] N. Hovakimyan, B. J. Yang, and A. Calise. An adaptive output feedback control methodology for non-minimum phase systems. *Automatica*, 42(4):513–522, 2006.

[23] Chengyu Cao and N. Hovakimyan. Design and analysis of a novel adaptive control architecture with guaranteed transient performance. *Automatic Control, IEEE Transactions on*, 53(2):586 –591, march 2008. ISSN 0018-9286. doi: 10.1109/TAC.2007.914282.

[24] Tansel Yucelen and Anthony Calise. Derivative-free model reference adaptive control. *AIAA Journal on Guidance, Control, and Dynamics*, 34(8):933–950, 2012. AIAA paper number 0731-5090, doi: 10.2514/3.19988.

[25] Nhan Nguyen, Kalamanje Krishnakumar, John Kaneshige, and Pascal Nespeca. Dynamics and adaptive control for stability recovery of damaged asymmetric aircraft. In *AIAA Guidance Navigation and Control Conference*, Keystone, CO, 2006.

[26] M. Steinberg. Historical overview of research in reconfigurable flight control. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 219(4):263–275, 2005.

[27] Girish Chowdhary and Eric N. Johnson. Concurrent learning for convergence in adaptive control without persistency of excitation. In *49th IEEE Conference on Decision and Control*, pages 3674–3679, 2010.

[28] Girish Chowdhary, Maximillian Muhlegg, Tansel Yucelen, and Eric Johnson. Concurrent learning adaptive control of linear systems with exponentially convergent bounds. *International Journal of Adaptive Control and Signal Processing*, 2012. doi: 10.1002/acs.2297. URL http://onlinelibrary.wiley.com/doi/10.1002/acs.2297/abstract.

[29] Girish Chowdhary and Eric N. Johnson. Theory and flight test validation of a concurrent learning adaptive controller. *Journal of Guidance Control and Dynamics*, 34(2):592–607, March 2011.

[30] Girish Chowdhary, Tongbin Wu, Mark Cutler, Nazim Kemal Üre, and Jonathan How. Experimental results of concurrent learning adaptive controller. In *AIAA Guidance, Navigation, and Control Conference (GNC)*, Minneapolis, MN, August 2012. AIAA. URL http://acl.mit.edu/papers/chow_GNC12_conc_applications.pdf. Invited.

[31] Ronen I. Brafman and Moshe Tennenholtz. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research (JMLR)*, 3:213–231, 2001.

[32] Langford J. Kakade S., Kearns M. Exploration in metric state spaces. In *International Conference on Machine Learning (ICML)*, 2003.

[33] Ali Nouri and Michael L. Littman. Multi-resolution exploration in continuous spaces. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 1209–1216. MIT Press, 2009.

[34] 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. ISBN 978-1-4503-0619-5.

[35] B. Etkin and Reid L. D. *Dynamics of Flight, Stability and Control*. John Wiley and Sons, 1996.

[36] 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.

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

[38] S. P. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Journal of Machine Learning Research (JMLR)*, 38:287–308, 2000.

[39] A. W. Moore and C. G. Atkeson. Prioritized sweeping: Reinforcement learning with less data and less time. *Journal of Machine Learning Research (JMLR)*, 13:103–130, 1993.

[40] Kemal N. Ure, Girish Chowdhary, Jonathan P. How, , and John Vian. Health aware planning under uncertainty for collaborating heterogeneous teams of mobile agents. *IEEE Transactions of Robotics*, 2012 (to be submitted).

[41] Girish Chowdhary and Eric N. Johnson. A singular value maximizing data recording algorithm for concurrent learning. In *American Control Conference*, San Francisco, CA, June 2011.

[42] M. Muhlegg, G. Chowdhary, and Johnson E. Concurrent learning adaptive control of linear systems with noisy measurement. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, MN, August 2012.

[43] C. Tomlin, J. Lygeros, and S. Sastry. Aerodynamic envelope protection using hybrid control. volume 3, pages 1793–1796, 1998.