Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Information Sharing in Cooperative Unmanned Aerial Vehicle Teams

Yan Liao[1], Yan Jin[1], Ali A. Minai*[1], Marios M. Polycarpou[1,2]
[1]Dept. ECECS, University of Cincinnati, Cincinnati, OH 45221-0030
[2]Dept. ECE, University of Cyprus, 75 Kallipoleos, Nicosia 1678, Cyprus

***Abstract*— This paper considers the problem of cooperative search and task response in a heterogeneous team of unmanned aerial vehicles (UAVs) with limited communication. The UAVs are engaged in a mission to search and verifiably destroy targets in an uncertain environment. The UAVs do not have access to centralized information, and each UAV makes its decisions based on subjective information obtained through observation and communication with other UAVs. We present a strategy for information sharing and fusion, and study the impact of this strategy's parameters on the performance of the UAV team.**

I. INTRODUCTION

Autonomous vehicles are currently used for difficult and hazardous missions such as space exploration, mine-clearing and aerial surveillance. With technological improvements reducing costs, they are likely to find extensive use in broader areas such as disaster relief and environmental monitoring. An especially promising possibility is that groups of such vehicles might work cooperatively to accomplish complex tasks over extended areas. Developing efficient algorithms for such cooperative behavior is, therefore, an important and active area of research [1].

In this paper, we consider the case of unmanned aerial vehicles (UAVs), engaged in a cooperative search and destroy mission over an extended region. Individual UAVs scan the region, searching for targets, verifying their existence, attacking them with appropriate munitions, and confirming their destruction. However, rather than operating separately, they cooperate in two ways: 1) by sharing information among the team; and 2) by coordinating their tasks. In general, such coordinated scheduling is computationally prohibitive, but heuristic methods have been proposed by many researchers to achieve reasonable performance [2], [3], [4], [5], [6], [7], [8].

In our previous work, we have reported on an algorithm where UAVs autonomously make cooperative task allocation decisions using a common *information base*, providing them with a global view of the mission [7], [8]. The focus of the approach is on the use of very simple decision-making rules by individual UAVs rather than solving a global optimization problem. However, the assumption of a centralized information base represents a serious limitation from the viewpoints of efficiency, practicality and security. In this paper, we report on a decentralized version of our approach.

The decentralized system we describe requires the specification of three components:

1) **The information-sharing policy (ISP):** Who communicates what to whom, when and how?
2) **The information-fusion policy (IFP):** How is received information combined with the existing subjective information base (SIB)?
3) **The decision-making algorithm:** How does each UAV use its SIB to make its decisions?

The main contribution of this paper is to define an information-sharing policy based on three communication contingencies, an information-fusion policy comprising four fusion cases and an estimation component supplementing the decision algorithm. The goal is to bring the UAV team's performance as close as possible to that achieved by the centralized algorithm with noise-free, instantaneous communication.

II. PROBLEM FORMULATION

A. Scenario

The mission scenario considered throughout this paper comprises the following elements:

- A bounded $L_x \times L_y$ *mission environment* in which the UAVs operate. The environment is represented in the UAVs' information bases as a grid of cells:
$$\{(x, y), x = 1, \ldots, L_x, y = 1, \ldots, L_y\}.$$
- M stationary targets, T_i, $i = 1, \ldots, M$. Of the M targets, M_k are suspected initially while $M_h = M - M_k$ need to be discovered gradually through search. The value of M or M_h is unknown *a priori*. The targets are drawn from M_T types. It is assumed that each cell contains at most one target, and that targets do not cross cell boundaries.
- N heterogeneous UAVs, u_k, $k = 1, \ldots, N$. The characteristics of the UAVs are described in detail in Subsection C below.

The mission of the UAV team is to discover, identify and verifiably destroy all targets in the environment.

B. Environment Description

Each cell, (x, y), has a *target identifier*, $J(x, y, t) = j$, $j = 0, \ldots, M_T$, indicating that cell (x, y) has a target of type j at time t. A value of $J(x, y, t) = 0$ denotes the absence of a target in (x, y). The target identifiers are not known to the UAVs, and their estimation is a primary mission task.

Please address all correspondence to Ali A. Minai (ali.minai@uc.edu).

A canonical *task set*, G, defines the tasks that the UAVs can undertake at cell (x, y). Here

$$G = \{Search,\ Confirm,\ Attack,\ BDA\}$$

where *BDA* stands for battle damage assessment. UAVs cooperatively *search* the environment for unknown targets and seek to *confirm* the existence and type of any that are suspected. Once a target is confirmed and identified with sufficient certainty, an appropriately armed UAV proceeds to *attack* it. The results of the *attack* are verified through a *BDA* task, where a UAV scans the target location to assess damage.

The *search* task is termed an *automatic* task, since every UAV automatically scans each cell it passes through. The other three tasks — *confirm, attack* and *BDA* — are termed *assignable* tasks, since they are assigned to a specific UAV that must go to the designated location and execute the task.

Every cell, (x, y), has a *task status*, $\tau(x, y, t) \in G$, at time t. Each *assignable* task has a *task assignment status*, $\varsigma(x, y, t)$, drawn from the set

$$H = \{Available, Associated, Assigned, Active\}.$$

The assignment status indicates whether the task is open for volunteering (available), has been provisionally assigned to some UAVs but may still be re-assigned (associated), has been firmly assigned to a UAV (assigned), or is currently being executed by a UAV at the location (active).

Together, the $J(x, y, t)$, $\tau(x, y, t)$ and $\varsigma(x, y, t)$ describe the *task environment* in which the UAVs operate.

C. UAV Model

The state, $L(u_k, t) = \{\Lambda(u_k, t), \Gamma(u_k, t)\}$, of UAV u_k at time t comprises two parts: the *physical state*, $\Lambda(u_k, t)$, and the *functional state*, $\Gamma(u_k, t)$.

1) UAV Physical State: The physical state $\Lambda(u_k, t)$ includes the following:

- A unique *ID*, u^k, used to tag information sent out to other UAVs.
- An *expertise matrix*, $\Xi^{u_k}(t) = \{\xi^{u_k}_{j,\tau}(t)\}$ where $\xi^{u_k}_{j,\tau}(t)$ indicates UAV u_k's capability for performing task τ on a target of type j at time t. In the current model, we assume a fixed expertise matrix, $\Xi^{u_k} = \{\xi^{u_k}_{j,\tau}\}$.
- *Position* $(x^{u_k}(t), y^{u_k}(t))$, *speed* $v^{u_k}(t)$ and *heading angle* $\psi^{u_k}(t)$. UAV u_k moves on a continuous trajectory modelled with a widely used kinematic model [9].

2) UAV Functional State: The functional state, $\Gamma(u_k, t) = (x^{u_k}_D(t), y^{u_k}_D(t), \tau^{u_k}_D(t), \varphi^{u_k}(t))$, indicates the current destination $(x^{u_k}_D(t), y^{u_k}_D(t))$ of UAV u_k (if any), the task $\tau^{u_k}_D(t)$ it must perform at that destination, and the corresponding *commitment status* $\varphi^{u_k}(t)$. The commitment status, $\varphi^{u_k}(t)$, of UAV u_k takes a value from the set:

$$K = \{Open,\ Competing,\ Committed,\ Executing\},$$

indicating whether the UAV has no commitment (*open*), is volunteering for a task or has been associated with one (*competing*), is assigned to a task (*committed*), or is performing it (*executing*). The UAV's commitment status matches the assignment status of its destination task.

3) UAV Actions: As UAV u_k moves in the environment, it performs an *action* at each cell it visits or scans. The canonical *action set* is:

$$A = \{null, sense, attack\}.$$

- If the action is *sense*, the sensor system on the UAV returns an *observation value* $b^{u_k}(x, y, t) \in \{0, 1, ..., M_T\}$ indicating that UAV u_k detects a target of type j in cell (x, y) at time t. The sensor model used here [9] considers possible sensor inaccuracy and has a fixed rectangular footprint.
- If the action is *attack*, the UAV fires a munition at time t, trying to destroy the target in cell (x, y).
- If no action is taken on cell (x, y), UAV u_k has a *null* action. The state of the cell is left unchanged in that case.

D. Subjective Information Base

As mentioned earlier, each UAV in the decentralized model carries its own *subjective information base (SIB)*, representing its view of the mission status. Information in a UAV's SIB comes from two sources: 1) Information generated by the UAV's own actions; and 2) Information received from other UAVs. Due to constraints such as limited communication range and periodic (rather than continuous) transmission, a UAV's SIB is expected to have inaccuracies and uncertainties, and the SIBs of two UAVs will not, in general, be identical.

The SIB, $I^{u_k}(t)$, of UAV u_k at time t consists of the following components:

- The *target occupancy probability (TOP)* map, $P^{u_k}(x, y, t) = \{P^{u_k}_j(x, y, t), j = 0, \ldots, M_T\} \forall (x, y)$ at time t, where $P^{u_k}_j(x, y, t)$ is the estimated probability that a target of type j is present at (x, y, t). The probabilities satisfy the constraint

$$\sum_{j=0}^{M_T} P^{u_k}_j(x, y, t) = 1. \qquad (1)$$

  The TOP map is used by the UAV to estimate $J(x, y, t)$ and $\tau(x, y, t)$.
- The *task environment estimate (TEE)*, $\Upsilon^{u_k}(x, y, t) = \{J^{u_k}(x, y, t), \tau^{u_k}(x, y, t), \varsigma^{u_k}(x, y, t)\}\ \forall (x, y)$ at time t, where J^{u_k}, τ^{u_k} and ς^{u_k} are the UAV's estimates of the *target type*, *task status* and *assignment status* at (x, y, t), respectively. The TEE is used by the UAV to make decisions on which tasks to volunteer for.
- The *UAV map*, $L^{u_k}(u_s, t) = \{\Lambda^{u_k}(u_s, t), \Gamma^{u_k}(u_s, t)\}$ $\forall u_s$ at time t, where $\Lambda^{u_k}(u_s, t)$ and $\Gamma^{u_k}(u_s, t)$ are estimates of UAV u_s's physical and functional states, respectively, by UAV u_k.
- The *uncertainty map*, $\chi^{u_k}(x, y, t), \forall (x, y)$ at time t, quantifying how much UAV u_k does not know about the environment. It is used to direct the *search* for targets if no *assignable task* is available.

- The *history buffer*, $\Delta^{u_k}(x,y,t)\forall(x,y)$at time t, which has the following information for each cell: 1) A *reference TOP-task pair*, $\pi^{u_k}(x,y)$; 2) A *reference time-stamp*, $\bar{t}^{u_k}_{x,y}$, which is the time at which the reference TOP was actually generated; and 3) All internal or received sensor readings for cell (x,y) generated since $\bar{t}^{u_k}_{x,y}$.
- The *broadcast buffer*, δ^{u_k}, which records information to be sent out to other UAVs in the next broadcast. It consists of: 1) sensor readings, $b^{u_s}(x,y,t'_{x,y})$, for various cells, (x,y), taken by UAV, u_s, at times $t'_{x,y}$ associated with the source UAV ID, u_s, and time stamp, $t'_{x,y}$, for each reading; 2) the UAV's own state, $L(u_k,t^-)$, at t^-, the time just prior to t,tagged with UAV ID; and 4) single cell TOPs, $\hat{P}(x',y',t^*_{x',y'})$, for some cells, (x',y') with corresponding task transitions $\tau^{u_k}(x',y',t^*_{x',y'})$ at $t^*_{x',y'}$ (see below).

III. INFORMATION BASE DYNAMICS

UAV u_k's SIB is updated continually to reflect its current knowledge of the environment. Updates to the SIB are triggered in two ways: 1) *Autonomous updates*, based on the UAV's own actions; and 2) *Communication-based updates*, caused by information received from other UAVs. In this section, we describe the procedure for updating each component of the SIB in all circumstances.

A. Autonomous SIB Updates

1) TOP Dynamics: When UAV u_k makes an observation or executes an attack at location (x,y), it updates its TOP map by using a simplified version of the Bayesian update function previously developed by us [7], [8].

2) TEE Update: UAV u_k's new estimate of a target's presence in the cell is explicitly reflected by its TOP map. The *task assignment status* is updated according to u_k's *commitment status*.

The dynamics of the task state in each cell, (x,y) is also determined by changes in the TOP map . This is modelled as a deterministic automaton, g_h, whose transitions depend on threshold crossings in $P(x,y,t)$(Fig. 1),

$$\tau^{u_k}(x,y,t) = g_h(\tau^{u_k}(x,y,t^-), P^{u_k}(x,y,t); \bar{\rho}), \quad (2)$$

where the parameter vector $\bar{\rho}$ represents the set of threshold values used for transitions. More details can be found in our previous work [7], [8].

3) UAV Map Update: The states of other UAVs are updated based only on information received through communication, and has no autonomous component.

4) Uncertainty Update: The uncertainty, $\chi^{u_k}(x,y,t)$, represents the UAV's uncertainty about two things: 1) the existence of a target in (x,y); and 2) the type of the target. The uncertainty is a passive function of the TOP, $P^{u_k}(x,y,t)$, and does not have its own dynamics. Thus, it is updated

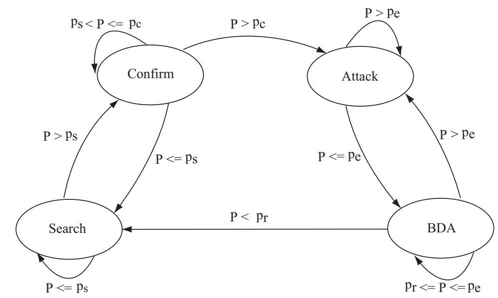

Fig. 1. Task Dynamics: p_s=suspicion threshold, p_c=certainty threshold, p_e=exit threshold, p_r=resolution threshold.

each time the TOP is updated, using the definition:

$$\begin{aligned}\chi^{u_k}(x,y,t) &= g_u(P^{u_k}(x,y,t) \\ &= \omega_\chi[-P^{u_k}_0(x,y,t)\log P^{u_k}_0(x,y,t) \\ &- (1-P^{u_k}_0(x,y,t))\log(1-P^{u_k}_0(x,y,t))] \\ &+ \frac{(1-\omega_\chi)}{\log M_T}[\sum_{l=1}^{M_T}(-P^{u_k}_l(x,y,t)\log P^{u_k}_l(x,y,t))], \quad (3)\end{aligned}$$

where ω_χ is a parameter between 0 and 1. The first term of the equation (3) is the entropy of target existence, while the second term is the entropy of target type.

5) History Buffer Update: Autonomous updates of the history buffer for cell (x,y) occur as follows:

- Whenever UAV u_k makes an observation, $b^{u_k}(x,y,t)$ in (x,y), it is added to the history buffer.
- When a TOP update leads to a task transition in $\tau^{u_k}(x,y,t)$, the resulting TOP-task pair is stored in the history buffer as the reference, $\pi(x,y)$, along with its time-stamp as the reference time-stamp, $\bar{t}^{u_k}_{x,y}$, and all sensor readings for that cell are cleared from the buffer.

6) Broadcast Buffer Update: An autonomous update to the broadcast buffer is straight forward. To put sensor readings, u_k's current state and/or reference TOP-task pairs into the buffer.

B. Communication-Based SIB Updates

Communication-based updating of the SIB can be seen in terms of two components: 1) an *information-sharing policy* that specifies what is communicated to whom under what conditions; and 2) an *information-fusion policy* that specifies how new information is merged with existing information to update the SIB. Both of these are discussed in this section.

1) Information-Sharing Policy: Communication between UAVs is assumed to occur over a network that is noise free and instantaneous. However, communication is range-limited. The *communication range* of UAV u_k is defined as $\gamma^{u_k}_c$. We assume that all UAVs have the same communication range, $\gamma^{u_k}_c = \gamma_c, k = 1, \ldots, N$, so that UAVs within range of each other can communicate mutually. Thus all links, when they exist, are undirected, and the communication between UAVs is mediated by a continuously changing undirected graph.

The information-sharing policy we use comprises three *communication contingencies* as follows:

- *Regular broadcast*: Each UAV sends out the contents of its broadcast buffer δ^{u_k} every t_b seconds, and reinitializes the buffer.
- *Event-triggered broadcast*: When a sensing action at time t by UAV u_k causes a task transition in cell (x, y), or the UAV executes an *attack* leading to a consequent change in the TOP and task of the cell, it immediately sends out the updated TOP, $\hat{P}(x, y, t^*_{x,y})$, the corresponding new task status, $\tau(x, y, t^*_{x,y})$, and the transition time stamp, $t^*_{x,y}$, for cell (x, y).
- *Opportunistic SIB Exchange*: UAVs u_k and u_s exchange their entire SIBs if: 1) they are closer than a threshold distance γ_o at time t; and 2) they were outside of the threshold γ_o at time t^-.

For the present, the lengths of the history buffer and the broadcast buffer are assumed to be unconstrained, though we are also considering the finite-length cases.

2) Information Fusion Contingencies: The communication contingencies described above lead to a UAV u_k possibly receiving four kinds of information, thus requiring four protocols to incorporate the information into its own SIB.

The four kinds of information are:

- The state,$L(u_s, t^-)$, of a UAV u_s at t^-, the time just prior to t, and UAV ID, u_s;
- A sensor reading, $b^{u_s}(x, y, t'_{x,y})$, for a cell (x, y), along with the time stamp, $t'_{x,y}$, and the ID, u_s;
- An event-generated TOP-task update $\pi(x, y, t^*_{x,y})$ along with the update time stamp, $t^*_{x,y}$;
- An entire SIB, $I^{u_s}(t^-)$.

These are merged as follows:

Single UAV State:

- Accept the new update as is if u_k and u_s are within communication range of each other;
- Otherwise extrapolate last received state of u_s to estimate its current one.

Sensor Readings:

- Check History Buffer $\Delta^{u_k}(x, y, t)$ for sensor reading $b^{u_s}(x, y, t'_{x,y})$.
- If $t'_{x,y} \geq \bar{t}^{u_k}_{x,y}$, and the reading is not found, update u_k's SIB according to the dynamics in Autonomous Updates, and add the reading to the history buffer; if not, discard the reading.

Thus, if the received reading is recent enough to belong in the history buffer but is not found there, it is considered novel, and is treated like an observation by the UAV itself. It is discarded if it is found to be a duplicate (so it has already been incorporated in the current TOP), or if it pre-dates a subsequent event-driven TOP update.

If multiple sensor readings are received for the same cell, they are applied sequentially by iterating *TOP Dymamics*. This is possible because the TOP update function is commutative with respect to updates (see [7] for details). The other components of SIB update accordingly.

However, if the TOP update caused by received sensor readings leads to a task transition, it won't be treated as an *Event* since such a transition is only u_k's estimate.

Event-Generated Single-Cell TOP-Task Update:

- If the received TOP's time-stamp is more recent than the current (x, y) reference time-stamp, $\bar{t}^{u_k}_{x,y}$, of the history buffer:
  - Replace the existing reference TOP, reference task status and reference time-stamp in the history buffer with the received values, $\hat{P}(x, y, t^*_{x,y})$, $\tau(x, y, t^*_{x,y})$ and $t^*_{x,y}$, respectively.
  - Update the TOP $P^{u_k}(x, y, t)$ with sensor readings in the history buffer whose time-stamps are later than $t^*_{x,y}$.
  - Update the task status $\tau^{u_k}(x, y, t)$ based on the updated TOP, and adjust the TEE $\Upsilon^{u_k}(x, y, t)$ accordingly.
  - Discard all sensor readings for cell (x, y) with time-stamps earlier than $t^*_{x,y}$ from the history buffer.
  - Add the received TOP, task status and time-stamp to the broadcast buffer.
  - Update the uncertainty map, $\chi^{u_k}(x, y, t)$ according to the updated TOP map.
- If the received TOP is not more recent than the current reference time-stamp, discard it.

Since an event-generated update represents information coming directly from the UAV responsible for the update, it is accepted at face value unless there is an even more recent update available. However, UAV u_k may also have received (or generated) readings for cell (x, y) in the time *after* the update's time-stamp (and incorporated them in its current TOP). The TOP is recalculated by incorporating these recent readings into the received update, and the resulting TOP and task status are to replace the current TOP map and the TEE for cell (x, y).

Entire Subjective Information Base:

When u_k receives the entire SIB from another UAV, u_s, it updates every component of its SIB except the broadcast buffer, since there is no newly generated information to broadcast by u_k. The update proceeds as follows.

For each cell, (x, y):

- Accept the more recent reference TOP, task status and time-stamp.
- Merge all sensor readings without duplication; discard all sensor readings with time-stamps earlier than the updated reference time from the history buffer.
- Update the reference TOP with sensor readings in the updated history buffer and replace the current $P^{u_k}(x, y, t)$ with this value.
- Update the task status $\tau^{u_k}(x, y, t)$ based on the updated TOP, and adjust the TEE $\Upsilon^{u_k}(x, y, t)$ accordingly.
- Update the uncertainty map, $\chi^{u_k}(x, y, t)$ according to the updated TOP map.

For other UAVs' states, accept the information originated from the source UAV then estimate its current situation.

IV. COOPERATIVE DECISION MAKING

The approach taken by the UAVs to make control decisions is to 1) maintain their SIB as well as possible (as described above); 2) estimate unknown or incomplete information, including the current status of other UAVs and targets. This, along with the SIB, comprises the *subjective situation estimate (SSE)*; 3) make a decision based on the assumption that the information in the SSE is correct.

The motivation for step (1) is obvious. The degree of completeness in the SIB is balanced against the cost of communication and security considerations, as discussed earlier. Step (2) is required because information in the SIB is not up-to-date, and a UAV cannot make its decision without knowing (or estimating) what the current status of other UAVs and targets is. Step (3) is simply based on practicality: Once it has made its best effort to obtain an accurate SSE, the UAV has no choice but to treat it as true.

To obtain its SSE, UAV u_k treats itself as u_s in order to estimate the actions of u_s and their results. The estimated information is taken as correct until the true status is received by u_k. In this way an estimated SIB $\bar{I}^{u_k}(t)$ is obtained.

In general, the decision making algorithm is realized as follows. First, UAV u_k selects a candidate assignable task from the estimated TEE $\bar{\Upsilon}^{u_k}(x, y, t)$. If no assignable tasks are available, the uncertainty map $\bar{\chi}^{u_k}(x, y, t)$ is used to guide the *search* task.

In order to cooperate with other members, UAV u_k searches through its UAV map $\bar{L}^{u_k}(u_s, t)$. UAV u_k calculates each UAV u_s's *cost value*, $h_{s,i}^{u_k}$, with respect to all *available* or *associated* assignable tasks, $\tau_i^{u_k} \in \tau^{u_k}(x, y, t)$:

$$h_{s,i}^{u_k} = \omega_c d_{s,i}^{u_k} + (1 - \omega_c) \exp(-\xi_{\tau_i^{u_k}}^{u_s}), \tag{4}$$

where ω_c is a parameter valued in [0,1], $d_{s,i}^{u_k}$ is the normalized distance between UAV u_s and the location of task $\tau_i^{u_k}$, and $\xi_{\tau_i^{u_k}}^{u_s}$ is the expertise of UAV u_s for task $\tau_i^{u_k}$.

UAV u_k works as if it is the central controller, and uses a semi-greedy bipartite matching algorithm to match UAVs with tasks. UAVs that are within distance D_a of their matched task are $assigned$ the task and are given the $committed$ status, while UAVs that are further away are $associated$ with their matched tasks and are given the $competing$ status. We allow only one UAV to be $assigned$ to a task but up to n_a UAVs can be $associated$ with one task. Similarly, each UAV can only be $committed$ to a single task, but we allow it to be $competing$ for up to m_a tasks. When a UAV has no task to choose, it has $open$ status and follows a *path of maximum local uncertainty*, i.e., one that takes it through cells with the highest uncertainty in its local neighborhood — within turning constraints. The purpose is to maximize the benefit from *search* in a greedy way, and the path followed is termed a *search path*.

After such an initial assignment, UAV u_k moves toward its own task and assumes that all other UAVs move according to its commands: each UAV with an $assigned$ task moves towards that task, UAVs with no $assigned$ task move towards their lowest-cost $associated$ task, while the rest follow search paths. UAV u_k takes sensor readings as it moves and updates its own TOP. When UAV u_k reaches its $assigned$ task, it performs the task and updates its own TOP there. A new task (possibly the same as the previous one) is then cued according to the transition function, and the UAV u_k's status reverts to $open$. Each new assignable task is cued with an $available$ status. All changes in UAV u_k's SIB are communicated according to the communication contingencies.

At all times, UAV u_k communicates with other reachable UAVs, estimates their costs for all $available$ and $associated$ tasks and changes its decision accordingly. All UAVs work the same way as UAV u_k does. The process continues until the region is completely searched and all targets are neutralized, or some time limit is reached.

V. SIMULATION RESULTS AND DISCUSSION

To evaluate the effect of the proposed strategy on the performance of cooperative UAV teams, we conducted Monte Carlo simulations using an event-driven simulator. The purpose of the simulations was to investigate how the quality of the information available to each UAV under the decentralized approach was affected by: 1) the communication range, γ_c, and 2) the broadcast interval, t_b, for regular broadcasts.

In the simulations, we consider two types of UAVs: *target recognition (TR)* UAVs and *attack (A)* UAVs, which are characterized by the difference of their expertise matrices, $\{\xi_{j,\tau}^{u_k}\}$. All UAVs are assumed to have sensors needed for search, but with different sensing capabilities; only *(A)* UAVs have the capability of accomplishing *attack*. The goal for the UAV team is to detect and neutralize all targets as rapidly as possible. Thus, we use this time, termed the *target neutralization time (TNT)* as the measure of performance.

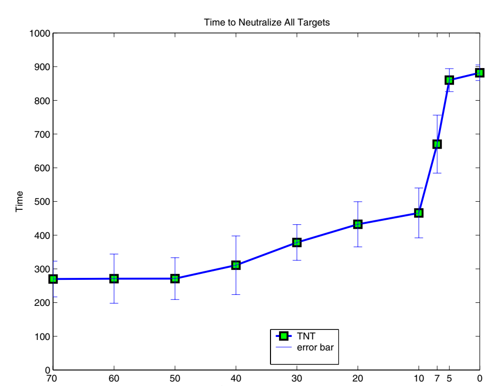

Fig. 2. The effect of communication range to the time in neutralizing all targets

The first set of simulation results (Fig. 2) illustrates the effect of different communication ranges to the performance of a team of 2 TR and 2 A UAVs in an 50×50 environment with 4 targets which are all suspected *a priori*. The communication range is varied from $50\sqrt{2}$ (global communication) to 0 (no communication). In this case, the communication interval is fixed at 1 time unit. The graph clearly shows how communication range will affect cooperation among the UAV team. When the UAVs have a perfect communication network, complete neutralization is achieved faster. Not much deterioration in performance occurs when the communication range drops from $50\sqrt{2}$ to 50, since the UAV team can still share information most of the time. When the communication range decreases further, performance begins to decline slowly. However, a significant loss of performance does not occur until the communication range is close to zero, where each UAV is operating entirely on its own and all information sharing is achieved by observing the environment. The latter situation is called *stigmergetic communication* in the swarm literature [10], and our results show that even a very limited active information-sharing strategy can improve cooperative performance significantly over purely stigmergetic communication.

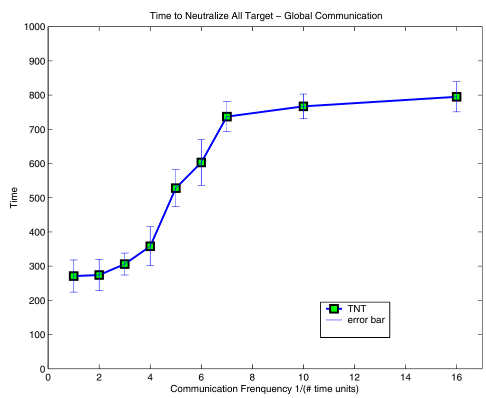

Fig. 3. The effect of communication frequency to the time in neutralizing all targets

In the second set of simulations, the interval between regular broadcasts by UAVs is varied, but communication is assumed to be global. From Fig. 3, the following phenomena can be observed: 1) As the communication frequency changes from *once every 1 time unit* to *once every 4 time units*, TNT does not increase too much. The reason is that there is not much information accumulated in the buffer within a 4- time-unit delay. 2) TNT increases almost linearly as the frequency of broadcasts changes from *once every 4 time unit* to *once every 8 time units*. 3) Beyond a frequency of once every 8 steps, TNT again becomes flat, indicating that the benefits of information sharing have been largely lost by this point, and each UAV is operating mostly based on its own observation.

Though the results reported here are based only on limited simulations, several interesting observations can be made:

1) Performance has a highly nonlinear dependence on both communication range and frequency. In particular, there seems to be a soft threshold for both parameters, beyond which performance degrades rapidly by a factor of 2 to 3. This suggests that the benefits of cooperation in the system studied have an approximately "all-or-none" form. It may also be possible to link this with ideas from critical systems and percolation theory [11] and the theory of random graphs [12], [13]. However, that requires further study of the connectivity features of the virtual network defined by the UAVs.
2) Cooperation based on information exchange — even if it is imperfect — leads to a very significant improvement in performance. For example, there is almost a 70% reduction in TNT when the UAVs go from no communication to communicating over a small transmission range of 10 units, or from communicating once every 16 time steps to communicating once every 4 steps.
3) It is possible to use significantly less than global communication range and to communicate less frequently than once every time step without a serious loss of performance. This indicates that a decentralized approach can lead to significant savings in communication overhead and a reduction in the risk of detection by adversarial surveillance.

VI. CONCLUSION

The reseach results indicate that significant improvements in efficiency and security are possible by using a decentralized scheme with limited communication, and provide motivation for further research on selecting appropriate communication parameters based on the UAV team composition and mission characteristics.

REFERENCES

[1] P. Chandler and M. Pachter, "Research issues in autonomous control of tactical UAVs," in *Proc. of Amerian Control Conference*, 1998, pp. 394–398.
[2] M. Polycarpou, Y. Yang, and K. Passino, "A cooperative search framework for distributed agents," in *Proceedings of the 2001 IEEE International Symposium on Intelligent Control*, September 2001, pp. 1–6.
[3] A. Gil, K. Passino, and A. Sparks, "Cooperative scheduling of tasks for networked uninhabited autonomous vehicles," in *Proc. of the 42nd IEEE Conference on Decision and Control*, Maui, HI, Dec 2003), pp. 522–527.
[4] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning," in *Proc. of European Control Conference*, 2001, pp. 2603–2608.
[5] R. Beard and T. McLain, "Multiple UAV cooperative search under collision avoidance and limited range communication constraints," in *Proc. of the 42nd IEEE Conference on Decision and Control*, Maui, HI, Dec 2003, pp. 25–30.
[6] M. Flint, E. Fernández-Gaucherand, and M. Polycarpou, "Cooperative control for UAV's searching risky environments for targets," in *Proc. of the 42nd IEEE Conference on Decision and Control*, 2003, pp. 3567–3572.
[7] Y. Jin, M. Polycarpou, and A. Minai, "Cooperative real-time task allocation among groups of UAVs," in *Recent Developments in Cooperative Control and Optimization*, S. Butenko, R. Murphey, and P. Pardalos, Eds. Kluwer Academic Publishers, 2004, pp. 207–224.
[8] Y. Jin, Y. Liao, A. Minai, and M. Polycarpou, "Balancing search and target response in cooperative unmanned aerial vehicle (uav) teams," *IEEE Transactions on Systems, Man and Cybernetics - Part B*, to appear.
[9] P. Chandler, S. Rasmussen, and M. Pachter, "UAV cooperative path planning," in *Proc. of AIAA Guidance, Navigation, and Control Conference*, 2000, pp. 1255–1265.
[10] M. Dorigo, G. D. Caro, and L. Gambardella, "Ant algorithms for discrete optimization," *Artificial Life*, vol. 5, no. 2, pp. 137–172, 1999.
[11] D. Stauffer and A. Aharony, *Introduction to Percolation Theory*. London: Taylor & Francis, 1992.
[12] B. Bollobas, *Random Graphs*. London: Academic Press, 1985.
[13] R. Albert and A.-L. Barabasi, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, pp. 47–97, 2002.