Cooperative Task Scheduling for Networked Uninhabited Air Vehicles

ALVARO E. GIL
KEVIN M. PASSINO, Fellow, IEEE
SRIRAM GANAPATHY
The Ohio State University
ANDREW SPARKS
Air Force Research Laboratory

In the work presented here, we study a cooperative control problem for a network of uninhabited air vehicles (UAVs) where it is assumed that after deployment a set of tasks is given to a group of UAVs and the UAVs must cooperate to decide which UAV should process each task. The cooperation must occur during real-time operation due to a need to repeatedly process each task, in spite of imperfect communications (e.g., messages with random but bounded delays), and the possibility that tasks "pop-up." We show how to view this as a cooperative scheduling problem, and how to derive bounds on mission-level performance metrics. Simulations are used to compare the approach with a noncooperative strategy and to provide design guidelines for the cooperative scheduler.

Manuscript received February 9, 2006; revised September 1, 2006; released for publication March 5, 2007.

IEEE Log No. T-AES/44/2/926540.

Refereeing of this contribution was handled by T. F. Roome.

This work was supported by the AFRL/VA and AFOSR through the OSU Collaborative Center of Control Science under Grant F33615-01-2-3154. Partial support was also provided by NIST.

Authors' current addresses: A. E. Gil, Xerox Corporation, Webster, NY; K. M. Passino, Dept. of Electrical Engineering, The Ohio State University, 411 Dreese Lab, 2015 Neil Ave., Columbus, OH 43210-1272, E-mail: (k.passino@osu.edu); S. Ganapathy, General Motors Global Research and Development, Honeoye Falls, NY; A. Sparks, AFRL/VACA, 2210 8th St., Wright Patterson AFB, OH 45433-7531.

I. INTRODUCTION

Groups of possibly many uninhabited air vehicles (UAVs) of different types, connected via a communication network to implement a "vehicle network," are technologically feasible and hold the potential to greatly expand operational capabilities at a lower cost (e.g., due to the economies of scale gained by manufacturing many simpler vehicles). Cooperative control for navigation of such vehicle groups involves coordinating the activities of several agents so they may work together to complete tasks in order to achieve a common goal. The coordination can occur via a communication network and the goal could be to optimize the task completion rate. Cooperative control can be useful in a variety of applications including multi-processor computing systems, networked flexible manufacturing systems, and multiple electronic receivers being coordinated to locate multiple radar emitters in an uncertain environment. Here, we study the use of "cooperative scheduling strategies" [1] for coordinating UAVs to perform tasks in a way that tries to minimize the time it takes to complete them, while at the same time trying to perform especially quickly the most important tasks. A mission scenario is provided in which this approach has direct applicability; however, we also explain how the problem we solve arises as a key component of more general scenarios. We provide a theoretical analysis of the properties of the cooperative control system and also provide insights into the effects of communication imperfections on the ability to cooperate effectively. Our simulations compliment these insights and give design guidelines for tuning the controller. An early abbreviated treatment of this work appeared in [2].

There is a significant amount of current research activity focused on cooperative control of UAVs. Solutions to general cooperative control problems can be obtained via solutions to vehicle route planning (VRP) problems [3]. While VRP methods can be used to allocate UAVs to tasks in order to minimize the mission completion time, generally the methods are only applicable when uncertainties are not present in the environment. Moreover, VRP formulations generally consider a finite number of tasks to be completed in a finite time (unlike here). Receding horizon control approaches are studied in [4]–[7]. Work focusing on cooperative search and coordinated sequencing of tasks include [6]–[11] and the "map-based approaches" in [12]–[19]. Using such approaches, significant mission performance benefits can be realized via cooperation in some situations, most notably when there is not a high level of uncertainty. Here, we show how to cope with a type of uncertainty that enters in the form of imperfect communications modeled as random but bounded delays on messages communicated to coordinate activities. Due to the presence of so much uncertainty it is generally not possible to accurately predict far into the future, and hence generally not useful to employ optimization approaches to develop long sequences of planned coordinated operations either off- or on-line (as in many of the above-mentioned studies). It is well known that the complexity of coordinated sequencing/planning is significant when there are, for instance, many UAVs and tasks (e.g., consider the complexity analysis of VRP problems in [3]); however, with significant uncertainty arising from communications, coordinated sequencing/planning of long sequences of tasks is not useful anyway so in this case the challenge should often not even be confronted (predictions can be so inaccurate and result in such poor decisions that it is simply better not to predict ahead). Instead, the challenge is to overcome the effects of uncertainty so that benefits of cooperation can still be realized. Hence, when uncertainty dominates we are not able to achieve the high level of coordination achieved in many of the above-mentioned studies; we simply seek to achieve some benefit from cooperation. Moreover, all the above techniques assume perfect communications. This differs from the approach taken here since our goal is to implement cooperative scheduling strategies that can cope with imperfect communications. Furthermore, this paper pays attention to the amount of information shared through the communication network, scalability issues when the number of UAVs and tasks are high, and real-time implementation of the scheduling strategies. Recent work, however is beginning to focus on the case where there are communication imperfections. In this respect three notable studies are in [20] where the authors consider the problem of dynamic reassignment of tasks among a cooperative group that communicates relevant information asynchronously with arbitrarily finite delays, [21] where the authors study the synchronization of information for cooperative control, and [22] where the authors study a task load balancing approach to cooperative control when there are network delays.

There is an important connection between collective robotics and the work here that could be useful for manufacturing systems. The design of collective autonomous agents has received significant attention and an overview of the research directions in this field can be found in [23]. Studies of the behavior of robot teams communicating over a network are found in [24], [25]. In [26], the authors study how optimality, stability, and communication network issues in autonomous agents working in an uncertain environment (i.e., uncertainties in the system states due to communication delays, in the strategies, and in the pay-off resulting from the agent choices) affect the group performance metric.

Of all the current work in cooperative control, the most closely related to this study is the "persistent area denial (PAD)" problem studied in [27]. In [27], pop-up targets are modeled as Markov chains and cooperative control strategies are implemented to reach all targets in the shortest time. The allocation of UAVs to targets depends on both the distance to the targets and the remaining time of the target's appearance. Our problem is a type of PAD problem. Here, our N tasks need to be revisited by M UAVs ($M < N$) as quickly as possible in a coordinated manner in spite of communication imperfections (in [27] perfect communication is assumed). Although our work does not consider probabilistic pop-up targets exactly as in [27], the theoretical analysis of the cooperative scheduling strategy defined here still holds for the processing of pop-up tasks under certain special conditions. For instance, suppose that for each task, if it is not present at the current time, it will pop up within a fixed number of time units. Delays in task appearance could be due to environmental characteristics, the UAV's sensing limitations, or delays in getting task information from other sources (e.g., a sensor on a high-flying platform). If we know lower and upper bounds on the amount of time between each task's appearance, then we can use this information in our theoretical analysis to derive stability results for the case of pop-up tasks.

The scheduling strategies proposed here provide what is generally a suboptimal solution that seeks to pay attention to high priority tasks and the inter-task travel times at the same time. However, a "global optimal solution" is not likely to be feasible, even if we had perfect communication capabilities, since it requires the solution to a high-dimensional nonlinear optimization problem to pick vehicle paths over an infinite time horizon. Moreover, since we consider the case where there are imperfect communications and the possibility of pop-up tasks, a global optimization approach to design is infeasible. The strategies defined here are inexpensive from a computational point of view since the distributed decisions are only based on the computation of either a maximum or an average value of both the ignored time of processing tasks and inter-task travel times. The cooperative scheduling strategies can easily be utilized independent of the number of UAVs. In fact, there is always just one computation involved in the decision-making process (i.e., a maximum or average value) and the number of the elements considered in the computation on each UAV 1) decreases when the number of the UAVs increases (since for a fixed number of tasks, if there are more UAVs, then typically the number of elements considered in the strategy is smaller), and 2) increases when the number of tasks increases. Hence, the computations needed for real-time implementation of these cooperative scheduling strategies do not expand exponentially when the number of UAVs or tasks increases (i.e., our approach is "scalable").

In this paper, our focus is on modeling a cooperative control problem that has communication imperfections (Sections II and III) and then introducing some strategies (Section IV) that seek to schedule the next task a UAV should perform to avoid ignoring high priority tasks for too long and yet minimize travel time to perform tasks. We then prove in Section V that for such a strategy, even in the presence of delays in communications, we can find a bound on the longest time that the UAVs will ignore any task. This bound, which is a function of the cooperative control problem parameters, provides insights into how, for example, inter-task distances, the number of UAVs, and communication delays will affect the dynamics of cooperation and mission performance objectives. Simulations in Section VI are used to evaluate the performance of the approach and to provide some design guidelines. Section VII explains how the problem arises as a key component of other cooperative control problems and concluding remarks are provided in Section VIII.

II. COMMUNICATION IMPERFECTION SOURCES AND UNCERTAINTIES IN THE ENVIRONMENT

Since our work represents a significant departure from cooperative control based on perfect information, in this section we overview a number of sources of communication imperfections and environmental uncertainties for networked UAVs. This will help to justify some of the details of our problem formulation in the next section, and motivate the importance of explicitly taking into account communication imperfections for cooperative control. Moreover, it shows why it is important to perform comparative analysis with a noncooperative controller, something we do in our simulations.

First, the network may have a communication topology that is not fully connected in the sense that each UAV may not be able to directly communicate with every other UAV. Also, each inter-vehicle communication "link" may be imperfect in that there may be delays or bandwidth constraints in sending or receiving information, misordering of messages/information, or noise that corrupts the information. Centralizing the information gathered by the group of UAVs for the purpose of coordinating their actions is a natural approach in networked UAVs; however, centralization of information (e.g., via one special UAV) does not overcome the problem of the presence of imperfect communications since a centralized location would have to be communicated with anyway in order to share information and coordinate actions. Moreover, it may not be possible to have one central place to keep all information as if this central place is one special UAV there is a problem with fault tolerance of the whole system (if there is a problem with the special UAV the whole system may not be able to function). Moreover, it is typically not possible to keep all information at the "home base" as it may be out of communication range and full autonomy is not achieved.

Imperfect communications arise from many sources. First, there is no perfect communication network in spite of recent and envisioned advances in communication network technology. Link bandwidth, delays (e.g., random but bounded ones), and noise are significant problems. The communication topology may vary with time in unpredictable ways. The need for network security typically makes these problems even worse (e.g., there are more significant delays due to the need for encryption in some cases). Even typical envisioned networks for aircraft can have communication delays between 1–4 s, with problems in message misordering. It is not clear at this time what bandwidth will be possible (for some envisioned cooperative control methods the bandwidth is more than sufficient); however, a reasonable approach to avoid confronting that constraint is to avoid the need for passing extraordinary amounts of information (e.g., high resolution images) when there are hard real-time constraints.

It is also important to recognize that communication imperfections can arise from sources other than the physical network itself. For instance, if "line-of-sight" is needed for the communication technology (as is the case for some aircraft-aircraft communication systems), then as a group of UAVs moves there could be occlusions due to, for example, the terrain. Moreover, in some envisioned systems there will be "range constraints" on communications (e.g., due to power limitations on communications) that could easily lead to dynamic breaking and reestablishment of communication links (i.e., communication topology changes). In some contexts, such effects may be modeled as communication delays that can be of quite a significant length of time (e.g., minutes or more). Next, note that in some missions there may be a need to periodically communicate with the home base to have a type of partially autonomous "semi-tele-operated" UAV. One good example of this is when UAVs are sent on a mission but are not allowed to automatically "classify" targets/threats, and then attack based on their conclusions. Instead, UAVs are required to send sensor data back to a home base where humans will do the classification. The UAVs then have to wait for a response from the humans in order to take any actions such as attacking a target. Delays of this sort can often be modeled as a type of communication delay. To see why, first view the home base as another "agent" in the group of UAVs. Next, the delay induced by the humans can be modeled as a lag in getting a response over the communication link with the home base. Clearly, in this case the delays could be on the order of 10 min or more. More money can be invested to alleviate some of the limitations of the physical communication network, but these other problems can be more significant and difficult to solve.

Uncertainty in the environment arises from having imperfect information about the tasks that need to be processed by the UAVs or malfunctioning of the UAVs' devices. For instance, once a UAV is pursuing a task, it may need to travel to where the task is; however, if this UAV only knows that the task to process is within a specific area (due to imperfect sensor information), then this UAV may spend more time trying to search for the task location than what the UAV originally estimated. Another clear example related to the environment is the pop-up type task. In this case the UAV can be ready to process the task but it needs to wait for the appearance of the task in order to start the processing of it. Thus, there is a time delay related to these uncertainties. There may be cases where the UAVs need to have their local clocks synchronized by a global clock. Desynchronization of the local clocks, which could be viewed as a malfunctioning device, could lead the UAVs to incorrect decisions that would affect the performance of the whole group of UAVs.

In summary, we see that there can be significant uncertainties in communications and the environment. Cooperation requires shared information, either via a priori information or communication of information gathered during a mission. If that information is not perfect then we naturally expect to achieve lower levels of cooperation and hence performance. Uncertainty creates a type of passing of "bad information" that typically leads to poor group decision making, and if the information is quite poor it could be worse to try to cooperate than to simply take a noncooperative approach where no communications are required. This last statement is justified in the context of the problem we study via both our theoretical and simulation-based analysis.

III. PROBLEM STATEMENT

Suppose that there are several UAVs that need to process a set of tasks where processing a task requires a UAV to go to a certain location in a region of finite size. Here we assume that a set of UAVs is given a number of tasks, their respective locations, and characteristics (e.g., the priority of each task). It is assumed that the tasks must be repeatedly visited and processed (e.g., for repeated surveillance of points spread across a large region where the points of interest are more numerous than the number of UAVs) [27]. UAVs must work together autonomously in order to maximize the rate of task processing for the highest priority tasks.

A. Tasks, Prioritized Time, and Processing Time

Suppose that the number of tasks is fixed and that we number and denote them as $P = \{1,2,\ldots,N\}$. Assume that the number of UAVs is constant and we number them and denote the set of UAVs as $Q = \{1,2,\ldots,M\}$ where $N > M$. If $N \leq M$, then depending on the spatial task distribution, at least one UAV can be dedicated to each task or each UAV's processing capability can be coordinated in such a way that each one visits several tasks. Let p_i, $i \in P$ denote the priority (importance) of processing task i. Let t denote time. Let $T_i(t)$, $i \in P$, $t \geq 0$ denote the "prioritized time" since last processing of task i, and t_i denote the time since the last processing of task i (e.g., if task i was last processed at time zero then the quantity $T_i(t) = p_i t_i$ is its "prioritized time"). We assume that there is a cooperative scheduling strategy that decides which task a UAV should process next. Assume that there is a global clock (e.g., via GPS) that keeps all the UAVs' clocks synchronized (a reasonable assumption in the context of this class of applications). Let $x^i = [x_1^i, x_2^i, \theta^i]^\top$ denote the coordinates in the (x_1^i, x_2^i) plane and orientation θ^i of the ith task. Let $d(x_v^j(t), x^i)$ be the distance that UAV j must travel from its current location and orientation $x_v^j(t) = [x_{v_1}^j(t), x_{v_2}^j(t), \theta_v^j(t)]^\top$ to process task i at x^i. Here, as in a number of UAV studies [8, 9, 22] we use a Dubin's car [28] as a model for UAVs flying at constant altitude and at a constant velocity v, where

$$\begin{aligned} \dot{x}_{v_1}^j(t) &= v \cos\theta_v^j(t) \\ \dot{x}_{v_2}^j(t) &= v \sin\theta_v^j(t) \\ \dot{\theta}_v^j(t) &= \omega_{\max} u_v^j(t) \end{aligned} \tag{1}$$

where $\omega_{\max} > 0$ is the maximum angular velocity, and the steering input $u_v^j(t)$ is subject to the following constraint for all $t \geq 0$,

$$|u_v^j(t)| \leq 1 \Leftrightarrow |\dot{\theta}_v^j(t)| \leq \omega_{\max}. \tag{2}$$

Collisions between UAVs are ignored in this work since it is not the primary goal of this study. We use this model only to be concrete and to enable us to perform numerical simulation studies. In fact, all results in this paper hold for any vehicle such that if its current state is known, then an upper bound on how long it takes to reach another given state can be computed (and due to the current high level of understanding of aircraft dynamics this includes virtually any current aircraft).

If communication delays are finite and tasks are in a finite region, then we can find a $\bar{d} \geq d(x_v^j(t), x^i)$ for all i, j and $t \geq 0$. Given a certain scenario that contains a fixed number of tasks located in a finite region, we could determine $\bar{d}$ by choosing the longest

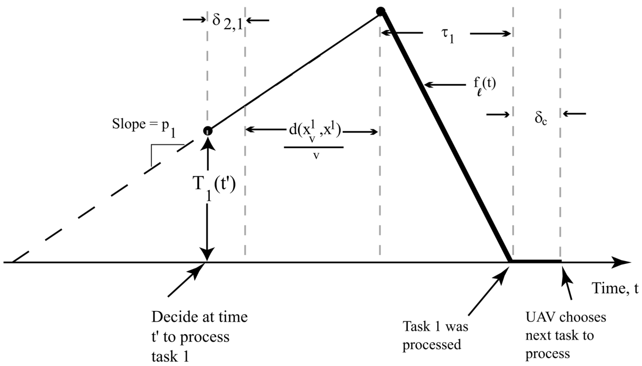

Fig. 1. Illustration of timing of UAV decision making and size of prioritized time since last processing.

travel distance that any UAV must travel to any task considering their respective orientation angles. Suppose that all UAVs move either on a constant minimum turn radius or on a straight line for the Dubin's car (i.e., the optimal trajectories as explained in [29]). Let τ_{i_j}, $0 < \boldsymbol{\tau} \le \tau_{i_j} \le \bar{\tau}$, $i \in P$, $j \in Q$, denote the processing time UAV j takes to process task i with $\boldsymbol{\tau}$ ($\bar{\tau}$) being the minimum (maximum) time any UAV takes to process task i after it first arrives at it (here "processing" could include multiple passes over an object at different angles in order to detect or identify/classify it). Let $\delta_{i,j}$, $0 \le \delta_{i,j} \le \bar{\delta}$ be the random but bounded time delay that represents the amount of time any UAV takes to switch from processing task i to task j, $j \neq i$ (e.g., to set up UAV's sensors or munitions). Let $\delta_c > 0$ be the random but bounded time delay incurred when UAV j has finished processing any task but it needs to wait for information held by another UAV in order to make a decision (see below for more details). Assume that δ_c is bounded by $\bar{\delta}_c > 0$.

To clarify, consider the case where there are only three tasks ($N = 3$), named "task 1," "task 2," and "task 3" and two UAVs ($M = 2$). Suppose that at some time t', the value of prioritized time since last processing task 1 is $T_1(t') > 0$ as shown in Fig. 1 and the last task processed by UAV 1 is task 2. At time $t' + \delta_{2,1}$ UAV 1 is heading to task 1. Then, at time $t' + \delta_{2,1} + d(x_v^1, x^1)/v$ UAV 1 is at the location where task 1 is and it initiates the processing of task 1, and the amount of time that it takes to do so is dictated by the τ_1 parameter. When processing for task 1 is completed, the UAV sends a request to UAV 2 to coordinate decision making and thereby incurs the delay δ_c. Finally, at time $t' + \delta_{2,1} + [d(x_v^1, x^1)/v] + \tau_1 + \delta_c$ UAV 1 chooses the next task to perform. Moreover, we assume that $T_1(t)$ in Fig. 1 could have any shape for the time when UAV 1 is processing task 1 to represent how UAV 1 processes this task. Fig. 1 shows that $T_i(t) = f(t)$ for

$$t' + \delta_{i,1} + \frac{d(x_v^1, x^1)}{v} \le t \le t' + \delta_{i,1} + \frac{d(x_v^1, x^1)}{v} + \tau_1$$

and we assume that function $f(\cdot)$ must additionally satisfy $f(t) \le f_\ell(t)$, where $f_\ell(t)$ is the bold line shown in Fig. 1. Notice that the processing time τ_1 does not depend on the value of $T_1(t' + \delta_{2,1} + d(x_v^1, x^1)/v)$ since no matter what the current value of the prioritized time since last processing is, the UAV will always take τ_1 time units to process task 1. One consequence of this is that the slope of $f_\ell(t)$ will not be the same each time a UAV is processing a task.

B. Asynchronous Group Decision Making

We define the set $U(t) \subset P$ as the "unattended" tasks not processed or being pursued by any UAV at the current time t, while the set $U_j^a(t) = \{i_j^*(t)\} \cup U(t)$ is the set of tasks that can be considered for processing by UAV j, $j \in Q$. Here, $i_j^*(t)$ is the task being processed or pursued by UAV j at time t. Define $A(t)$ as the set of tasks processed or pursued by the group of M UAVs at the current time t; hence $P = U(t) \cup A(t)$, $t \ge 0$. Let

$$S(t) = \{\{T_i(t) : i \in U(t)\}, t_s\}$$

denote the information set available for UAV j at time t when it makes a scheduling decision, where t_s is a "time stamp" that indicates the last time that the $T_i(t)$, $i \in U(t)$ were updated.

Next, we explain how asynchronous decision making across the group of UAVs is accomplished. Define a UAV $j^u \in Q$ that holds the set of information for scheduling decisions $S(t)$. We assume that whenever a UAV $\ell \in Q$ where $\ell \neq j^u$ (if $\ell = j^u$ there is no need for a request) finishes processing a task at time t^f such that $T_{i_\ell^*}(t^f) = 0$ (i.e., the instant the task has been processed), it broadcasts a request for the set $S(t)$ to all the UAVs. Let the amount of time it takes to broadcast the request and receive $S(t)$ be $\delta_c > 0$ which again is random, but bounded by a constant $\bar{\delta}_c > 0$. In the time interval $[t^f, t^f + \delta_c]$ that UAV ℓ waits for $S(t)$, $T_{i_\ell^*}(t') = 0, t' \in [t^f, t^f + \delta_c]$, which means that UAV ℓ keeps processing task i_ℓ^* until it receives $S(t)$. This is consistent with our definition of the prioritized time since last processing since task i_ℓ^* has already been completely processed at time t^f and it is not being ignored during the interval $[t^f, t^f + \delta_c]$ anymore. The instant that UAV ℓ gets $S(t)$ (and the "request queue" defined below), it becomes UAV j^u, it compares the t_s value to its local clock, and it proceeds to update all $T_i(t), i \in U(t)$, values if there exists any mismatch (the time stamp t_s indicates to UAV ℓ when was the last time that UAV j^u updated all $T_i(t), i \in U(t)$). By doing so, UAV ℓ makes a decision on what task to process next with up-to-date information (note that even if the UAV's local clocks are not synchronized, the decision maker will be able to select a new task to process).

Since two or more UAVs could request the set $S(t)$ at the same time, we need to use a mutual exclusion algorithm which coordinates the access of all UAVs to the set $S(t)$ in such a way that this set can be accessed and updated by only one UAV at a time. Assume UAV j^u has a request queue. There are certain ways of creating that queue, one possibility being the first-in first-out (FIFO) policy, and another is to simply use a predefined order (e.g., requests made from UAV 1 up to UAV M). Thus, if UAV j^u has already built a queue and updated the set $S(t)$, it proceeds to send $S(t)$ and the request queue to the UAV located in the head of the queue when it transmits it. The UAV that receives $S(t)$, and the request queue updates the set $S(t)$, and passes this set along with the queue to the new UAV at the head of the queue, and this process is repeated.

Note that we have described the case where the set $S(t)$, and the request queue, are passed along the network and they are held by the UAV that requested this information; it is clear for this case that a "tracking" mechanism may be needed to know the current UAV that holds this information (e.g., a sparsely connected communication topology), unless broadcast type requests are made as we assume here. However, another scenario can be studied as well, where UAV j^u always holds the set $S(t)$, and the request queue, and whenever a UAV $\ell \in Q$, $\ell \neq j^u$, requests the set $S(t)$ held by UAV j^u, it modifies it with the new unattended tasks and time stamp, and sends it back to UAV j^u. Regardless of the strategy used to share $S(t)$, here the key point will be that it is shared over an asynchronous network where random but bounded delays can be incurred as we discuss next.

Let k^j, $k^j \in \{0,1,2,\ldots\}$, denote the index of the sequence of times that UAV $j \in Q$ makes task scheduling decisions. Let D_{k^j} be the time when UAV $j \in Q$, decides to process task $i_j^*(k^j)$, and assume that at the initial time $D_{k^j} = 0$ for $k^j = 0$. Let D_{k^j+1} be the next decision time for UAV j, which is when it completes the processing of task $i_j^*(k^j)$ and gets $S(t)$ from UAV j^u. For each j define $D_{k^{j^c}}$ to be the closest decision time made by any other UAV j^c, previous to the decision time D_{k^j+1} (so given j we can define j^c at each D_{k^j+1}). Moreover, having introduced this notation, it can be noted that UAV j^c is always going to be equal to UAV j^u (the queue holder), which means that UAV j will receive the request queue from UAV j^c at time D_{k^j+1}. Note that if no other UAV except j makes a decision between times D_{k^j} and D_{k^j+1}, then $D_{k^{j^c}}$ is just equal to D_{k^j} so $j^c = j$. Since $\delta_{i,j} \geq 0$ and $\delta_c > 0$ we know that $D_{k^j+1} > D_{k^{j^c}}$. By the definition of j^c, $D_{k^j} \leq D_{k^{j^c}} < D_{k^j+1}$, and we know that

$$D_{k^j+1} - D_{k^{j^c}} < D_{k^{j^c}+1} - D_{k^{j^c}} \leq \bar{\delta} + \frac{d(x_v^{j^c}(D_{k^{j^c}}), x^{i^*_{j^c}(k^{j^c})})}{v} + \bar{\tau} + \bar{\delta}_c. \quad (3)$$

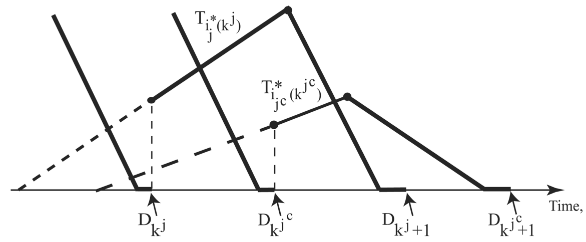

Fig. 2. Example decision times for $M = 2$ UAVs.

This can be seen in Fig. 2. Note that in the time interval $t \in [D_{k^{j^c}}, D_{k^j+1}]$ the set $U(t)$ is constant. This will be useful in our proof below.

IV. STABLE COOPERATIVE SCHEDULING

Tasks are spread in a limited region and each UAV can process just one task at a time. There are M resources that must be shared (UAVs capabilities), and the scheduling strategy must decide how they are shared (what task to process). Note that if no UAV was actively engaged in processing, then clearly $T_i(t) \to \infty$, $i \in P$, $t \to \infty$ since no tasks are processed. We consider the system to be "unstable" if $T_i(t) \to \infty$, $t \to \infty$ for any $i \in P$. Hence, the goal of the scheduling allocation strategy is to try to avoid $T_i(t) \to \infty$ for any $i \in P$ and indeed it will try to keep the $T_i(t)$ values as small as possible since this represents that the set of UAVs has recently processed each task. We consider a scenario to be "stable" when there exists a $B > 0$ such that $T_i(t) < B$, $i \in P$ and $t \geq 0$ (we are then thinking of stability as in [30] as Lagrange stability or uniform ultimate boundedness).

First, we show that scheduling strategies that choose every task $i \in P$ within finite time interval result in stable systems. Studying this case will help motivate our choice of cooperative scheduling methods. For simplicity, we study the case where there is just one UAV ($M = 1$) processing task. Let S_i denote the time elapsed between two consecutive decisions for the same task $i \in P$; this is simply how long it will take for the one UAV to choose the same task $i \in P$ again (there could be other decisions for tasks $j \neq i$ in between). Let W_i denote the time elapsed between the two contiguous completions of task i (i.e., the time difference between two consecutive $T_i(t^1) = 0$ and $T_i(t^2) = 0$, $t^2 > t^1$). Now, if the tasks in the environment are always processed in a preestablished order, then we can easily derive an expression for S_i as

$$S_i = \sum_{i=1}^{N} \left(\delta_{j,i} + \frac{d(x_v^1(t), x^i)}{v} + \tau_i \right) \leq N \left(\bar{\delta} + \frac{\bar{d}}{v} + \bar{\tau} \right), \quad i \in P, \quad t \geq 0.$$

Moreover, any scheduling strategy that commands one UAV to persistently process all tasks will result in a stable system. Hence, there exists a $B > 0$ such that $T_i(t) \leq B$ for all $i \in P$ and $t \geq 0$. In Fig. 1 we can see that the maximum value of T_i occurs at the peak of the plot, which is the time when the UAV reaches for the first time the location where the task is. According to the definition of $T_i(t)$, we can obtain then the ignored time for this peak, $t_i^{\text{peak}}(t)$, as

$$t_i^{\text{peak}}(t) = \frac{T_i(t)}{p_i} \leq \frac{B}{\underline{p}} \tag{4}$$

where $\underline{p} = \min_i\{p_i\}$.

Moreover, the ignored time for any task i is

$$t_i(t) \leq \frac{B}{\underline{p}}. \tag{5}$$

Now, we can use (5) to obtain that $W_i(t) \leq B/\underline{p} + \tau_i \leq B/\underline{p} + \bar{\tau}$, and since in this case there are always two consecutive decisions for any task i among three contiguous completions of processing of the same task, i.e., $S_i(t) < 2W_i(t)$, so

$$S_i(t) < 2\left(\frac{B}{\underline{p}} + \bar{\tau}\right), \qquad i \in P.$$

Since we have shown that any scheduling strategy that satisfies the conditions defined above will stabilize the scenario considered here, we could use the ones introduced in [30, 1] for the $M = 1$ case in this framework; however, while these policies will consider the sizes of the T_i values (they will avoid ignoring high priority tasks), they will likely lead to wasting UAV fuel since they completely ignore travel times due to varying spatial distances.

This is why we are motivated to introduce a cooperative scheduling strategy that seeks to pay attention to two variables: high priority tasks and travel time (fuel expenditure). Moreover, the above methods in [30] only provide stability for the $M = 1$ case where here we need to consider the $M > 1$ case with asynchronous decision making and bounded communication delays where simple bounding arguments on times between completions of tasks are not possible. Furthermore, we are interested in deriving bounds for the ignored time since last processing in terms of known parameters, not just parameters values that are just known to exist (e.g., B above).

A. Cooperative Scheduling Strategies

Next, we introduce a particular cooperative scheduling strategy that we study in the remainder of the paper. Other strategies that can be applied to this particular problem can be found in [1] which are extensions of those in [30]. For this particular case, at time k^j the cooperative scheduling strategy on each UAV j chooses to process task $i_j^*(k^j)$, such that $i_j \in U_j^a(D_{k^j})$

$$T_{i_j^*(k^j)}(D_{k^j}) - \frac{d(x_v^j(D_{k^j}), x^{i_j^*(k^j)})}{v} \geq \frac{1}{N - M + 1} \sum_{i_j \in U_j^a(D_{k^j})} \left[T_{i_j}(D_{k^j}) - \frac{d(x_v^j(D_{k^j}), x^{i_j(k^j)})}{v}\right] \tag{6}$$

and makes no other decision until it has finished processing task $i_j^*(k^j)$ and received $S(t)$. Note that the quantities of both sides of (6) can be positive or negative and if they are both zero then any task can be chosen for processing. Ties are broken with an arbitrary choice. Note that when UAV j finishes processing task $i_j^*(k^j - 1)$, it chooses a new task $i_j^*(k^j)$ from the set $U_j^a(D_{k^j})$, and then replaces it with $i_j^*(k^j - 1)$ to form $U(D_{k^j})$. Generally for $j \neq j'$, $D_{k^j} \neq D_{k^{j'}}$, and $U_j^a(D_{k^j}) \neq U_{j'}^a(D_{k^{j'}})$ so (6) represents how decisions are made over a range of M times. Since there can be many more decisions made by one UAV than another it could be that $D_{k^j} - D_{k^{j'}} \to \infty$ as $k^j \to \infty$ and $k^{j'} \to \infty$, $j \neq j'$. Note that although the UAVs could complete processing of their respective tasks at the same time, their decisions will occur at different times since the UAVs will make choices depending on the queue held by UAV j^u so that they will pick different tasks to process due to the use of the mutual exclusion algorithm.

Equation (6) produces a set of admissible choices for what UAV j can process. In particular, the strategy "process M closest highest priority tasks" chooses to process task $i_j^*(k^j)$ where

$$i_j^*(k^j) = \arg\max_{i_j} \left\{T_{i_j}(D_{k^j}) - \frac{d(x_v^j(D_{k^j}), x^{i_j(k^j)})}{v}\right\}, \quad i_j \in U_j^a(D_{k^j}) \tag{7}$$

is a special case of (6) in the sense that it represents one possible choice for (6); hence when in the Appendix we do stability analysis for (6) it also applies if we use (7) for our strategy. It has been shown in an experimental testbed for networked cooperative scheduling strategies, an "electromechanical arcade," that a better performance is obtained by the strategy shown in (7) compared with scheduling strategies that seek to optimize just one objective (e.g., sizes of the T_i values) [31]. In the electromechanical arcade the authors consider both an "environment" that is highly uncertain (e.g., due to uncertain pop-up target appearance times) and imperfect communications that make it difficult for the decision makers to coordinate their actions. This experiment can be viewed as a 1-D version of the model introduced in Section IIIA since the "guns" (UAVs) need to process the targets (tasks) in a cooperative manner.

Notice that the left-hand side of (6) could be viewed as a cost function and the goal of the strategy is to locally optimize the difference between the prioritized time since last processing and the travel time from current UAV location to the chosen task location. Another way to view this cost function is by thinking of the chosen task $i_j^*(k^j)$ by UAV j as the one whose prioritized "ignore time" combined with the UAVs travel time is greater or equal to the average value of those variables for each task contained in the set $U_j^a(D_{k^j})$. Thus, if all the tasks contained in the set $U_j^a(t)$ have the same prioritized time since last processing but they are located at different distances from a UAV, then this UAV chooses to process the task that is closest to it. In this sense, the strategy is "myopic" (but consider the discussion in Section I on why "look-ahead" can be detrimental for the type of cooperative control problem we consider).

In some cases the p_i values are set a priori by constraints of the problem. In other cases it is possible to view them as controller design parameters that can be tuned to improve performance. Recall that the p_i parameters are embedded in the T_i variables, so by changing these parameters any UAV can put more or less emphasis in the time since last processing. That is, if the p_i values are all too small, then the UAVs will tend to choose the closest tasks at any decision time, whereas if all p_i values are too big, then the UAVs will tend to choose tasks to process based on the ignored time, neglecting the travel distance to the tasks.

V. MAIN THEORETICAL RESULT

In this section we present our main theoretical result, which focuses on the stability (boundedness) of the t_i variables when the strategy defined in (6) is used. The proof of this theorem can be found in the Appendix.

THEOREM *Assume that $N > M$. For the cooperative scheduling strategy in* (6) *a specific bound on the ultimate longest time that any UAV will ignore task $i \in P$ is given by*

$$\lim_{t\to\infty} t_i(t) \le \max\{B^1, B^2\}$$

where

$$B^1 = \frac{(\bar{\delta} + \bar{\tau} + \bar{\delta}_c)}{\underline{p}}\left(\sum_{i=1}^{N}\frac{p_i}{M} - \underline{p}\right)(NM - M^2 + M + 1) + \frac{\bar{d}}{\underline{p}v}\left(M(N - M + 1) + \bar{p} + \sum_{i=1}^{N}\frac{p_i}{M} - \underline{p}\right) + \frac{\bar{\delta}}{\underline{p}}\bar{p}$$

$$B^2 = \frac{\left(\bar{\delta} + \bar{\tau} + \bar{\delta}_c + \bar{d}/v\right)}{\underline{p}}\left(\sum_{i=1}^{N}\frac{p_i}{M} - \underline{p}\right)(NM - M^2 + M + 1) + \frac{\bar{p}}{\underline{p}}\left(\bar{\delta} + \frac{\bar{d}}{v}\right)$$

where $\underline{p} = \min_i\{p_i\}$, $\bar{p} = \max_i\{p_i\}$, and $\bar{d} = \max\{d(x_v^j, x^i)\}$.

Next, we make a few remarks about the ultimate bound obtained and how to study other scenarios not considered in this paper.

REMARK 1 Note that when the bounds on the time delays $\bar{\delta}$ and $\bar{\delta}_c$ increase, the bound also increases (e.g., network delays can result in ignoring tasks longer). The ultimate bound decreases if the velocity of the UAVs increases since the UAVs can move faster to process tasks. If the priority values assigned to tasks increase, then the ultimate bound does also since then we may spend more time processing the higher priority tasks and hence an increased amount of time ignoring the lower priority tasks. If all tasks are spread out more (i.e., $\bar{d}$ increases), then the ultimate bound increases since it takes longer to travel to process tasks, and this provides a clear idea how "task spatial density" could affect the ultimate bound. The ultimate bound increases if the processing time of every task is increased since the UAVs are busier processing and hence ignore other tasks longer. If the number of tasks and UAVs are about the same, then the ultimate bound decreases.

REMARK 2 Uncertainties in the location of tasks and pop-up tasks can be added to the model introduced in Section III. For instance, let $\delta(t) > 0$ denote the delays from the switch to processing a different task, the inter-task travel time, the additional time spent on finding the real task location, and the waiting time for pop-up task appearances. The delay that represents the time any UAV takes to switch from processing task i to task j, $j \ne i$ is defined here as $\delta_{i,j} \le \bar{\delta}$, and the inter-task travel time is defined as $d(x_v^j(t), x^i)/v \le \bar{d}/v$. Now, suppose that the additional time spent by any UAV on finding task i is bounded and denoted by $\delta_s^i(t) \le \bar{\delta}_s$. Moreover, suppose that for task type i there is some bound δ_p^i on the amount of time that it would take for any UAV to first realize that the task i may appear, if that was the only task that the UAV is waiting to process (clearly this would depend on the task appearance period). Suppose that $\delta_e^{i,j}(t)$ denotes the delay incurred by UAV j in first getting an indication of the presence of task i, from the time that it arrives to the correct position where the task is. Recall that this value is going to be determined by the time period of appearance of each task i. Note that if we let

$$\bar{\delta}_p = \max_i\{\delta_p^i\}$$

then $\delta_e^{i,j}(t) \le \bar{\delta}_p$. Let

$$\delta(t) \le \bar{\delta} + \frac{\bar{d}}{v} + \bar{\delta}_s + \bar{\delta}_p = \delta.$$

For convenience, we let δ denote a constant that is the least upper bound on $\delta(t)$ (i.e., we simply remove the time index to denote the least upper bound on the

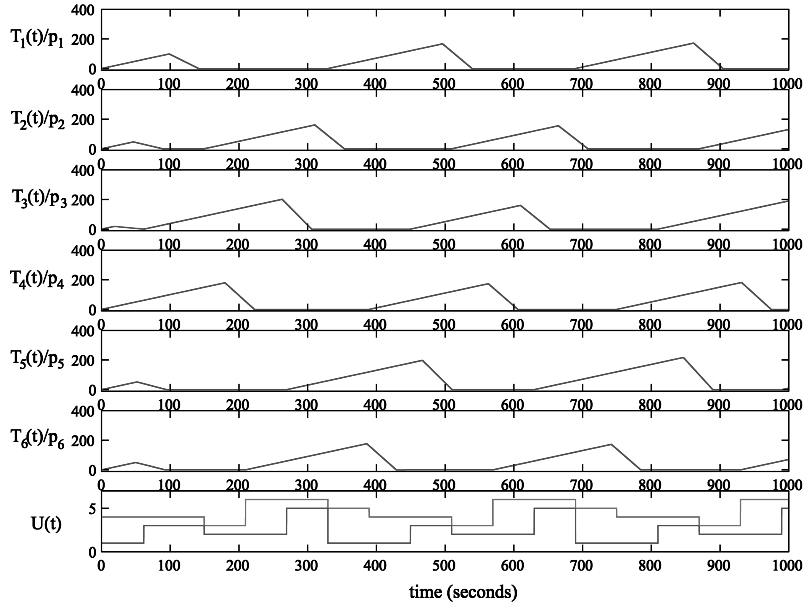

Fig. 3. Time since last processing (seconds) of every task and unattended set when $p_i = 50$, $i \in \{1,2,\ldots,6\}$.

variable). This value δ will replace then the terms $\delta_{i,j}$ and $d(x_v^j(t), x^i)/v$ in appropriate places where they appear, even in the ultimate bound of the main result. Note that these new variables (i.e., $\delta_e^{i,j}(t)$ and $\delta_s^i(t)$) included in the model could now lead us to the introduction of new strategies that account for the time delays caused by the appearance of the pop-up tasks or the additional time spent by any UAV in order to find the task's real location. Hence, the UAVs could then seek to achieve a desire to balance several objectives: high priority processing tasks, minimal inter-task travel times, minimal search time, and minimal waiting appearance time for pop-up tasks.

One important point to highlight here is the fact that the analysis can take into account the behavior of heterogeneous UAVs (i.e., UAVs with different processing capabilities for the same task) via different τ_i. It also inherently takes into account trade-offs between task priorities, network delays, and spatial separation between tasks, something that has not been analytically characterized in past studies of cooperative control.

VI. SIMULATIONS

Here we show in simulation how the cooperative scheduler will ignore lower priority tasks longer and how the ignored time for tasks is affected by communication delays. Also, we introduce a noncooperative controller and derive design guidelines for both noncooperative and cooperative scheduling systems.

A. Influences of Priorities and Communication Delays

We run two simulations with the following values: the sampling time is $T_s = 0.1$ s, the length of the simulation is 1000 s, there are $N = 6$ tasks, $M = 4$ UAVs, the switching times for UAVs 1 and 2 are $\delta_{i,j}^1 = \delta_{i,j}^2 = \delta_{i,j}^3 = \delta_{i,j}^4 = 0$ s, and we consider a fixed communication delay of $\delta_c = 60$ s (i.e., a delay in the middle of the range of possible values that could arise in the type of problem we consider as discussed in Section II).

The first simulation considers $p_i = 50$, $i \in P$. Fig. 3 shows in the top 6 plots the time since last processing of any UAV for all tasks, and the unattended tasks in the bottom plot.

Next, we let $p_1 = 10$ and do not change the rest of the values and get the result in Fig. 4. Notice that since the priority of task 1 has been decreased this task is ignored more than in the first simulation. This shows that UAVs can be forced to ignored tasks by assigning low priority values to tasks. Furthermore, it can be seen that there are intervals where the waiting time for the information set $S(t)$ is greater than δ_c = 60 s due to the amount of previous requests that have been made by other UAVs.

Now, we let all the priorities have the same values ($p_i = 50$, $i \in P$) and we study two cases: no communication delays, $\delta_c = 0$, as shown in Fig. 5 and random but bounded delays, $\delta_c \leq 180$ s, as shown in Fig. 6. It is seen from the figures that the delay decreases the rate at which tasks are processed and also increases the ignored time of each task.

B. Comparative Analysis: Noncooperative Versus Cooperative Strategies

Here, we also consider a group of M UAVs not connected over a communication network. Due to the lack of communication and hence lack of

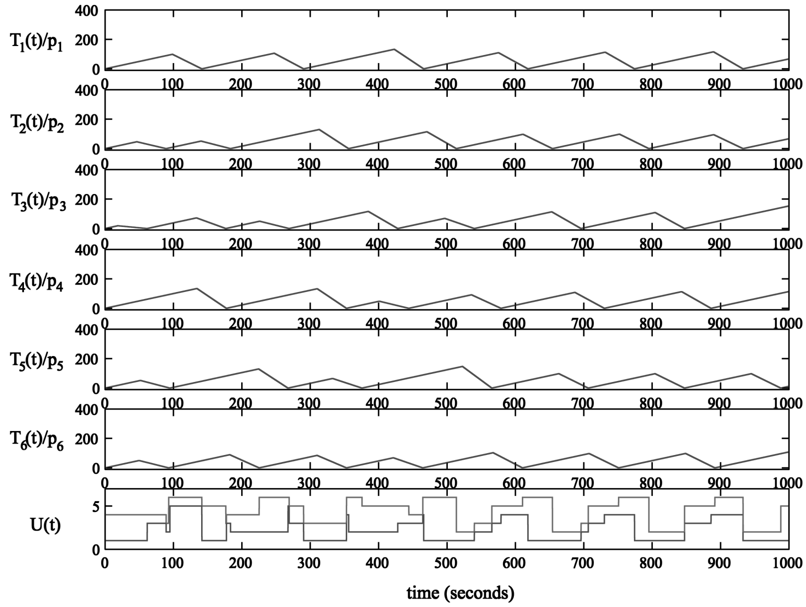

Fig. 4. Time since last processing (seconds) of every task and the unattended set when $p_1 = 10$, $p_i = 50$, $i \in \{2,\ldots,6\}$.

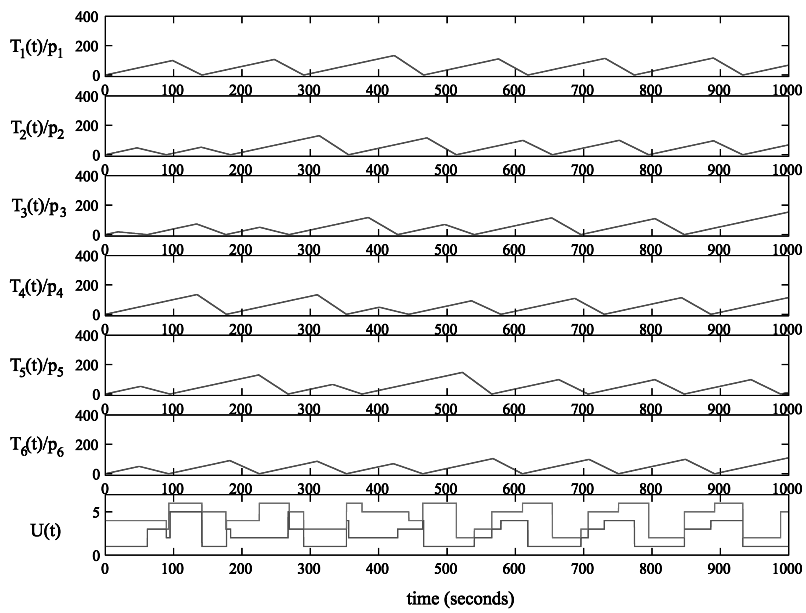

Fig. 5. Time since last processing (seconds) of every task and unattended set without communication delays.

coordinated decision making we call such a strategy "noncooperative." Each UAV has its own $T_{i_j}(t)$, $i \in P$, $j \in Q$ and the prioritized time since last processing for each task is given by $T_i(t) = \min_j\{T_{i_j}(t)\}$, $j \in Q$. Notice that from the task point of view, this represents how long a task has been ignored by any UAV. For this case, UAVs make scheduling decisions based on (6) but considering all tasks at each decision time, hence there could be cases where one or more UAVs are processing the same task during a certain time interval which is generally a waste of processing resources.

We seek to determine if cooperative strategies are always superior to noncooperative ones. We would like to obtain conditions under which it is best not to cooperate. We run a Monte Carlo simulation with the following values: the sampling time $T_s = 0.1$ s, fixed delays

$$\delta_c \in \{1,10,50,100,150,200,300,400\} \text{ s}$$

a set of densities with standard deviation

$$\sigma \in \{500,1000,2500,4000,5000\} \text{ m}$$

(we use a 2D Gaussian distribution for tasks in the middle of the region with mean $(\bar{x},\bar{y}) = [5000,5000]$ m), no switching delays, and the simulation length of 2500 s. Each delay-density case consists of 100 simulations. The number of

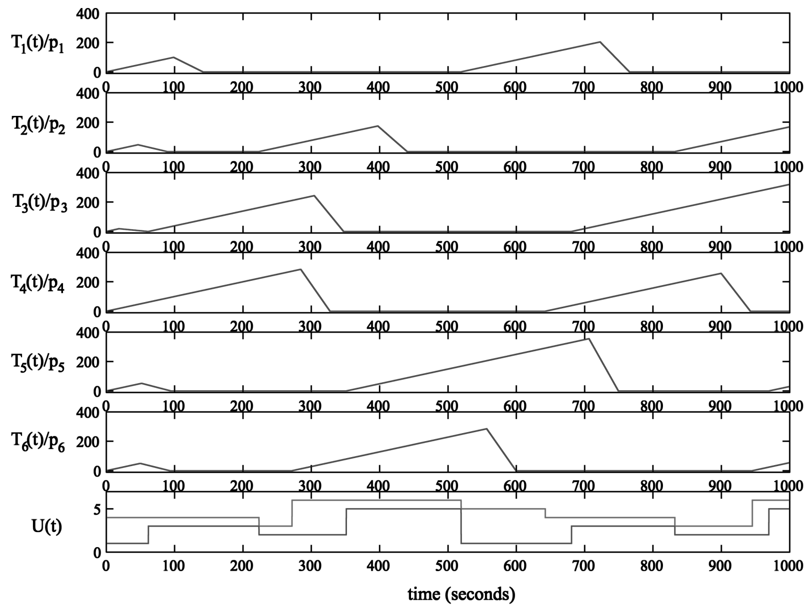

Fig. 6. Time since last processing (seconds) of every task and unattended set for random but bounded communication delays.

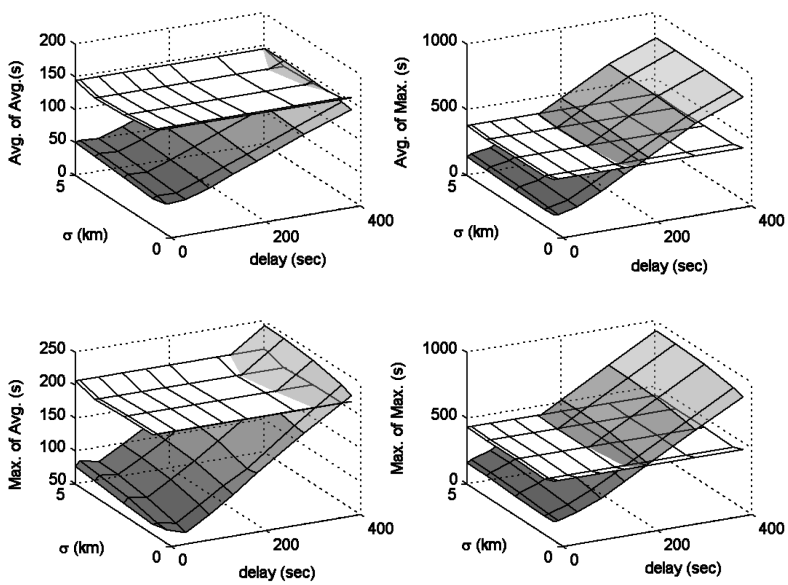

Fig. 7. Performance measures of Monte Carlo simulation (white surface is for noncooperative case and shaded surface is for cooperative case).

simulations for each delay-density combination was chosen to be 100 because the standard deviation of the performance measures (introduced below) did not change significantly and settled to a relatively small constant value beyond 100 simulations (less than 6.67 in all cases).

To establish a fair comparison between the performance of noncooperative and cooperative strategies, we need to introduce a way to evaluate the performance of the UAVs. We compute the average of the ignored time since any task has been processed $(1/N)\sum_{i=1}^{N} t_i(k)$ at each step k. We also compute the time average of this quantity (i.e., the time average of the average values, "average-of-average") and the maximum average value, "max-of-average," achieved over the entire simulation run. We compute the maximum time that any task has been ignored at each time step k, $\max_i\{t_i(k)\}$. We also compute the time average of this quantity (i.e., the time average of the maximum values, "average-of-max") and the maximum of the maximum values, "max-of-max," achieved over the entire simulation run.

Fig. 7 shows that different performance measures give different bounds on the delay for which the cooperative case degrades to the performance of the noncooperative case. For all the cases, the performance measure increases very slowly for lower values of delays (less or equal to 1 min), when compared with the higher values of delays, where there is a steady increase in the performance measure. The effect of decreasing density only results in a relatively small increase in the different performance measures (of course densities vary over a larger range and their impact will be more significant); this increase is due to the targets being spread further apart. As expected, the average-of-average and max-of-average result in a higher value of the delay "cross-point," where the cooperative case degrades to the noncooperative case, than the max-of-max and average-of-max since the former considers performance over average values while the latter considers performance over maximum values (e.g., the max-of-max quantifies worst-case performance). Notice that there are flat regions or valleys for small delay values for some of the performance measures. This arises due to the fact that small delays can result in multiple UAVs waiting for the $S(t)$ set at the same time and while doing this they all hold their corresponding $T_i(t)$ values at zero. In our case with 4 UAVs and 6 targets this leads to a performance improvement up to a certain magnitude of delay. If, however, there are many more tasks than UAVs, then generally it will be the case that increasing the delay will always lead to performance degradation. In summary, communication delays have a prominent influence in the degradation of the cooperative case to the noncooperative one. The above analysis could provide design guidelines for deciding when it is beneficial to cooperate and when it is not.

VII. ROLE IN GENERAL COOPERATIVE CONTROL PROBLEMS

In this section we explain the role of the cooperative control problem that we consider here in more general scenarios. In some cases we explain how our analysis can be extended for these more general problems, and in others we highlight additional modifications of the cooperative strategies that can be used to counteract the myopic nature of the strategies.

A. Maximizing UAV's Task Completion Rate

Fig. 8 depicts an opportunity to generalize the problem setting considered here by allowing the number of tasks to be time varying due to task arrivals and departures and due to UAVs assigning different priorities for the processing of tasks during the mission. Here, the group of UAVs can receive tasks to process from several sources: home base (e.g., humans demanding the processing of specific tasks), global sensors (e.g., satellites or high-flying platforms), and the UAV's on-board sensors. Some tasks are processed by UAVs and "depart" the system, while others that are processed generate more tasks

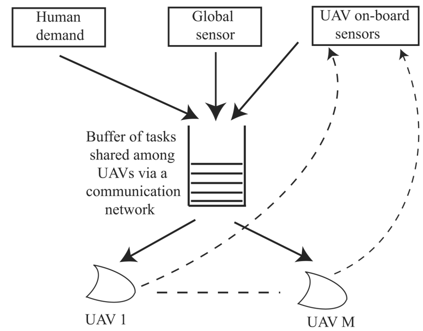

Fig. 8. Multiple tasks processed by M UAVs.

(e.g., when a search task is performed and a target is found, a classification task is generated).

The scenario considered here represents some progress towards solving this general problem. To see this, consider the case where the UAVs in Fig. 8 need to perform search, classify, attack, and verify operations for each task [8]. Suppose that each task in the buffer has a prioritized time T_i associated with it and that these are used to make scheduling decisions just like we do earlier. Suppose, however, that we allow the UAVs to manipulate the priority values associated with each such task during the mission. Initially, one UAV could have assigned a high priority value to a certain task that is considered of extreme importance for the mission; however, when this or another UAV performs classification, it may realize that the task is not what it was initially thought to be (e.g., not a valid target). A UAV then can assign a new priority value to tasks associated with that target. Define new variables in the model used here as the priority of searching $p_i^s(t)$, the priority of classifying $p_i^c(t)$, the priority of attacking $p_i^a(t)$, and the priority of verifying $p_i^v(t)$, for any task $i \in P$. If the number of tasks is fixed (i.e., there are no new task arrivals and departures to the buffer), the analysis performed in the Appendix still holds so long as the $p_i(t)$ values are bounded by a known constant. Moreover, even if the number of tasks $N(t)$ in the buffer changes over time, if we know that $N(t) \leq \bar{N} < \infty$ for a known $\bar{N}$, then we could use the same bound obtained in our main result by replacing N with $\bar{N}$ in this case. The problem is, however, that our framework is not easily extendable to the case where arbitrary task arrivals and departures can occur so we cannot quantify UAV system throughput (maximum task completion rate) in this general case.

B. Heuristic Modifications of Cooperative Scheduling Strategies

Next, some ideas for counteracting the effect of the myopic nature of the cooperative scheduling strategies defined in Section IVA are provided.

1) *Individual Decision Making by UAVs*: We can introduce the weights $\underline{w} \leq w_i^j(t) = \min_{j' \neq j}\{d(x_v^{j'}(t), x^i)\}/$

$d(x_v^j(t), \underline{x}^i) \leq \bar{w}$, $j \neq j'$, for all $j, j' \in Q$, $i \in P$ and redefine the cooperative scheduling strategies as

$$w_{i_j^*}^{j^*}(D_{kj}) \left(T_{i_j^*(kj)}(D_{kj}) - \frac{d(x_v^j(D_{kj}), x^{i_j^*(kj)})}{v} \right)$$
$$\geq \frac{\sum_{i_j \in U_j^a(D_{kj})} \left[w_{i_j}^j(D_{kj}) \left(T_{i_j}(D_{kj}) - d(x_v^j(D_{kj}), x^{i_j(kj)})/v \right) \right]}{N - M + 1}$$

and choose the same Lyapunov-like function to derive a different upper bound (i.e., this will include the parameters $\underline{w}$ and $\bar{w}$) on the ultimate longest time that any UAV will ignore task $i \in P$. The effect of these weights would be to attenuate the value from the expression

$$T_{i_j(k^j)}(D_{kj}) - \frac{d(x_v^j(D_{kj}), x^{i_j(k^j)})}{v}$$

if some UAV j' is considerably closer to task i than UAV j at time t (i.e., $w_i^j(t) \ll 1$ since $\min_{j' \neq j}\{d(x_v^{j'}(t), x^i)\} \ll d(x_v^j(t), x^i)$), then once UAV j is done with the processing of its current task, task i may not be a potential candidate to be chosen by UAV j, provided that there are other tasks very close to the current location of UAV j. The global effect of these weights could lead the UAVs to choose tasks with high priorities and minimal travel times located in the vicinity of the current location of the UAVs. Another way to interpret this effect is by noticing that UAV j is not interested in processing task i next since there is another UAV j' that is closer to this task and it might be processing task i in the near future.

2) *Passing the Request Queue Between UAVs*: The reason why UAVs are forced in Section IIIB to keep visiting the same task in the time interval $[t^f, t^f + \delta_c]$ while it waits for the set $S(t)$ is because 1) the number of tasks is fixed, and 2) we are studying surveillance-type problems in our framework. Note that we could overcome this "limitation" for a search, classify, attack, and verify scenario when the number of tasks are time varying as it is explained in Section VIIA. Thus, once a UAV finishes processing a task, it could broadcast a request for the set $S(t)$, but this time, this UAV could search for new tasks in the environment in the time interval $[t^f, t^f + \delta_c]$.

3) *Random Delays and Waiting time for Appearance of Pop-Up Tasks*: We could borrow ideas developed for flexible manufacturing systems to avoid long UAVs' waiting time to execute a particular task, which might result in poor mission performance. For instance, universally stabilizing supervisory mechanisms [32] can be utilized to decide for how long a UAV should wait for either the reception of the set $S(t)$ or for the appearance of a task.

4) *Preemptive Strategies*: A UAV could broadcast the set $S(t)$ to the rest of UAVs at any decision time, and at the time of reception the rest of UAVs could possibly switch tasks. In this way, revision of previous UAV decisions can be reconsidered as soon as new information received from other UAVs becomes available.

VIII. CONCLUDING REMARKS

We have derived stability (boundedness) conditions for network-based cooperative scheduling strategies that seek to optimize a cost function at each decision time when all UAVs know a priori detailed information about all tasks in a limited area. We have also shown in Monte Carlo simulations design guidelines for cooperative and noncooperative strategies based on task density and poor performance in the communication channels. For future work, we are most interested in addressing the problem of how to derive stability properties for maximizing UAVs' throughput when the number of tasks is time varying and when UAVs search, classify, attack, and verify tasks in a limited area as discussed in Section VII. In addition to this and the general idea of using FMS scheduling methods for UAV groups, there are a number of specific research directions that could be studied. For instance, in Section IVA we mentioned how it is possible to view the p_i values as controller design parameters that can be tuned to improve performance. Since (6) seeks to optimize two objectives, putting more emphasis in one variable may result in degradation of the performance metrics; hence, it would be useful to study optimization methods (off-line or on-line ones) that tune the p_i in order to achieve a better performance during the mission. Next, it might be worthwhile to study the possibility of obtaining tighter bounds on the ignored time of tasks when both scheduling strategies, cooperative and noncooperative, are combined. We have already seen in Section VI under which conditions it is beneficial for the UAVs to cooperate and when not to do it. The goal would be to define a switching strategy that commands the UAVs either to cooperate or not to cooperate based on measured delays in order to minimize the mission time.

APPENDIX. PROOF

In this appendix we provide the proof of the theorem stated in Section V. Let

$$V(t) = \sum_{i=1}^{N} T_i(t).$$

The proof to follow focuses on the strategy where the task i_j^* is chosen by UAV $j \in Q$. This proof proceeds by extending the one in [1]. There are, however, fundamental differences since the scheduling strategy is quite different (it represents a desire to achieve more than one objective and the values of the strategy can be positive or negative), and the slope of $T_i(t)$ could be different every time that any UAV is processing a task as pointed out in Section III whereas for the problem in [1] that slope is always constant. One implication of this last point will be that we do not need a "capacity condition" as we do in [1].

First, note that

$$V(t) = \sum_{j=1}^{M} \left(T_{i_j^*}(t) + \sum_{i \in U(t)} \frac{T_i(t)}{M} \right). \quad (8)$$

Define the function $V_j(t)$ for UAV j as

$$V_j(t) = T_{i_j^*}(t) + \sum_{i \in U(t)} \frac{T_i(t)}{M}. \quad (9)$$

Consider the values at the set of decision times D_{k^j},

$$\sum_{j=1}^{M} V_j(D_{k^j}) = \sum_{j=1}^{M} \left[T_{i_j^*(k^j)}(D_{k^j}) + \sum_{i \in U(D_{k^j})} \frac{T_i(D_{k^j})}{M} \right]. \quad (10)$$

Since $T_{i_j^*(k^j)}(D_{k^j+1}) = 0$, $j \in Q$ (where $i_j^*(k^j)$ was the task that was just processed by UAV $j \in Q$),

$$\sum_{j=1}^{M} V_j(D_{k^j+1}) = \sum_{j=1}^{M} \left[\sum_{i \in U(D_{k^j+1})} \frac{T_i(D_{k^j+1})}{M} \right]. \quad (11)$$

Note that by definition $U(D_{k^j+1}) = U(D_{k^{j^c}})$ since the unattended set will not change for every UAV j for t such that $D_{k^{j^c}} \le t \le D_{k^j+1}$. Hence, considering how long the tasks have been ignored during this time period,

$$\sum_{j=1}^{M} V_j(D_{k^j+1}) = \sum_{j=1}^{M} \left\{ \sum_{i \in U(D_{k^{j^c}})} \left[\frac{T_i(D_{k^{j^c}})}{M} + \frac{p_i}{M}(D_{k^j+1} - D_{k^{j^c}}) \right] \right\}. \quad (12)$$

From (9) for any UAV j^c at time $D_{k^{j^c}}$

$$\sum_{i \in U(D_{k^{j^c}})} \frac{T_i(D_{k^{j^c}})}{M} = V_{j^c}(D_{k^{j^c}}) - T_{i_{j^c}^*(k^{j^c})}(D_{k^{j^c}}). \quad (13)$$

We use (3), (12), and (13) to obtain

$$\sum_{j=1}^{M} V_j(D_{k^j+1}) \le \sum_{j=1}^{M} \left\{ V_{j^c}(D_{k^{j^c}}) - T_{i_{j^c}^*(k^{j^c})}(D_{k^{j^c}}) + \sum_{i \in U(D_{k^{j^c}})} \frac{p_i}{M} \left(\bar{\delta} + \frac{d(x_v^{j^c}(D_{k^{j^c}}), x^{i_{j^c}^*(k^{j^c})})}{v} + \bar{\tau} + \bar{\delta}_c \right) \right\}$$

$$= \sum_{j=1}^{M} \left\{ V_{j^c}(D_{k^{j^c}}) - \left(T_{i_{j^c}^*(k^{j^c})}(D_{k^{j^c}}) - \frac{d(x_v^{j^c}(D_{k^{j^c}}), x^{i_{j^c}^*(k^{j^c})})}{v} \sum_{i \in U(D_{k^{j^c}})} \frac{p_i}{M} \right) + (\bar{\delta} + \bar{\tau} + \bar{\delta}_c) \sum_{i \in U(D_{k^{j^c}})} \frac{p_i}{M} \right\}. \quad (14)$$

Focus now on the first $\sum_{i \in U(D_{k^{j^c}})} p_i/M$ term in (14) and notice that the proof can be divided in two cases as follows: a) When $\sum_{i \in U(D_{k^{j^c}})} p_i \le M, k^{j^c}$, and b) when $\sum_{i \in U(D_{k^{j^c}})} p_i > M, k^{j^c}$.

Case a To start, we seek to remove the first $\sum_{i \in U(D_{k^{j^c}})} p_i/M$ term in (14) in order to make the term in parenthesis the same as our strategy in (6). Note that

$$\sum_{j=1}^{M} V_j(D_{k^j+1}) \le \sum_{j=1}^{M} \left\{ V_{j^c}(D_{k^{j^c}}) - \left(T_{i_{j^c}^*(k^{j^c})}(D_{k^{j^c}}) - \frac{d(x_v^{j^c}(D_{k^{j^c}}), x^{i_{j^c}^*(k^{j^c})})}{v} \right) + (\bar{\delta} + \bar{\tau} + \bar{\delta}_c) \sum_{i \in U(D_{k^{j^c}})} \frac{p_i}{M} \right\}. \quad (15)$$

We use the definition of the cooperative scheduling strategy for $j = j^c$ in (15)

$$T_{i_{j^c}^*(k^{j^c})}(D_{k^{j^c}}) - \frac{d(x_v^{j^c}(D_{k^{j^c}}), x^{i_{j^c}^*(k^{j^c})})}{v} \ge \frac{1}{N - M + 1} \sum_{i_j \in U_j^a(D_{k^{j^c}})} \times \left[T_{i_j}(D_{k^{j^c}}) - \frac{d(x_v^{j^c}(D_{k^{j^c}}), x^{i_j(k^{j^c})})}{v} \right].$$

But notice that

$$T_{i_{j^c}^*(k^{j^c})}(D_{k^{j^c}}) - \frac{d(x_v^{j^c}(D_{k^{j^c}}), x^{i_{j^c}^*(k^{j^c})})}{v} \ge \frac{V_{j^c}(D_{k^{j^c}})}{(N - M + 1)} - \frac{1}{N - M + 1} \sum_{i_j \in U_j^a(D_{k^{j^c}})} \times \left[\frac{d(x_v^{j^c}(D_{k^{j^c}}), x^{i_j(k^{j^c})})}{v} \right].$$

Combine this with (15) to get

$$\sum_{j=1}^{M} V_j(D_{k^j+1}) \le \sum_{j=1}^{M} \left\{ V_{j^c}(D_{k^{j^c}}) \left(1 - \frac{1}{N-M+1}\right) + \frac{1}{N-M+1} \times \sum_{i_j \in U_j^a(D_{k^{j^c}})} \left[\frac{d(x_v^{j^c}(D_{k^{j^c}}), x^{i_j(k^{j^c})})}{v} \right] + (\bar{\delta} + \bar{\tau} + \bar{\delta}_c) \sum_{i \in U(D_{k^{j^c}})} \frac{p_i}{M} \right\}. \tag{16}$$

Also

$$\sum_{j=1}^{M} V_j(D_{k^j+1}) \le \sum_{j=1}^{M} \left\{ V_{j^c}(D_{k^{j^c}}) \left(1 - \frac{1}{N-M+1}\right) + \frac{\bar{d}}{v} + (\bar{\delta} + \bar{\tau} + \bar{\delta}_c) \left(\sum_{i=1}^{N} \frac{p_i}{M} - \underline{p} \right) \right\} \tag{17}$$

since

$$\sum_{i \in U(D_{k^{j^c}})} \frac{p_i}{M} = \sum_{i=1}^{N} \frac{p_i}{M} - \sum_{i \in A(D_{k^{j^c}})} \frac{p_i}{M} \le \sum_{i=1}^{N} \frac{p_i}{M} - M\frac{\underline{p}}{M}. \tag{18}$$

Notice here that we can also obtain a tighter bound on the right side of (18) by using the M minimum values of p_i rather than $M\underline{p}$.

Define

$$\beta = \frac{\bar{d}}{v} + (\bar{\delta} + \bar{\tau} + \bar{\delta}_c) \left(\sum_{i=1}^{N} \frac{p_i}{M} - \underline{p} \right).$$

Notice that $\beta > 0$ and that

$$\sum_{j=1}^{M} V_j(D_{k^j+1}) \le \sum_{j=1}^{M} \left\{ V_{j^c}(D_{k^{j^c}}) \left(1 - \frac{1}{N-M+1}\right) + \beta \right\}.$$

But, notice that

$$\sum_{j=1}^{M} V_j(D_{k^j+1}) \le \left(1 - \frac{1}{N-M+1}\right) \sum_{j=1}^{M} \{V_{j^c}(D_{k^{j^c}})\} + M\beta. \tag{19}$$

This means that we have a contractive mapping in (19). Notice, however, that in (19) we have on the left-hand side $V_j(D_{k^j+1})$ and on the right $V_{j^c}(D_{k^{j^c}})$ so the mapping is contractive as we go from k^{j^c} to $k^j + 1$, $j \in Q$. The sums in (19) account for all time so that the contractive mapping is valid for all $t \ge 0$. To explain why, recall how we defined j^c: Given a j we have some D_{k^j+1} and from that value we define j^c as the index of the UAV that most recently finished its processing at time $t < D_{k^j+1}$. So given any D_{k^j+1} we can always find a j^c (at $t = 0$ we consider all UAVs to have just finished processing to get $T_{i_j}(0) = 0$). Any time t such that $T_{i_j}(t)$ becomes zero and after having received the unattended set it is labeled as D_{k^j+1}. Then the time range $[D_{k^{j^c}}, D_{k^j+1}]$ is considered in the mapping since the left-hand side of (19) is evaluated at the right side of this interval for one j and the right side of (19) is evaluated at the left side of the interval. Now, the key is to note that we can relabel this j^c (and hence $D_{k^{j^c}}$) as $D_{k^{j'}+1}$ since it was the time that UAV j^c finished so it is the new decision time for the UAV. Then there is of course a j'^c, and so forth, so that all time intervals are considered in the contractive mapping. Note that this accounts for the fact that the sums in the contraction are based on a set of times that are not necessarily contiguous. Note also that in (19) it is not possible that $V_j(D_{k^j+1}) = V_{j^c}(D_{k^{j^c}})$ for all j since $\delta_c > 0$ so that $D_{k^{j^c}} < D_{k^j+1}$ by definition. Also it is not possible that for all j, $V_j(D_{k^j+1}) \to V_{j^c}(D_{k^{j^c}})$ as $k^j \to \infty$. While this may happen for some j values, it cannot happen for all such values. Note that if for $M-1$ UAVs $V_j(D_{k^j+1}) \to V_{j^c}(D_{k^{j^c}})$, then for the remaining UAV, say j', $D_{k^{j'c}} < D_{k^{j'}+1}$, $V_{j'}(D_{k^{j'}+1}) \not\to V_{j'c}(D_{k^{j'c}})$.

Define for $k \ge 0$

$$\bar{V}(k) = \sum_{j=1}^{M} V_{j^c}(D_{k^{j^c}})$$

and

$$\bar{V}(k+1) = \sum_{j=1}^{M} V_j(D_{k^j+1})$$

so that $\bar{V}(k+1)$ is the sum at the set of times when all UAVs have already finished processing their respective tasks and have received $U(D_{k^{j^c}})$ in order to choose which tasks they will process next. Now, we use $\bar{V}(k)$ and $\bar{V}(k+1)$ in (19) to get

$$\bar{V}(k+1) \le \gamma \bar{V}(k) + \zeta \tag{20}$$

where $0 < \gamma = (1 - 1/(N-M+1)) < 1$ and $\zeta = M\beta$ which are both constants. Equation (20) is a difference inequality with a solution that is bounded for all k by

$$\bar{V}(k) \le \left(\bar{V}(0) - \frac{\zeta}{1-\gamma}\right)\gamma^k + \frac{\zeta}{1-\gamma}. \tag{21}$$

Notice that if $\bar{V}(0) > \zeta/(1-\gamma)$ ($\bar{V}(0) < \zeta/(1-\gamma)$) then since $\gamma^k \to 0$ as $k \to \infty$, $\bar{V}(k)$ decreases (increases) to $\zeta/(1-\gamma)$ as $k \to \infty$. Now,

$$\frac{\zeta}{1-\gamma} = M(N-M+1)\beta. \tag{22}$$

This gives us a bound on the transient and ultimate bound $\bar{V}(k)$ values as $k \to \infty$, at the decision times. Next, we need to consider the times in between the decision times. To do this, note that for all j^c, k^{j^c}, and k

$$V_{j^c}(D_{k^{j^c}}) \le \bar{V}(k). \tag{23}$$

Next, consider the case where $V_{j^c}(D_{k^{j^c}} + \delta_{i,j^c(k^{j^c})} + d(x_v^{j^c}(D_{k^{j^c}}), x^{i^*_{j^c}(k^{j^c})})/v)$ occurs at any t, $D_{k^{j^c}} < t < D_{k^j+1}$. If this is the case, then for any $T_i(t) \le f_\ell(t)$ and any t, $D_{k^{j^c}} \le t \le D_{k^{j^c}} + \delta_{i,j^c(k^{j^c})} + d(x_v^{j^c}(D_{k^{j^c}}), x^{i^*_{j^c}(k^{j^c})})/v$ we have

$$V_{j^c}(t) \le V_{j^c}\left(D_{k^{j^c}} + \delta_{i,j^c(k^{j^c})} + \frac{d(x_v^{j^c}(D_{k^{j^c}}), x^{i^*_{j^c}(k^{j^c})})}{v}\right). \tag{24}$$

On the other hand, for any t, $D_{k^{j^c}} + \delta_{i,j^c(k^{j^c})} + d(x_v^{j^c}(D_{k^{j^c}}), x^{i^*_{j^c}(k^{j^c})})/v \le t \le D_{k^j+1}$ if we let $\gamma = \delta_{i,j^c(k^{j^c})} + d(x_v^{j^c}(D_{k^{j^c}}), x^{i^*_{j^c}(k^{j^c})})/v$ we have

$$\begin{aligned}
&V_{j^c}(D_{k^j+1})\\
&= T_{i^*_{j^c}(k^{j^c})}(D_{k^j+1}) + \sum_{i\in U(D_{k^j+1})} \frac{T_i(D_{k^j+1})}{M}\\
&= T_{i^*_{j^c}(k^{j^c})}(D_{k^{j^c}}+\gamma) - \frac{T_{i^*_{j^c}(k^{j^c})}(D_{k^{j^c}}+\gamma)(D_{k^j+1} - D_{k^{j^c}} - \gamma)}{\tau_{i^*_{j^c}(k^{j^c})}}\\
&\quad + \sum_{i\in U(D_{k^{j^c}}+\gamma)} \frac{T_i(D_{k^{j^c}}+\gamma)}{M} + \sum_{i\in U(D_{k^{j^c}}+\gamma)} \frac{p_i(D_{k^j+1} - D_{k^{j^c}} - \gamma)}{M}\\
&= V_{j^c}(D_{k^{j^c}}+\gamma) + \underbrace{(D_{k^j+1} - D_{k^{j^c}} - \gamma)}_{>0}\\
&\quad \times \left(\sum_{i\in U(D_{k^{j^c}}+\gamma)} \frac{p_i}{M} - \frac{T_{i^*_{j^c}(k^{j^c})}(D_{k^{j^c}}+\gamma)}{\tau_{i^*_{j^c}(k^{j^c})}}\right).
\end{aligned} \tag{25}$$

Here, we have two possibilities as follows.

Case a.1 If the slope of the task currently being processed is greater or equal to the sum of the slopes of the unattended tasks over M, i.e.,

$$\frac{T_{i^*_{j^c}(k^{j^c})}(D_{k^{j^c}}+\gamma)}{\tau_{i^*_{j^c}(k^{j^c})}} \ge \sum_{i\in U(D_{k^{j^c}}+\gamma)} \frac{p_i}{M}, k^{j^c}$$

then $V_{j^c}(D_{k^j+1}) \le V_{j^c}(D_{k^{j^c}}+\gamma)$. Thus, we obtain that $V_{j^c}(t) \le V_{j^c}(D_{k^{j^c}}+\gamma)$ for any t, $D_{k^{j^c}} \le t \le D_{k^j+1}$.

For this case then

$$\begin{aligned}
&V_{j^c}(D_{k^{j^c}}+\gamma)\\
&= T_{i^*_{j^c}(k^{j^c})}(D_{k^{j^c}}+\gamma) + \sum_{i\in U(D_{k^{j^c}}+\gamma)} \frac{T_i(D_{k^{j^c}}+\gamma)}{M}\\
&= T_{i^*_{j^c}(k^{j^c})}(D_{k^{j^c}}) + \bar{\delta} p_{i^*_{j^c}(k^{j^c})} + \frac{d(x_v^{j^c}(D_{k^{j^c}}), x^{i^*_{j^c}(k^{j^c})}) p_{i^*_{j^c}(k^{j^c})}}{v}\\
&\quad + \sum_{i\in U(D_{k^{j^c}})} \frac{T_i(D_{k^{j^c}})}{M} + \sum_{i\in U(D_{k^{j^c}})} \frac{p_i\bar{\delta}}{M}\\
&\quad + \sum_{i\in U(D_{k^{j^c}})} \frac{d(x_v^{j^c}(D_{k^{j^c}}), x^{i^*_{j^c}(k^{j^c})}) p_i}{vM}\\
&= V_{j^c}(D_{k^{j^c}}) + \bar{\delta}\left(p_{i^*_{j^c}(k^{j^c})} + \sum_{i\in U(D_{k^{j^c}})} \frac{p_i}{M}\right)\\
&\quad + \frac{d(x_v^{j^c}(D_{k^{j^c}}), x^{i^*_{j^c}(k^{j^c})})}{v}\left(p_{i^*_{j^c}(k^{j^c})} + \sum_{i\in U(D_{k^{j^c}})} \frac{p_i}{M}\right)\\
&\le V_{j^c}(D_{k^{j^c}}) + \left(\bar{\delta} + \frac{\bar{d}}{v}\right)\left(\bar{p} + \sum_{i=1}^{N} \frac{p_i}{M} - \underline{p}\right)
\end{aligned} \tag{26}$$

where both the second and last terms of the right-hand side of the equation were derived by using (18).

Next, note that for all j^c and $t \ge 0$

$$T_{i^*_{j^c}}(t) + \sum_{i\in U(t)} \frac{T_i(t)}{M} = V_{j^c}(t)$$

so that using (24) and (26)

$$T_{i^*_{j^c}}(t) + \sum_{i\in U(t)} \frac{T_i(t)}{M} \le V_{j^c}(D_{k^{j^c}}) + \left(\bar{\delta} + \frac{\bar{d}}{v}\right)\left(\bar{p} + \sum_{i=1}^{N} \frac{p_i}{M} - \underline{p}\right).$$

Hence, using (22)

$$\begin{aligned}
\lim_{t\to\infty} t_{i^*_{j^c}}(t) &\le \lim_{t\to\infty}\left[t_{i^*_{j^c}}(t) + \sum_{i\in U(t)} \frac{t_i(t)}{M}\right]\\
&\le \frac{M}{\underline{p}}(N-M+1)\beta + \left(\frac{\bar{\delta}}{\underline{p}} + \frac{\bar{d}}{\underline{p}v}\right)\left(\bar{p} + \sum_{i=1}^{N} \frac{p_i}{M} - \underline{p}\right)\\
&\le \frac{M}{\underline{p}}(N-M+1)\left(\frac{\bar{d}}{v} + (\bar{\delta} + \bar{\tau} + \bar{\delta}_c)\left(\sum_{i=1}^{N} \frac{p_i}{M} - \underline{p}\right)\right)\\
&\quad + \left(\frac{\bar{\delta}}{\underline{p}} + \frac{\bar{d}}{\underline{p}v}\right)\left(\bar{p} + \sum_{i=1}^{N} \frac{p_i}{M} - \underline{p}\right)
\end{aligned}$$

$$\leq \frac{(\bar{\delta}+\bar{\tau}+\bar{\delta}_c)}{\underline{p}}\left(\sum_{i=1}^{N} p_i - M\underline{p}\right)(N-M+1)$$

$$+\frac{\bar{d}}{\underline{p}v}\left(M(N-M+1)+\bar{p}+\sum_{i=1}^{N}\frac{p_i}{M}-\underline{p}\right)$$

$$+\frac{\bar{\delta}}{\underline{p}}\left(\bar{p}+\sum_{i=1}^{N}\frac{p_i}{M}-\underline{p}\right). \quad (27)$$

Now, we must show that each task will get chosen by some UAV $j \in Q$ infinitely often so that every task becomes $i^*_{j^c}$ persistently so that (27) provides a bound for each $t_i(t)$ $i \in P$. Note that we have a bound for every k^{j^c} for every $t_{i^*_{j^c}(k^{j^c})}(D_{k^{j^c}})$ via (27), and using (3), $D_{k^j+1} - D_{k^{j^c}}$ is bounded. This results in a maximum bound on the time that the unattended set will not be changed. "Ignored time" for tasks rises so eventually any ignored task in $U(t)$ will be taken off $U(t)$ and hence become $i^*_{j^c}$.

Case a.2 Now, we study the case when $T_{i^*_{j^c}(k^{j^c})}(D_{k^{j^c}}+\gamma)/\tau_{i^*_{j^c(k^{j^c})}} < \sum_{i\in U(D_{k^{j^c}+\gamma})} p_i/M$, for some k^{j^c}. If this is the case, then $V_{j^c}(D_{k^j+1}) > V_{j^c}(D_{k^{j^c}}+\gamma)$. Thus, we obtain that $V_{j^c}(t) \leq V_{j^c}(D_{k^j+1})$ for any t, $D_{k^{j^c}} \leq t \leq D_{k^j+1}$. However, this cannot always be the case because if $V_{j^c}(D_{k^j+1}) > V_{j^c}(D_{k^{j^c}}+\gamma)$, k^{j^c}, then the function V_{j^c} is increasing all the time, which means that all T_is in the unattended set grow much more faster than the value of $T_{i^*_{j^c}}$ being currently processed. But if this is the case, then sooner or later the condition $T_{i^*_{j^c}(k^{j^c})}(D_{k^{j^c}}+\gamma)/\tau_{i^*_{j^c(k^{j^c})}} \geq \sum_{i\in U(D_{k^{j^c}+\gamma})} p_i/M$ will be satisfied since any ignored task will later become $i^*_{j^c}$ and this is the reason why we say that $T_{i^*_{j^c}(k^{j^c})}(D_{k^{j^c}}+\gamma)/\tau_{i^*_{j^c(k^{j^c})}} < \sum_{i\in U(D_{k^{j^c}+\gamma})} p_i/M$ holds for some k^{j^c}.

Now, for this case

$$V_{j^c}(D_{k^j+1}) = T_{i^*_{j^c}(k^{j^c})}(D_{k^j+1}) + \sum_{i\in U(D_{k^j}+1)} \frac{T_i(D_{k^j+1})}{M}$$

$$= T_{i^*_{j^c}(k^{j^c})}(D_{k^{j^c}}) + \gamma p_{i^*_{j^c}(k^{j^c})} - (D_{k^j+1} - D_{k^{j^c}} - \gamma) \times \frac{T_{i^*_{j^c}(k^{j^c})}(D_{k^j+1} - D_{k^{j^c}} - \gamma)}{\tau_{i^*_{j^c}(k^{j^c})}}$$

$$+ \sum_{i\in U(D_{k^{j^c}})} \frac{T_i(D_{k^{j^c}})}{M} + (D_{k^j+1} - D_{k^{j^c}}) \sum_{i\in U(D_{k^{j^c}})} \frac{p_i}{M}$$

$$\leq V_{j^c}(D_{k^{j^c}}) + \gamma p_{i^*_{j^c}(k^{j^c})} + (D_{k^j+1} - D_{k^{j^c}}) \sum_{i\in U(D_{k^{j^c}})} \frac{p_i}{M}$$

$$\leq V_{j^c}(D_{k^{j^c}}) + \bar{p}\left(\bar{\delta}+\frac{\bar{d}}{v}\right)$$

$$+\left(\bar{\delta}+\bar{\tau}+\bar{\delta}_c+\frac{\bar{d}}{v}\right)\left(\sum_{i=1}^{N}\frac{p_i}{M}-\underline{p}\right). \quad (28)$$

Next, we apply the same steps taken in (27) to obtain that

$$\lim_{t\to\infty} t_{i^*_{j^c}}(t) \leq \frac{M}{\underline{p}}(N-M+1)\beta + \frac{\bar{p}}{\underline{p}}\left(\bar{\delta}+\frac{\bar{d}}{v}\right)$$

$$+\frac{(\bar{\delta}+\bar{\tau}+\bar{\delta}_c+\bar{d}/v)}{\underline{p}}\left(\sum_{i=1}^{N}\frac{p_i}{M}-\underline{p}\right)$$

$$\leq \frac{M}{\underline{p}}(N-M+1)\left(\frac{\bar{d}}{v}+(\bar{\delta}+\bar{\tau}+\bar{\delta}_c)\left(\sum_{i=1}^{N}\frac{p_i}{M}-\underline{p}\right)\right)$$

$$+\frac{\bar{p}}{\underline{p}}\left(\bar{\delta}+\frac{\bar{d}}{v}\right)+\frac{(\bar{\delta}+\bar{\tau}+\bar{\delta}_c+\bar{d}/v)}{\underline{p}}\left(\sum_{i=1}^{N}\frac{p_i}{M}-\underline{p}\right)$$

$$\leq \frac{(\bar{\delta}+\bar{\tau}+\bar{\delta}_c)}{\underline{p}}\left(\sum_{i=1}^{N}\frac{p_i}{M}-\underline{p}\right)(NM-M^2+M+1)$$

$$+\frac{\bar{d}}{\underline{p}v}\left(M(N-M+1)+\bar{p}+\sum_{i=1}^{N}\frac{p_i}{M}-\underline{p}\right)+\frac{\bar{\delta}}{\underline{p}}\bar{p} \quad (29)$$

where it can be seen that the bound obtained in (29) is greater than the one shown in (27) due to the extra parameters in the first and last term in (29). Notice that (29) is equal to the variable B^1 shown in the statement of the theorem in Section V.

Case b Consider $\sum_{i\in U(D_{k^{j^c}})} p_i > M, k^{j^c}$. Note that for this case we have

$$T_{i^*_{j^c}(k^{j^c})}(D_{k^{j^c}}) - \frac{d(x_v^{j^c}(D_{k^{j^c}}), x^{i^*_{j^c}(k^{j^c})})}{v}$$

$$> T_{i^*_j(k^j)}(D_{k^j}) - \frac{d(x_v^j(D_{k^j}), x^{i^*_j(k^j)})}{v} \sum_{i\in U(D_{k^j})} \frac{p_i}{M} \quad (30)$$

and both the left and the right-hand side of (30) are greater or equal to

$$\frac{1}{N-M+1}\sum_{i_j\in U_j^a(D_{k^{j^c}})}\left[T_{i_j}(D_{k^{j^c}}) - \frac{d(x_v^{j^c}(D_{k^{j^c}}), x^{i_j(k^{j^c})})}{v}\sum_{i\in U(D_{k^{j^c}})}\frac{p_i}{M}\right].$$

If we use the above result in (14) then we are solving the problem for the strategy defined in (6). The final result will be, of course, more conservative. As in the last case, we have here two possibilities and we study them next.

Case b.1 If $T_{i^*_{j^c}(k^{j^c})}(D_{k^{j^c}}+\gamma)/\tau_{i^*_{j^c}(k^{j^c})} \geq \sum_{i\in U(D_{k^{j^c}}+\gamma)} p_i/M, k^{j^c}$, then $V_{j^c}(D_{k^j+1}) \leq V_{j^c}(D_{k^{j^c}}+\gamma)$. Thus, we obtain that $V_{j^c}(t) \leq V_{j^c}(D_{k^{j^c}}+\gamma)$ for any t, $D_{k^{j^c}} \leq t \leq D_{k^j+1}$.

Note that for this case (22) and (23) still hold except that β is different from the one obtained in case a. Furthermore, we have already derived in (26) that

$$V_{j^c}\left(D_{k^{j^c}} + \delta_{i,j^c(k^{j^c})} + \frac{d(x_v^{j^c}(D_{k^{j^c}}), x^{i^*_{j^c}(k^{j^c})})}{v}\right) \leq V_{j^c}(D_{k^{j^c}}) + \left(\bar{\delta} + \frac{\bar{d}}{v}\right)\left(\bar{p} + \sum_{i=1}^{N}\frac{p_i}{M} - \underline{p}\right).$$

Now, the procedure to obtain a bound for this case is the same as the one followed in Case a.1 except that for this particular case we have

$$\beta = \frac{\bar{d}}{v}\left(\sum_{i=1}^{N}\frac{p_i}{M} - \underline{p}\right) + (\bar{\delta} + \bar{\tau} + \bar{\delta}_c)\left(\sum_{i=1}^{N}\frac{p_i}{M} - \underline{p}\right)$$

where the term that multiplies to $\bar{d}/v$ in the first term of the right-hand side of the above equation is derived from (18). Notice that this term was not present in case a) since $\sum_{i\in U(D_{k^{j^c}})} p_i/M < 1$.

Hence,

$$\begin{aligned}\lim_{t\to\infty} t_{i^*_{j^c}}(t) &\leq \frac{(N-M+1)}{\underline{p}}\left(\sum_{i=1}^{N} p_i - M\underline{p}\right)\left(\frac{\bar{d}}{v} + \bar{\delta} + \bar{\tau} + \bar{\delta}_c\right) + \frac{(\bar{d}/v + \bar{\delta})}{\underline{p}}\left(\bar{p} + \sum_{i=1}^{N}\frac{p_i}{M} - \underline{p}\right)\\ &\leq \frac{(\bar{\delta} + \bar{\tau} + \bar{\delta}_c)}{\underline{p}}\left(\sum_{i=1}^{N} p_i - M\underline{p}\right)(N-M+1) + \frac{\bar{d}}{\underline{p}v}\left(\sum_{i=1}^{N} p_i - M\underline{p}\right)\left(N-M+1+\frac{1}{M}\right)\\ &\quad + \frac{\bar{\delta}}{\underline{p}}\left(\bar{p} + \sum_{i=1}^{N}\frac{p_i}{M} - \underline{p}\right) + \frac{\bar{p}\bar{d}}{\underline{p}v}. \end{aligned} \tag{31}$$

Case b.2 If $T_{i^*_{j^c}(k^{j^c})}(D_{k^{j^c}} + \gamma)/\tau_{i^*_{j^c(k^{j^c})}} < \sum_{i\in U(D_{k^{j^c}+\gamma})} p_i/M$, for some k^{j^c}, then $V_{j^c}(D_{k^j+1}) > V_{j^c}(D_{k^{j^c}} + \gamma)$. Thus, we obtain that $V_{j^c}(t) \leq V_{j^c}(D_{k^j+1})$ for any t, $D_{k^{j^c}} \leq t \leq D_{k^j+1}$. Therefore, by using both the same arguments and (28) as in Case a.2 we obtain that

$$V_{j^c}(D_{k^j}+1) \leq V_{j^c}(D_{k^{j^c}}) + \bar{p}\left(\bar{\delta} + \frac{\bar{d}}{v}\right) + \left(\bar{\delta} + \bar{\tau} + \bar{\delta}_c + \frac{\bar{d}}{v}\right)\left(\sum_{i=1}^{N}\frac{p_i}{M} - \underline{p}\right).$$

Hence,

$$\begin{aligned}\lim_{t\to\infty} t_{i^*_{j^c}}(t) &\leq \frac{M}{\underline{p}}(N-M+1)\beta + \frac{\bar{p}}{\underline{p}}\left(\bar{\delta} + \frac{\bar{d}}{v}\right) + \frac{(\bar{\delta} + \bar{\tau} + \bar{\delta}_c + \bar{d}/v)}{\underline{p}}\left(\sum_{i=1}^{N}\frac{p_i}{M} - \underline{p}\right)\\ &\leq \frac{M}{\underline{p}}(N-M+1)\left[\left(\frac{\bar{d}}{v} + \bar{\delta} + \bar{\tau} + \bar{\delta}_c\right)\left(\sum_{i=1}^{N}\frac{p_i}{M} - \underline{p}\right)\right]\\ &\quad + \frac{\bar{p}}{\underline{p}}\left(\bar{\delta} + \frac{\bar{d}}{v}\right) + \frac{(\bar{\delta} + \bar{\tau} + \bar{\delta}_c + \bar{d}/v)}{\underline{p}}\left(\sum_{i=1}^{N}\frac{p_i}{M} - \underline{p}\right)\\ &\leq \frac{(\bar{\delta} + \bar{\tau} + \bar{\delta}_c + \bar{d}/v)}{\underline{p}}\left(\sum_{i=1}^{N}\frac{p_i}{M} - \underline{p}\right) \times (NM - M^2 + M + 1) + \frac{\bar{p}}{\underline{p}}\left(\bar{\delta} + \frac{\bar{d}}{v}\right) \end{aligned} \tag{32}$$

where we can see that the bound obtained in (32) is greater than the one obtained in (31). Notice that (32) is equal to the variable B^2 shown in the statement of the theorem in Section V.

On the other hand, it can be easily seen that if $V_{j^c}(D_{k^{j^c}} + \gamma)$ does not occur at any t, $D_{k^{j^c}} \leq t \leq D_{k^j+1}$, then this cannot happen all the time since there exists a time interval different from $D_{k^{j^c}} \leq t \leq D_{k^j+1}$ where this event will take place. Note that we do not need to study this particular case since this is a special case of the ones studied in Cases a.2 and b.2. This concludes the proof of the theorem stated in Section V.

REFERENCES

[1] Gil, A. E., and Passino, K. M.
Stability analysis of network-based cooperative resource allocation strategies.
In *Proceedings of the IEEE Conference on Decision and Control*, Maui, HI, Dec. 2003, 1206–1211.

[2] Gil, A. E., Passino, K. M., Ganapathy, S., and Sparks, A.
Cooperative scheduling of tasks for networked uninhabited autonomous vehicles.
In *Proceedings of the IEEE Conference on Decision and Control*, Maui, HI, Dec. 2003, 522–527.

[3] Laporte, G.
The vehicle routing problem: An overview of the exact and approximate algorithms.
European Journal of Operational Research, **59** (1992), 345–358.

[4] Cassandras, C. G., and Li, W.
A receding horizon approach for solving some cooperative control problems.
In *Proceedings of the IEEE Conference on Decision and Control*, Las Vegas, NV, Dec. 2002, 3760–3765.

[5] Castanon, D. A., and Cassandras, C. G.
Cooperative mission control for unmanned air vehicles.
In *Proceedings of the AFOSR Workshop on Dynamic Systems and Control*, Pasadena, CA, Aug. 2002, 57–60.

[6] Bellingham, J., Richards, A., and How, J.
Receding horizon control of autonomous aerial vehicles.
In *Proceedings of the American Control Conference*, Anchorage, AK, May 2002, 3741–3746.

[7] Li, W., and Cassandras, C. G.
Stability properties of a cooperative receding horizon controller.
In *Proceedings of the IEEE Conference on Decision and Control*, Maui, HI, Dec. 2003, 492–497.

[8] Chandler, P. R., Pachter, M., and Rasmussen, S.
UAV cooperative control.
In *Proceedings of the American Control Conference*, Arlington, VA, June 2001, 50–55.
[9] Beard, R., McLain, T., and Goodrich, M.
Coordinated target assignment and intercept for unmanned air vehicles.
In *Proceedings of IEEE International Conference on Robotics and Automation*, Washington, D.C., May 2002, 2581–2586.
[10] Chandler, P., et al.
Complexity in UAV cooperative control.
In *Proceedings of the American Control Conference*, Anchorage, AK, May 2002, 1831–1836.
[11] Jin, Y., Minai, A. A., and Polycarpou, M. M.
Cooperative real-time search and task allocation in UAV teams.
In *Proceedings of the IEEE Conference on Decision and Control*, Maui, HI, Dec. 2003, 7–12.
[12] Passino, K. M., Polycarpou, M., Jacques, D., Pachter, M., Liu, Y., Yang, Y., Flint, M., and Baum, M.
Cooperative control for autonomous air vehicles.
In R. Murphey and P. M. Pardalos (Eds.), *Cooperative Control and Optimization*, vol. 66, Boston: Kluwer Academic Publishers, 2002, 233–271.
[13] Baum, M., and Passino, K.
A search-theoretic approach to cooperative control for uninhabited air vehicles.
Presented at AIAA GNC Conference, Monterrey, CA, Aug. 2001.
[14] Hespanha, J., and Kizilocak, H.
Efficient computation of dynamic probabilistic maps.
In *Proceedings of the 10th Mediterranean Conference on Control and Automation*, Lisbon, Portugal, July 2002.
[15] Hespanha, J., Kizilocak, H., and Ateskan, Y.
Probabilistic map building for aircraft-tracking radars.
In *Proceedings of the American Control Conference*, Arlington, VA, June 2001, 4381–4386.
[16] Mahajan, A., Ko, J., and Sengupta, R.
Distributed probabilistic map service.
In *Proceedings of the IEEE Conference on Decision and Control*, Las Vegas, NV, Dec. 2002, 130–135.
[17] Ganapathy, S., and Passino, K.
Agreement strategies for cooperative control of uninhabited autonomous vehicles.
In *Proceedings of the 2003 American Control Conference*, Denver, CO, June 2003, 1026–1031.
[18] Jun, M., Chaudhly, A. I., and D'Andrea, R.
The navigation of autonomous vehicles in uncertain dynamic environments: A case study.
In *Proceedings of the IEEE Conference on Decision and Control*, Las Vegas, NV, Dec. 2002, 3770–3775.
[19] Ganapathy, S., and Passino, K.
Distributed agreement strategies for cooperative control: Modeling and scalability analysis.
In *Proceedings of the Conference on Cooperative Control and Optimization*, Gainsville, FL, Dec. 2002, 127–147.
[20] Castanon, D. A., and Wu, C.
Distributed algorithms for dynamic reassigment.
In *Proceedings of the IEEE Conference on Decision and Control*, Maui, HI, Dec. 2003, 13–18.
[21] Beard, R. W., and McLain, T. W.
Multiple UAV cooperative search under collision avoidance and limited range communication constraints.
In *Proceedings of the IEEE Conference on Decision and Control*, Maui, HI, Dec. 2003, 25–30.
[22] Finke, J., Passino, K. M., and Sparks, A.
Cooperative control via task load balancing for networked uninhabited autonomous vehicles.
In *Proceedings of the IEEE Conference on Decision and Control*, Maui, HI, Dec. 2003, 31–36.
[23] Matarić, M. J.
Issues and approaches in the design of collective autonomous agents.
Robotics and Autonomous Systems, **16** (1995), 321–331.
[24] Balch, T., and Arkin, R. C.
Behavior-based formation control for multirobot teams.
IEEE Transactions on Robotics and Automation, **14** (Dec. 1998), 926–939.
[25] Suzuki, I., and Yamashita, M.
Distributed anonymous mobile robots: Formation of geometric patterns.
SIAM Journal on Computing, **28**, 4 (1999), 1347–1363.
[26] Billard, E. A., and Riedmiller, A. E.
Stability conditions for agents pursuing individual goals under uncertainty.
In *Proceedings of the IEEE International Symposium on Autonomous Decentralized Systems*, Apr. 1995, 360–367.
[27] Subramanian, S. K., and Cruz, J. B.
Adaptive models of pop-up threats for multi-agent persistent area denial.
In *Proceedings of the IEEE Conference on Decision and Control*, Maui, HI, Dec. 2003, 510–515.
[28] Dubins, L.
On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal position.
American Journal of Math, **79** (1957), 497–516.
[29] Sussman, H. J., and Tang, G.
Shortest paths for the Reeds-Shepp car: A worked out example of the use of geometric techniques in nonlinear optimal control.
Rutgers University, New Brunswick, NJ, Technical Report SYCON-91-10, 1991.
[30] Perkins, J. R., and Kumar, P. R.
Stable, distributed, real-time scheduling of flexible manufacturing/assembly/disassembly systems.
IEEE Transactions on Automatic Control, **34** (Feb. 1989), 139–148.
[31] Quijano, N., Gil, A. E., and Passino, K. M.
Experiments for decentralized and networked dynamic resource allocation, scheduling, and control.
IEEE Control Systems Magazine, **25** (Feb. 2005), 63–79.
[32] Kumar, P. R., and Seidman, T. J.
Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems.
IEEE Transactions on Automatic Control, **35** (Mar. 1990), 289–298.

Alvaro E. Gil received his B.S. and M.S. degrees in electrical engineering from Instituto Universitario Politécnico, Barquisimeto, Venezuela, in 1990 and 1998, respectively, and the Ph.D. degree in electrical engineering from The Ohio State University, Columbus, in 2003.

From 1990 to 1999, he held engineering positions in the Automation Department at Petróleos de Venezuela (PDVSA) in Maracaibo, Venezuela. From 2002 to 2003 he was a research associate at the Department of Electrical Engineering, The Ohio State University. He worked as a postdoctoral researcher at the Ohio State University between 2003 and 2005. He is now with Xerox Corporation, Webster, NY. His current research interests include networked cooperative resource allocation strategies, cooperative scheduling, distributed mobile sensor networks, and multiobjective control.

Kevin M. Passino (S'79—M'90—SM'96—F'04) is currently a Professor of Electrical and Computer Engineering at The Ohio State University and Director of the OSU Collaborative Center of Control Science that is funded by AFOSR and AFRL/VA.

Dr. Passino has served as the Vice President of Technical Activities of the IEEE Control Systems Society (CSS); was an elected member of the IEEE Control Systems Society Board of Governors; was the Program Chair of the 2001 IEEE Conference on Decision and Control; and is currently a Distinguished Lecturer for the IEEE Control Systems Society. He is coeditor (with P. J. Antsaklis) of the book *An Introduction to Intelligent and Autonomous Control* (Kluwer Academic Press, 1993); coauthor (with S. Yurkovich) of the book *Fuzzy Control*, (Addison Wesley Longman Publishing, 1998); coauthor (with K. L. Burgess) of the book *Stability Analysis of Discrete Event Systems*, (John Wiley and Sons, 1998); coauthor (with V. Gazi, M. L. Moore, W. Shackleford, F. Proctor, and J. S. Albus) of the book *The RCS Handbook: Tools for Real Time Control Systems Software Development*, (John Wiley and Sons, NY, 2001); coauthor (with J. T. Spooner, M. Maggiore, R. Ordonez) of the book *Stable Adaptive Control and Estimation for Nonlinear Systems: Neural and Fuzzy Approximator Techniques*, (John Wiley and Sons, NY, 2002); and author of *Biomimicry for Optimization, Control, and Automation*, (Springer-Verlag, London, UK, 2005).

Andrew Sparks received his B.S. and M.S. in mechanical engineering from the Massachusetts Institute of Technology, Cambridge, in 1986 and 1988 and his Ph.D. in aerospace engineering from the University of Michigan, Ann Arbor, in 1995.

Since 1988 he has been with the Air Vehicles Directorate at the U.S. Air Force Research Laboratory at Wright Patterson Air Force Base, OH, where he has worked on robust, multivariable control systems for a variety of air and space vehicles and subsystems. Most recently he worked as a team lead for the Unmanned Air Vehicle group of the Control Science Center of Excellence.

Sriram Ganapathy received his M.S. in electrical and computer engineering at The Ohio State University. He is currently a controls engineer at General Motors Global Research and Development at Honeoye Falls, NY.