Distributed Decision and Control for Cooperative UAVs using Ad-Hoc Communication

Yosi Ben-Asher, Moran Feldman, Sharoni Feldman and Pini Gurfil

Abstract— This work develops a novel distributed algorithm for task assignment (TA), coordination and communication of multiple UAVs engaging multiple targets and conceives an ad-hoc routing algorithm for synchronization of target lists utilizing a distributed computing topology. Assuming limited communication bandwidth and range, coordination of UAV motion is achieved by implementing a simple behavioral flocking algorithm utilizing a tree topology for distributed flight coordination. Distributed TA is implemented by a relaxation process, wherein each node computes a temporary TA based on the union of the TAs of its neighbors in the tree. The computation of the temporary TAs at each node is based on weighted matching in the UAV-target distances graph. A randomized sampling mechanism is used to propagate TAs among different parts of the tree. Thus, changes in the location of the UAVs and targets do not pass through the root of the tree. Simulation experiments show that the combination of the flocking and the TA algorithms yields the best performance.

***Index Terms*—Distributed Control, Distributed Algorithms, Mobile Communication.**

I. Introduction

The problem of design, development and control of multi-agent systems has been studied in recent years for many applications. In particular, the use of systems consisting of multiple autonomous robots or unmanned aerial vehicles (UAVs) has been proposed in order to meet the requirements of complex missions [1]. Control, communication and decision support systems for UAVs constitute rapidly evolving research and development fields, as indicated by the Department of Defense UAV Roadmap 2002-2027 [2]. The use of groups of cooperating UAVs in order to perform various missions is currently studied throughout the world and is considered a main research goal by the United States Air Force Research Laboratory (AFRL) [3].

Assigning multiple UAVs to perform tasks cooperatively is a challenge that requires the development of specialized algorithms [4]-[7]. These algorithms may be classified into two main types: *optimal* and *heuristic*. While optimal algorithms yield better results in terms of task assignment (TA) [9], they are usually more sensitive to system uncertainties, enemy behavior, and environment changes. Heuristic algorithms [10], on the other hand, are usually sub-optimal but more robust. An issue strongly related to cooperative UAV motion is *flocking* (also referred to as *formation flying*), which has been extensively studied in the last two decades [11], following the seminal work of Reynolds [8].

In this work, we are concerned with solving the following cooperative decision and control problem: *A given number of UAVs and targets are dispersed on a given theater. Maximize the ratio between the number of intercepted targets and the number of launched munitions given a fixed number of flock payloads*

We use the Metrical Routing Algorithm (MRA) [12] for ad-hoc communication. This algorithm maintains connectivity by dynamically connecting the UAVs using a minimal set of rooted spanning trees (RSTs). The proposed algorithms for coordinating the UAVs uses the RST structure as a black-box building block. Thus, the dynamic communication details are masked out by the dynamics of the underlying set of RSTs. We mainly focus on coordination and decision support rather than on details related to the ad-hoc communication.

The TA algorithm is implemented by a relaxation processes over the RST nodes. Each node computes a flow of updated TA plans by collecting TA plans from its neighbors in the tree and merging them. This relaxation process achieves two goals simultaneously:

- Distribution: Independent generation of TA plans at each node, such that each plan is updated to include only local changes.
- Robustness: Local changes are propagated to other nodes through a common root only.

We use randomized sampling to overcome the relatively slower rate at which global information is propagated. This sampling is done by sending locally updated TAs to remote nodes selected at random. This algorithm is an enhanced version of a basic version that appeared in [13].

Thus, while most researchers assume a given system, propose a new control algorithm, and examine the algorithm's performance compared to other known algorithms, we take a novel approach and develop a distributed UAV decision and control system comprising *all* three layers: *flocking*, *communication* and *task assignment*. We evaluate the system's performance by examining *efficiency*, measured by the ratio between the number of killed targets and the number of munitions launched by the group of UAVs at a given time. For our simulation experiments, we assume that every UAV is equipped with two types of sensors: A Ground Moving Target Indicator (GMTI) that detects vehicle movement and an Electro Optical (EO) sensor used to track the target and guide the missiles. The detection radius of the GMTI is assumed to be larger than the detection radius of the EO sensor. We show that the proposed algorithms considerably increase the flock efficiency.

II. The Flocking Layer: Heuristic Relative UAV Control

This section discusses the heuristic control algorithm for UAV coordination based on Reynolds' behavioral flocking algorithm [8], which may be summarized as follows: Let $U_k \in G$ denote some UAV constituting a node in the graph G. U_k calculates its desired velocity as follows:

$$\mathbf{v}_d^k = \sum_{i=1}^{4} w_i^k \mathbf{v}_i^k \qquad \mathbf{v}_d^k \in \mathbb{R}^3, \tag{1}$$

where w is a constant scalar weight function, k is the UAV index and i is the algorithm law index, defined by the following *flocking rules*:

$i=1$: *Cohesion*; commands the UAV to converge onto the center of the flock, computed by each UAV from the data communicated to it by the other UAVs. We denote the desired cohesion velocity for U_k by $\mathbf{v}_1^k$, and by $\mathbf{x}^k$ the position vector of U_k. The cohesion velocity command may be written as

$$\mathbf{v}_1^k = \left\|\mathbf{v}^k\right\| \cdot \frac{\left(\mathbf{x}_{avg}^k - \mathbf{x}^k\right)}{R_{ref}}, \tag{2}$$

where $\|\cdot\|$ denotes the Euclidian vector norm, R_{ref} is a reference distance, usually related to the maximum payload detection range, representing the effective area of the UAV payload, and

$$\mathbf{x}_{avg}^k = \frac{\sum_{j=1}^{n} r_1^{jk} \mathbf{x}^j}{\sum_{j=1}^{n} r_1^{jk}}, \tag{3}$$

where r_1^{jk} is the cohesion rule weight for U_k relative to U_j, given by $r_1^{jk} = r_1(t, \mathbf{x}^k, \mathbf{x}^j)$. In Eq. (3) and the subsequent equations, n denotes the number of nodes of some subtree $G' \in G$, and not necessarily the total number of UAVs, to be denoted by *N*.

Although r_1^{jk} may be time dependant, it is more likely that it would be directly dependant upon the relative position, increasing as the relative distance between the UAVs decreases, or remain constant.

$i=2$: *Alignment*; matches the UAV's velocity vector to the mean velocity vector of the group. Alignment therefore attempts to steer the UAVs to fly in parallel to each other. We denote the desired alignment velocity for U_k by $\mathbf{v}_2^k$, and let r_2^{jk} be the alignment weight for U_k relative to U_j, so that

$$\mathbf{v}_2^k = \mathbf{v}_{avg}^k = \frac{\sum_{j=1}^{n} r_2^{jk} \mathbf{v}^j}{\sum_{j=1}^{n} r_2^{jk}} \tag{4}$$

Similarly to r_1, $r_2^{jk} = r_2(t, \mathbf{x}^k, \mathbf{x}^j)$ may be constant, time-dependant, or a function of the relative distance between U_k and U_j.

$i=3$: *Collision avoidance*; restricts the UAV from colliding with its nearest neighbors. To that end, U_k calculates its desired collision avoidance velocity, $\mathbf{v}_3^k$, relative to the other UAVs according to the formula

$$\mathbf{v}_3^k = \left\|\mathbf{v}^k\right\| \frac{\sum_{j=1}^{n} r_3^{jk} \cdot (\mathbf{x}^j - \mathbf{x}^k)/R_{ref}}{\sum_{j=1}^{n} r_3^{jk}}, \quad \forall j \neq k, \tag{5}$$

where r_3^{jk} is the collision avoidance rule weight of U_k avoiding collision with U_j, and $\mathbf{x}^j$ and $\mathbf{x}^k$ are the position vectors of U_j and U_k, respectively. The weight function r_3^{jk} is likely to be dependant upon the relative distance between U_k and U_j, equaling 1 for the closest neighbor to UAV *k* and 0 for all other UAVs.

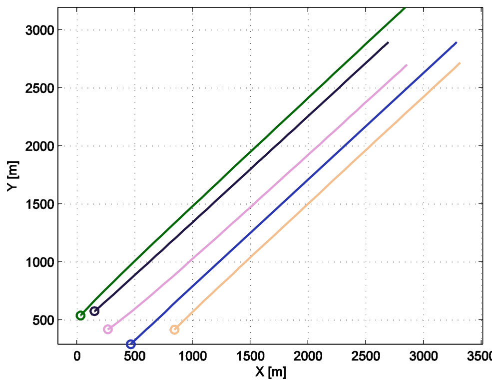
The desired velocity (1) is translated into an acceleration command using the following kinematic equation:

$$\mathbf{a}^k(t) = \frac{[(\mathbf{v}^k \times \mathbf{v}_d^k) \times \mathbf{v}^k]}{\left\|\mathbf{v}^k\right\|^2 \left\|\mathbf{v}_d^k\right\|} g\sqrt{(n_{max}^k)^2 - 1}, \tag{6}$$

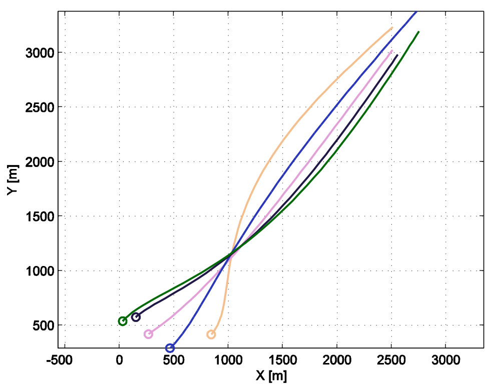
where $\mathbf{v}^k \in \mathbb{R}^3$ and $\mathbf{v}_d^k \in \mathbb{R}^3$ are the current and desired velocity of U_k, respectively, and n_{max}^k is the maximal load factor of U_k. The term $\mathbf{v}^k \times \mathbf{v}_d^k$ in Eq. (6) yields a vector perpendicular to the plane defined by the velocity vectors $\mathbf{v}^k$ and $\mathbf{v}_d^k$. This perpendicular vector is then vector multiplied again by $\mathbf{v}^k$ to define the direction, perpendicular to $\mathbf{v}^k$, in which the UAV will accelerate in order to reach the desired velocity $\mathbf{v}_d^k$. The quotient defines a unit vector in the desired maneuver direction, and then multiplied by the UAV maximal load factor to give the maneuver magnitude. This acceleration is integrated into velocity and position using the kinematic model:

$$\begin{aligned} \mathbf{v}^k[t(i)+\Delta t] &= \mathbf{v}^k[t(i)] + \mathbf{a}^k[t(i)]\Delta t \\ \mathbf{x}^k[t(i)+\Delta t] &= \mathbf{x}^k[t(i)] + \mathbf{v}^k[t(i)]\Delta t \end{aligned} \tag{7}$$

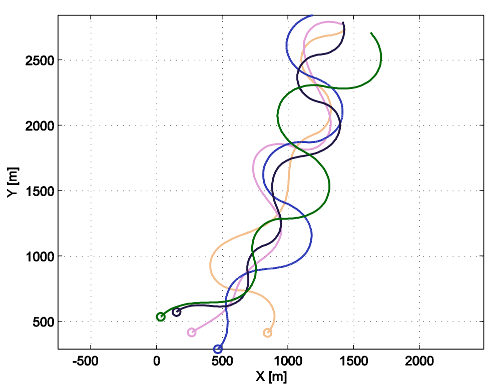
Fig. 1 depicts a number of flocking scenario, implementing different weights on the cohesion, alignment and collision avoidance rules. This figure shows that the movement of the UAVs in the field can be modified by a proper selection of the flocking weights.



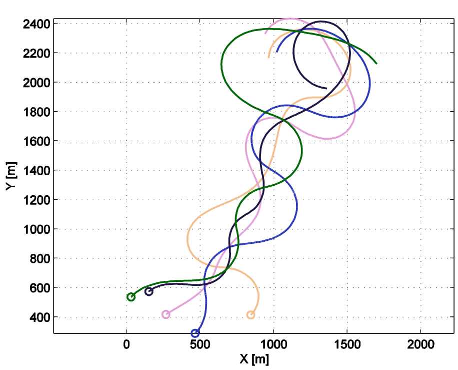
(a) $w_1 = 0, w_2 = 0.2, w_3 = 0.8$



(c) $w_1 = 0.01, w_2 = 0.99, w_3 = 0$



(b) $w_1 = 0.01, w_2 = 0.01, w_3 = 0.98$



(d) $w_1 = 0.1, w_2 = 0.2, w_3 = 0.7$

Fig. 1. 5-UAV flocking (starting points marked with an 'o'): (a) no cohesion creates straight and parallel paths to avoid collisions; (b) a combination of cohesion and alignment with much favored collision avoidance creates curved paths; (c) no collision avoidance creates a possible collision point; (d) increasing the cohesion and alignment weights increases the turn radii.

The implementation of Reynolds' algorithm in this work is carried out using a new approach: The flocking algorithm controls the velocity and heading of the UAVs. However, each UAV communicates with its closest neighbors *only* and is unable to get a global view of the heading and velocity of the entire flock. The control information including the flocking data propagates from node to node using the tree management protocol.

III. The Communication Layer: Metrical Routing Algorithm

In this section, we describe the metrical routing algorithm (MRA) [12], used as an ad-hoc communication protocol between the UAVs for communicating target list and flocking information.

The MRA protocol presented herein is a hybrid ad-hoc protocol in the sense that some traffic control is used to maintain the mapping of the communicating nodes. The small overhead of the MRA protocol used to maintain the mapping is a worthy investment, as the MRA is capable of handling successfully a demanding traffic load under a high node density and fast node movement. The MRA organizes the nodes in rooted trees in order to find short session paths between nodes on the tree. The algorithm attempts to minimize the number of trees by fusing separate adjacent trees into a single tree. As long as any node in one tree is not in the transmission range of any node in the other trees, the trees will function autonomously. As soon as a radio connection is created between two nodes, the trees will be fused into a single tree.

The MRA algorithm organizes the nodes in the field in rooted trees. Only nodes that belong to the same tree can create sessions among themselves. To ensure maximal connectivity, all nodes will try to organize themselves in a single tree. Every node in the field has a unique node-ID (similar to a phone number or an IP address) and virtual coordinates that may change depending on the changes in the tree structure. Every tree is identified by a "tree name" which is the ID of the root node. Nodes periodically send beacons, termed *hello* messages. Every node that receives a beacon checks whether the node that sent the beacon belongs to a different tree. If the nodes belong to different trees, they initiate a fusion process that fuses the separate trees into a single tree.

Initially, every node forms a separate tree of size 1. Every node in the tree can autonomously migrate to a neighboring tree regardless of the node position in the tree. The migrating node gets new coordinates in its new tree according to the node's new position. Naturally, when a node migrates from one tree to a new tree, it may carry along its neighboring nodes (since it belongs now to a bigger tree). In the macro view, the migration of the single nodes creates a fusion of smaller trees into larger ones.

The fusion process of two trees is parallel, that is, at any given time step, multiple nodes of the smaller tree join the larger tree. The implementation of the flocking and TA algorithms is based on the tree structure. Every tree runs these algorithms autonomously, as it cannot necessarily communicate with other trees. Existence of such communication will initiate a merge process that will ultimately result in a single tree.

IV. The Task Assignment Layer: Target List Management

The TA layer relies on the arrangement of the UAVs in trees and on the inter-communication capabilities using ad-hoc routing. Every UAV is *autonomous*, performing autonomous decisions and behaving according to the changes in the theater. However, when a UAV becomes a node in a tree created by the MRA, it upgrades its behavior and acts as a member of a group.

When a UAV operates as an individual – that is, when flocking and TA are disabled - each UAV randomly selects a flying heading and continues to fly in this direction until one of the following happens:

(a) The sensors detect a potential object to intercept. In this case, the UAV will select an optimal route in order to intercept the target. The process of calculating the route is performed under the UAV flight and maneuvering limitations.

(b) The UAV reaches the border of the theater. In this case, the UAV selects one of the following:

1. Return back to the theater using the same heading angle (ψ).
2. Return with a random heading angle.
3. Return to the field using the heading angle of $\pi - \psi$.

(c) The UAV randomly changes its flying direction with a probability of 10^{-5} per simulation cycle.

In this section we consider the problem of computing a targeting plan for a set of moving agents $G = \{U_1, U_2, \ldots, U_N\}$ (UAVs in our case) attacking moving targets $A = \{T_1, T_2, \ldots, T_m\}$ (vehicles in our scenario). We focus on a distributed solution over a special setting where the communication among $U_1, U_2, \ldots, U_N$ is carried out by an ad-hoc network, as described in the previous section. Using ad-hoc communication yields a complex and challenging setting wherein the following factors should be considered:

- Ad-hoc communication implies that communication links among $U_1, U_2, \ldots, U_N$ are constantly changing. Thus, there is no guarantee that a given subset of G that was previously connected will remain connected.
- At any stage new information regarding (a) new targets, (b) changes in the location of known targets and (c) new U_i's that are closer to a given target can pop-up.
- It is desired not to fix a targeting plan (i.e., assign targets to each U_i) in advance, but rather adopt the reactive setting wherein at any time step only a portion of the targets are assigned to some subset $G' \in G$.
- Centralized algorithms where all the data (location of $U_1, U_2, \ldots, U_N$ and $T_1, T_2, \ldots, T_m$) is collected and then processed may fail to obtain good solutions due to disrupted communication and long communication delays.

We hereby suggest an enhanced of the TA algorithm described in [13]. In this algorithm, target assignments are communicated among the UAVs using the MRA protocol described above. Unlike other ad-hoc routing algorithms, the MRA attempts to connect $G = \{U_1, U_2, \ldots, U_N\}$ (or a subtree thereof) by a minimal set of rooted trees that preserves geographical distances, namely distances on the rooted trees are usually proportional to the distances of $U_1, U_2, \ldots, U_N$ in the given engagement theater.

More formally, let $G(t)$ be the graph at time *t* wherein each two nodes U_i, U_j that can communicate have an edge in $G(t)$. The MRA algorithm attempts to cover $G(t)$ by a minimal set of spanning trees. These rooted trees can be naturally used for both distributed computing (of, e.g., the flocking layer) as well as for communication, in addition to propagation and computation of the TA layer. The proposed TA algorithm for $U_i \in G$ using the MRA protocol can be thus summarized as follows:

1. Each node U_i in a tree (or a subtree) locates all the detectable targets, identifies them and computes its distance to each target. The target ID is the target location. Note that computing a unique target ID is not always straightforward, since it may require fusion of the target location taken by several UAVs in adjacent time steps and locations.
2. At each time step $t(i)$, a node *v* constructs a weighted bipartite graph $B_v[t(i)]$ representing the distances between each U_i and T_j related to the subtree rooted at *v*. There are three events that lead to the creation of a new bipartite graph $B_v[t(i+1)]$:
   - A new $B_u[t(i+1)]$ is received from one of *v*'s children.
   - A new $B_{Fv}[t(i+1)]$ is received from *v*'s father or by a remote node through the random sampling mechanism.
   - There is a change in the target list **L** of *v*, i.e., *v* detects a new target or an old target disappears or destroyed.

   In each of these events, a new $B_v[t(i+1)]$ is computed by merging $B_u[t(i+1)]$ or $B_{Fv}[t(i+1)]$ or **L** into $B_v[t(i+1)]$.
3. The node *v* computes a minimum weighted matching $M_v[t(i+1)]$ of $B_v[t(i+1)]$ obtaining an attack plan that

minimizes the sum of distances of the UAVs in the subtree of *v* to their targets,

$$d^* = \sum_{i,j \in B_v[t(i+1)]} d_{ij} \quad (8)$$

4. $B_v[t(i+1)]$ is sent to the father F_v of *v* and $M_v[t(i+1)]$ is sent to all the children of *v*.
5. When a node *v* receives an attack plan $M_{Fv}[t(i+1)]$ from its father, it checks to see if it is assigned a new target; if so, it "leaves" its current target and starts to engage the recommended target.
6. The attack plan $M_{Fv}[t(i+1)]$ is sent to all the children of the current node.

Note that in case a target is destroyed or disappears, it will be removed from each $B_v(t)$, since these are propagated only up the MRA trees.

The implementation of the target selection algorithm uses a single data structure to transfer the bipartite graph $B_v(t)$ and the attack graph $M_v[t(i+1)]$. This data structure is the *Target List* (TL). A simplified TL model is depicted by Fig. 2. This model ignores the parallelism in the TL flow between the UAVs. It explains the decision-making process and the decision overruling performed by higher levels of UAVs in the tree with broader views.

Fig. 2A presents the initial phase where U_4 and U_5 have detected target T_1, U_6 and U_7 have detected T_2, and U_8 detected T_3. Every UAV that has one or more targets will *autonomously* select a target from the possible targets in its TL and will commence a pursuit. The current state depicted in Fig. 2A is that U_4 and U_5 are in pursuit after T_1, U_6 and U_7 prosecute T_2 and U_8 will pursuit T_3. The pursuit process of the UAVs is independent of other UAV activities. Note that this is the initial phase, where the targets were detected by the GMTI detector but are not yet within the range of the UAV missiles launch distance (i.e. within the FOV of the EO payload). Every UAV stores a TL comprising all targets known to the UAV and indications on the target state. Every UAV then sends its TL to its father and children. U_4, U_5 and U_6, constituting leaves in the tree, send their TLs toward node 1, which is the subtree father. The decision taken by U_1 arrives to U_4, which is also the UAV attacking T_1. U_4 continues its attack while U_5 receives the same TL from its father, and finds out that it should abort its attack on T_1. U_5 will look for an alternative target without an owner in the TL that is within its GMTI range, or will search for a new target that might emerge.

Fig. 2B presents a situation in which U_1 had analyzed the TLs and decided that sub-tree *D* will be responsible to attack T_1, sub-tree *E* will not attack T_1 and sub-tree *F* will attack T_2. The decisions of U_1 are sent to its father node 0 and its children. A similar process takes place in the other parts of the tree.

In Fig. 2C, the root distributes the results of its decisions to its children. The decisions are embedded in its TL. The decision of U_0 is that subtree *C* will assume the responsibility to attack T_2, while subtree *A* will abandon its attack. These decisions will be distributed by every subtree towards its children until they reach the leaves. In the meantime, U_6 and U_7 continue their pursuit after T_2.

Fig. 2D presents a situation where the root decisions arrived to the attacking UAVs, and U_6 stopped its attack on T_2 while U_7 continues its attack. The upstream and downstream flow of TLs is not affected by changes in the tree structure or by appearance of new targets.

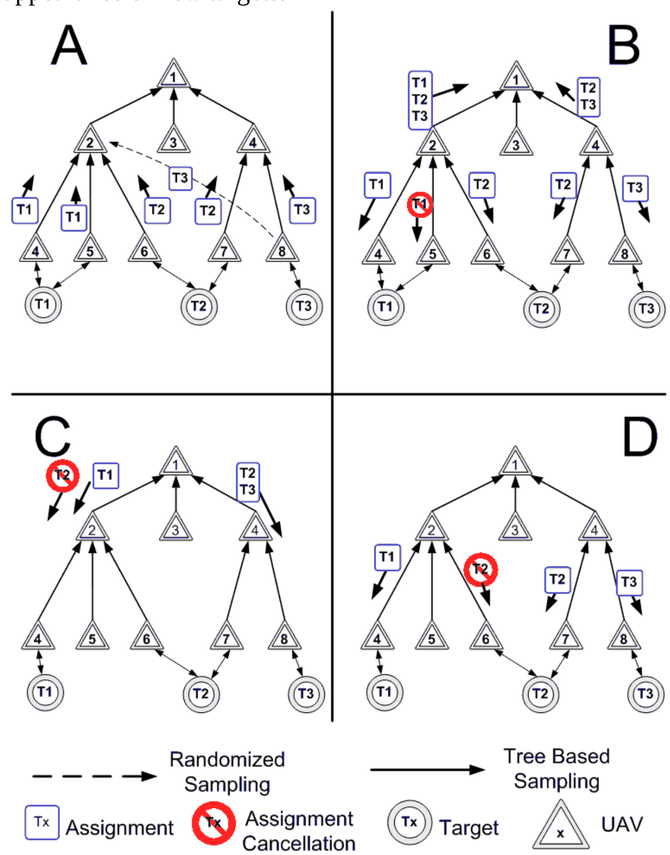


Fig. 2. Target List flow

V. Simulation and Visualization

Fig. 3 presents a snapshot of the theater as created by the simulator. The UAVs are identified by their position in the tree, where R is the root, R.1 is one of the children of the root and R.1.1 is a child of R.1. Fig. 3A presents the detection footprint of R.1.1 while Fig. 3B presents the detection footprint of R.1.

R.1.1 in Fig. 3A detected two potential targets. One of the targets, marked by a cross, was selected as the target to be attacked and this is the 1st priority target for this UAV. R.1 in Fig. 3B also detected two targets, where one of the targets is observed by both UAVs. The target marked with a cross was selected as the target to be attacked. Note that this UAV is attacking the farthest target and not the closest one. This decision was taken after analyzing the velocities and headings of the entities participating in this pursuit or by a decision of the upper layer in the tree (node R).

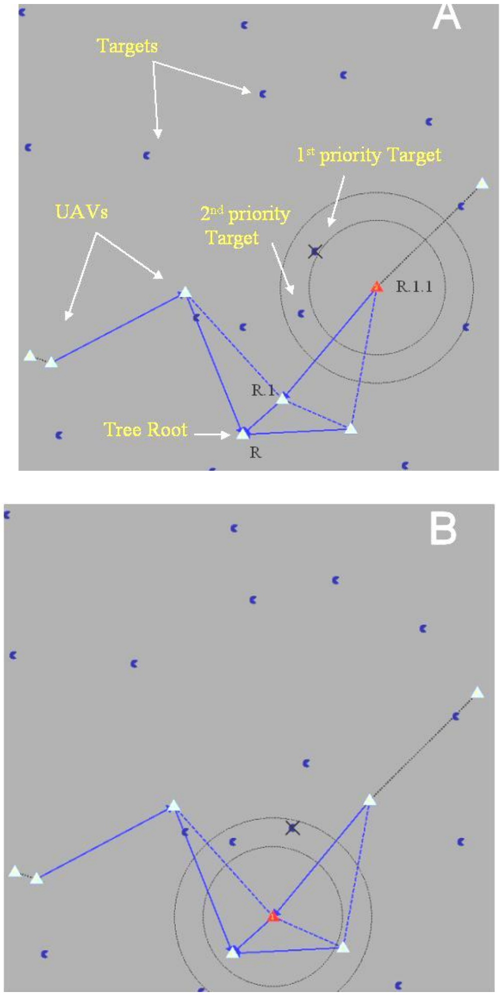


Fig. 3. Theater view during engagement.

VI. Experimental Results

The simulation experiments are aimed at evaluating the contribution of the flocking and TA algorithms to the performance of the UAVs using MRA-based ad-hoc communication. The main experiments comprise the following benchmarks:

1. Reference Monte-Carlo simulations performed without employing the flocking and TA algorithms;
2. Monte-Carlo simulations used to evaluate the contribution of the flocking algorithm; and
3. Combined Monte-Carlo tests where both flocking and TA algorithms are employed simultaneously.

Each simulation run in Cases 1-3 has been performed assuming that the missile hit probability is unity. All simulations were performed using a time interval of $t_f = 100$ sec. Additional simulation parameters are listed in Table 1.

Table 1: Simulation parameters

Theater dimensions	14 Km x 14 Km
UAV initial speed	100 Km/h – 120 Km/h (uniform distribution)
UAV radio transmission range	1.7 Km
Target speed	50 Km/h - 80Km/h (uniform distribution)
No. of targets	16
No. of UAVs	8-16
Missiles	Fixed value of 16. In case of 8 UAVs, every UAV carries 2 missiles. In case of 16 UAVs, every UAV carries 1 missile. For all other cases, every UAV carries randomly 1 or 2 missiles.
TL synchronization rate	30 Hz
Maximal UAV load factor	2g

The main performance evaluation measure is the average number of hits per UAV, calculated as the ensemble average over 50 Monte-Carlo runs.

Fig. 4 depicts the results of the Monte-Carlo simulations. In this figure, the *x*-axis is the UAV group size and the *y*-axis is the average number of hits per UAV. Each point on this graph represents an ensemble average of 50 Monte-Carlo runs. There are three curves shown in this figure, corresponding to Cases 1-3.

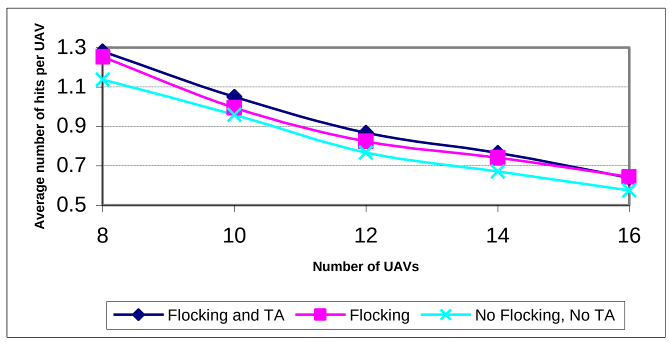


Fig. 4. Monte-Carlo simulation results for a $t_f = 100$ sec.

There are two important observations. First, it is seen that the flocking algorithm improves the average number of hits by up to 12%. Combining the TA and flocking algorithms improves the average number of hits per UAV by up to 14%. Second, increasing the number of UAVs reduces the average number of hits per UAV, as expected. Roughly, the relation is linear. This implies that when more UAVs are used, each platform can carry less munitions for intercepting the same amount if targets.

Fig. 5 shows the standard deviations of the average number of hits for the case of 14 UAVs. It is seen that the standard deviation remains practically invariant to the UAV cooperation method.

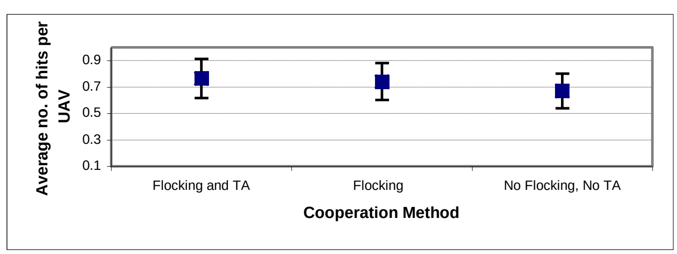


Fig. 5. Standard deviation intervals for the case of 14 UAVs

VII. Conclusions

We developed a distributed algorithm for task assignment, coordination and communication of multiple UAVs engaging multiple targets in an arbitrary theater. The algorithm used a relaxation method for computing both locally- and globally-updated task assignment plans. The relaxation is made over a tree structure generated by the underlying ad-hoc communication layer. Random sampling is used to enhance the propagation of changes between remote nodes in the theater.

Our simulation experiments raise a number of important conclusions. First, we conclude that the combination of flocking and task assignment gives the best performance, which is improved relative to the case with no flocking and no task assignment. An improvement of the average number of hits was observed for UAVs that were capable of both target list exchange and velocity coordination. Second, increasing the number of UAVs enables to reduce the amount of munitions per platform.

References

[1] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, “A taxonomy for multi-agent robotics,” *Autonomous Robots*, Vol. 3, 1996, pp. 375–397

[2] Department of Defense, *UAV Roadmap 2002-2027*, 2002.

[3] S. S. Banda, “Future directions in control for unmanned aerial vehicles,” April 2002, available at http://www.cds.caltech.edu/murray/cdspanel/fdc-apr02/banda-26apr02.pdf.

[4] M. J. Mataric, “Designing and understanding adaptive group behaviors,” *Adaptive Behavior*, Vol. 4, 1995, pp. 51–80.

[5] J. Reif and H. Wang, “Social potential fields: A distributed behavioral control for autonomous robots,” *Robotics and Autonomous Systems*, Vol. 27, 1999, pp. 171-194.

[6] M. J. Mataric, “Behavior based control: Examples from navigation, learning, and group behavior,” *Journal of Experimental and Theoretical Artificial Intelligence* , Vol. 9, No. 2-3, April 1997, pp. 323-336

[7] L. Steels, “The artificial life roots of artificial intelligence,” *Artificial Life*, Vol. 1, pp. 75–110, 1994.

[8] C. W. Reynolds, “Flocks, herds, and schools: A distributed behavioral model,” *Computer Graphics*, Vol. 21, 1987, pp. 25–34.

[9] K. Passino, M. Polycarpou, D. Jacques, M. Pachter, Y. Liu, Y. Yang, M. Flint, and M. Baum, “Cooperative control for autonomous air vehicles,” *Proceedings of the Cooperative Control Workshop, Florida*, December 2000.

[10] H. V. D. Parunak, M. Purcell, and R. O’Connell, “Digital pheromones for autonomous coordination of swarming uavs,” *Proceedings of the AIAA First Technical Conference and Workshop on Unmanned Aerospace Vehicles*, Vol. 3446, 2002.

[11] J. Toner and T. Yuhai, “Flocks, herds, and schools: A quantitative theory of flocking,” *Physical Review E*, vol. 58, No. 4, 1998, pp. 4828–4858.

[12] Y. Ben-Asher, S. Feldman, “Ad-hoc routing using virtual coordinates-based on rooted trees", *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, Taiwan, 2006.

[13] Y. Ben-Asher, S. Feldman, P. Gurfil and M. Feldman, “Hierarchical decision and control of cooperative UAVs using ad-hoc communication, *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Keystone, CO, 2006.